ISSN 1870-4069

Linguistic Elements Selection from a BCI Matrix Using Intelligent Computing

Marco Antonio Cardoso-Moreno, Francisco Hiram Calvo-Castro, Cornelio Yáñez-Márquez

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{mcardosom2021, hcalvo, cyanez}@cic.ipn.mx

Abstract. Brain-computer interfaces (BCI) are systems primarily developed to help people with a certain level of motor disability, achieve new channels of communication and interaction with their environment that depend only on each individual's brain activity, which is translated, by means of an algorithm, to the user's desired commands and actions. In other words, the main objective of BCI systems is to provide users with the ability to control computers or machines without requiring any limb movement, but only by their own brain activity instead. There are many ways in which brain activity is recorded; electroencephalography (EEG) being the technique most widely used. One of the most common BCI systems is the P300 Speller BCI, which bases its functioning on the occurence of event-related potentials (ERP); specifically P300 ERP, which appear in the EEG readings in the form of positive deflections around 300 ms after an unexpected stimulus has occured. The goal of spellers is to select and write the desired character from an array of letters presented to the user in the form of a matrix; the characters in this matrix are randomly flashed in order to activate the P300 potentials in the EEG readings. This paper analyzes the current state of the art regarding the algorithmic aspect of the BCI P300 spellers; we also present a first approach for signal preprocessing, intended to deliver feature vectors suitable for target character classification.

Keywords: BCI, EEG, ERP, P300.

1 Introduction

Brain-computer interfaces (BCI) are systems whose main purpose is to provide the user with new communication channels that are independent from those typically used by the brain, namely muscles and the nervous system [6, 3, 15-17].

BCI systems work by reading and interpreting brain activity from the user in order to translate it into commands [3, 8, 13]; data is gathered by electroencephalography (EEG) [4, 8, 13], which detects electrical charges resulting from brain activity by positioning electrodes on the user's scalp [13].

Although some other methods are available for brain activity data gathering, namely: functional magnetic resonance imaging (fMRI), near-infrared spectroscopy

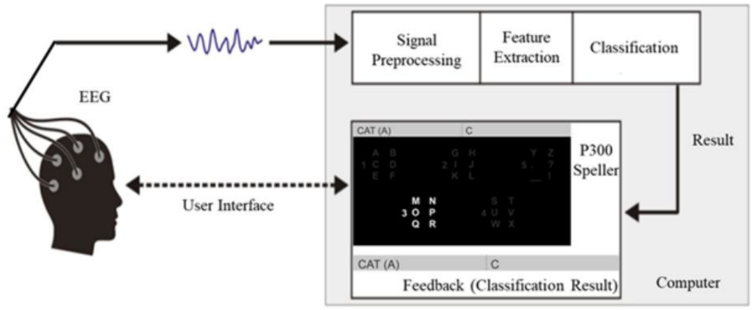

Fig. 1. Schematic of a P300 BCI speller system [13].

(NIRS) and magnetoencephalography (MEG) [**?**], EEG is widely prefered due to its noninvasiveness [6,13]

In general, a BCI consists of input and output channels, as well as an algorithm; recorded brain activity is used as input, whereas the user's desired commands correspond to the output. Between these channels we found the algorithm, which, in common terms, translates input data into output commands; since it works directly with brain activity, the algorithm must be capable of adapting to the dynamic features of this organ [17].

This work focuses on the BCI systems known as spellers, whose main objective is to provide the user with the ability of writing only by their brain activity; these systems are between the most studied ones, being the P300 BCI Speller the most widely adopted [3]. This BCI detects P300 ERP in the EEG readings, which appear in the form of a positive deflection in the signal, in the lapse of 250 ms to 750 ms after the user has been exposed to certain stimulus [18]; it bases its functioning on the oddball paradigm, which states that the less common the stimulus (unpredictable for the user), the higher the P300 intensity signal [2, 8, 15, 18]

Figure 1 shows an schematic of how, in general, a P300 BCI system is structured. The main components are: the visual interface, where the user will select the desired characters from; the EEG system needed to record brain activity followed by a signal acquisition phase; next there are a signal preprocessing stage, a feature extraction phase and a Machine Learning (ML) algorithm for the task of character classification and prediction; lastly, the algorithm's output is passed to the system in order for it to write the desired character.

The common approach to these BCI systems consists on facing the user with a 6×6 matrix, where each cell corresponds to an individual character: 26 letters, 9 digits and one blank space symbol, for a total of 36 cells. Randomly, a row or column is selected, and its light intensity is increased for a brief lapse; this procedure is carried on until all columns and rows have been intensified (Figure 2).

Therefore, for every character the user focuses their attention on, two out of twelve stimuli will contain such symbol, allowing for the isolation of particular P300 ERP corresponding to a given character in the matrix [2, 7].

Although widely adopted, there still are some aspects of the P300 BCI that need improvement. There is the need to find the right balance between the accuracy of the

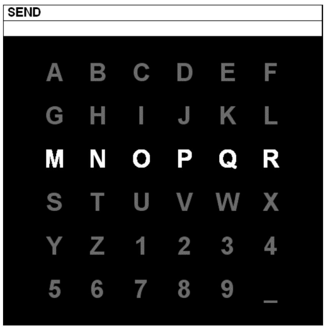

Fig. 2. Matrix presented to the user. The light intensity of the third row, from top to bottom, is increased in order to generate a P300 ERP [7].

character prediction algorithm and the information transfer rate (ITR); the algorithm's performance may vary drastically from subject to subject, and even between sessions for a given user; and lastly, there are people who do not meet the necessary ERP to use these systems [18], phenomenon generally refered to as "BCI illiteracy" or "BCI inefficiency" [5].

Our focus is, mainly, on the algorithmic aspect, specifically on character classification, recognition and prediction, as well as feature vector dimensionality reduction and signal processing. The dataset we selected to work with is the Wadsworth BCI Dataset (P300 Evoked Potentials) [7]; here, the user was faced with a matrix as the one presented in Figure 2.

2 State of the Art

In general, a P300 Speller has the following structure: a signal preprocessing phase, where one or several filters are applied to the EEG reading signals in order to increase the signal-to-noise ratio (SNR); a dimensionality reduction phase, where feature selection and feature extraction algorithms are applied to the feature vectors obtained after the preprocessing stage; and a classification phase, where a machine learning model is applied on the reduced vectors in order to classify the P300 signals [3, 4].

The classification problem consists on predicting whether a given feature vector in a character epoch belongs to one of 2 classes: target character (a P300 signal is present) or non-target character (no P300 signal is present) [10].

As these spellers are, in general, complex systems, there are two general approaches to trying to improve the system performance: the BCI user interface, where the P300 potentials are evoked, which includes the structure and layout of the characters in the screen, their morphology and the physical stimulation applied to the user; and the algorithm, which includes the preprocessing, feature extraction and selection, and classification phases [3].

There have been some efforts in improving the user interface presented to the user by, for example, imitating the nine-key keyboard layout used on mobile phones where users

Table 1. Commonly used techniques for each stage of a P300 BCI speller system.

BCI stage	Operation
Preprocessing	CAR filters Butterworth filters Chebyshev filters
Feature Selection and Extraction	ICA PCA Fast Fourier Transform Wavelet Transform Autoregressive model Cohen's class distribution Subsampling
Character Classification	Bayesian classifiers LDA SVM Artificial Neural Networks Random Forest

selected the initial characters from a 3×3 matrix, and then word suggestions were made from a dictionary [1]; also, there have been changes in the structure interface, from a 2D structure to a 3D one [11, 12].

The morphology of the symbols is another area of interest, it has been observed that changing a letter for a human face as part of the visual stimuli increases the speller's performance when compared to other, more traditional, approaches [5, 9]. Changing the typical one-color light intensification in the matrix for a mixed colors interface has also shown an increase in accuracy of approximately 16% [10]. Other approaches involve other kind of stimuli, such as auditory or a combination of both visual and auditory stimuli, as presented in [10].

Regarding the data preprocessing stages, the most common techniques used are: temporal filters, spatial filters, frequency filters, time-frequency filters and time-spatial filters [3]. For example, [10] used a common average reference (CAR) spatial filter after data standarization; [12] applied a bandpass Butterworth filter followed by spatial filters, [16] also used a Butterworth bandpass filter followed by a normalizarion procedure, whereas [13] made use of a Butterworth filter only; [4] performed a fourth order Chebyshev type I filter on the EEG signals, while [8] used an eighth order Chebyshev filter.

After the EEG data have been preprocessed, it is necessary to perform some feature selection and feature extraction tasks; here, the most common approach is to perform independent component analysis (ICA) and principal component analysis (PCA), as well as, althoug less commonly used, the Fast Fourier Transform, Wavelet Transform and autoregresive model [3].

[4] made use of the Cohen's class distribution, in conjunction with a fourth order Chebyshev filter to reduce the feature vectors dimension on their dataset. On the other hand, [8] applied a PCA to remove the less important features from their dataset.

[9, 13] used similar approaches to dimensionality reduction of their feature vectors with sub sampling, by selecting one sample for every 4 values, and 40 out of the 400 obtained samples, respectively. [18] made use of the stepwise linear discriminant

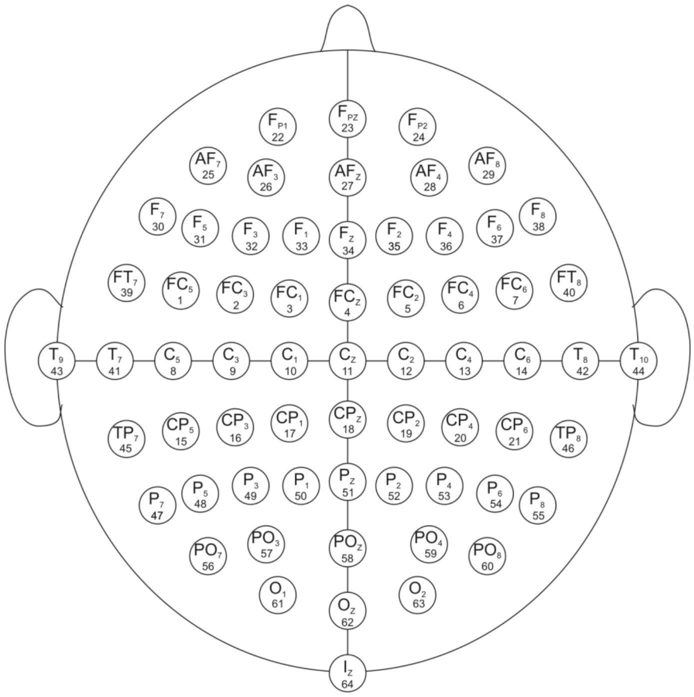

Fig. 3. Electrode layout and index designation [7].

analysis (SWLDA) to reduce the feature space by removing and adding features to it, depending on their contribution to the classification task, after the corresponding downsampling of the preprocessed EEG data.

The next step is to perform classification tasks over the resulting feature vectors, for which a machine learning algorithm is required, among the most widely adopted algorithms we found: Bayesian classifiers, linear discriminant analysis (LDA), support vector machines and artificial neural networks [3].

[2] used an LDA classifier, with a split validation of 10 symbols used for training and 40 symbols used for testing, since, according to the authors, cross-validation techniques are not a good fit for BCI applications due to the amount of data available and overfitting concerns.

LDA is also used by [5] with a validation split, where the first iteration of each subject was used as the training data and the following 4 runs were used as the testing set; performance was measured by defining the accuracy as the number vectors correctly classified as target characters divided by the total ammount of target characters; [9], as well as [12] made use of a Bayesian linear discrimant analysis (BLDA); finally, [13] performed a Stepwise linear discriminant analysis.

Support vector machines classifiers are also widely applied, [8] proposed an Ensemble of Weighted SVM (EWSVM), which was then compared against an Ensemble Weighted LDA (EWLDA) and combinations of SVM and LDA (linear

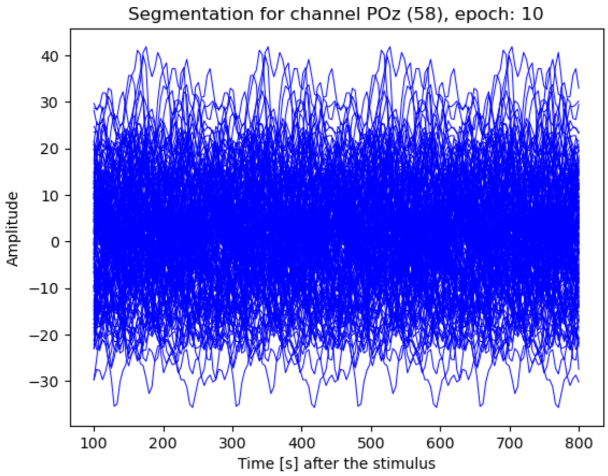

Fig. 4. Segmentation on channel POz for epoch 10.

kernels were used for all SVM implementations), the results show that the performance of EWSVM is better than that of EWLDA; [10] made a comparison between a linear SVM and an LDA, both with and without PCA feature extraction, where the LSVM without PCA was the model with best results, followed by LSVM with PCA.

[16] proposed different models, including SVM, Random Forest and Extreme Gradient Boosting (XgBoost), these models were applied to the dataset after oversampling techniques to overcome the imbalanced classes inherent to the speller; the results showed that SVM was the best algorithm with an accuracy of 94.2%, followed by XGBoost with an accuracy of 94.19% and finally, the random forest with a performance accuracy of 94.16%.

Artificial Neural Networks are another approach to character classification in P300 BCI spellers, [4] proposed the use of a Deep Neural Network (DNN), they used Stacked autoencoders to further reduce the dimensionality of the feature vectors from 651 features to 10, these 10 final features were passed to a Softmax classifier; in this study a duplication of the feature vectors labeled as target characters is carried on in order to overcome the imbalance issues. Table 1 presents commonly used techniques in each P300 BCI speller stage.

2.1 Difference of this Proposal with Respect to the State of the Art

It is observed that most of the solutions proposed in the state of the art for character classification and prediction involve certain assumptions, for example, many of the classifiers are linear, namely LDA and support vector machines with the use of a linear kernel, which means that it is assumed that the preprocessed EEG data is linearly separable.

It is important to further study up to what extent this is a valid assumption, since EEG readings have some non-linear features which might not be filtered out in the preprocessing and feature selection stages.

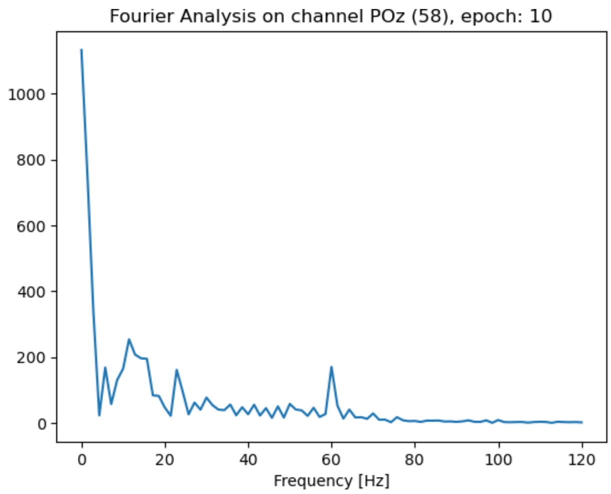

Fig. 5. Fourier analysis on Channel POz for epoch 10.

In future study, different classification algorithms will be tested in order to determine a new approach to character recognition in P300 BCI spellers, specifically: Multi-layer Perceptron, Long Short-Term Memory (LSTM) Neural Networks and a new paradigm, called Minimalist Machine Learning [19].

3 Project Development

3.1 The Dataset

As previously stated, the selected dataset for this work is the Wadsworth BCI Dataset (P300 Evoked Potentials) [7]. The dataset consists of four **.mat* files containing information on two test subjects (A and B) results, one for training and one for testing; i.e., there is a training and a testing file pair for each subject; training files have data regarding 85 character epochs, whereas testing files contain data of 100 character epochs.

Each character epoch consists of the following steps: a 2.5 s periods where the matrix was displayed with equal intensity for every row and column; then each row and column was randonmly intensified for 100 ms for a total of 12 intensifications, this process was repeated 15 times per epoch, resulting in 15 blocks of 12 intensifications (complete rows and columns), for a total of 180 intensifications; after each intensification, the matrix went back to an equal intensity state for 75 ms.

Signals were collected with a 64 electrode setup, whose layout, as well as index designation for this particular dataset, are shown in Figure 3; the matrix arrangement of characters is depicted in Figure 2. Data was gathered at a sample rate of 240 Hz.

3.2 Data Preprocessing

Signal data is presented as a three dimensional array, where the first dimension corresponds to character epochs, the second dimension corresponds to different time

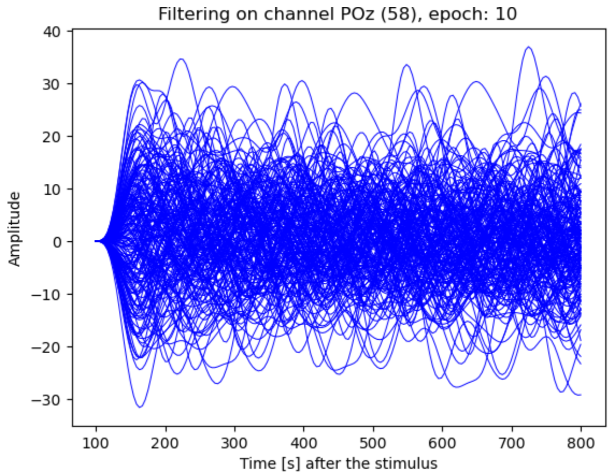

Fig. 6. Filtering on Channel POz for epoch 10.

steps along the epoch and the third dimension corresponds to each channel in the EEG hardware (electrodes). All values and results discussed in this section correspond to training data for subject A and EEG channel POz, located in the parieto-occipital region of the brain, since the occipital lobe is responsible for visual perception [14].

The first stage in data preprocessing is segmentation, i.e., breaking each character epoch into segments where the stimulus has been applied, to this purpose, for each channel, we break the whole epoch into 700 ms windows starting at exactly 100 ms after, and up to 800 ms, after the stimulus was applied; since this is the period where P300 ERP peaks show in EEG readings. This results, for each channel, in new two dimensional arrays where the first dimension corresponds to each intensification (180 in total), whereas the second contains the segment signals for 700 ms (168 samples in total). Figure 4 shows the results after the segmentation process.

From Figure 4, it is clear that EEG readings carry a lot of noise, therefore, it is imperative to apply a filter in order to reduce the ammount of noise present in the signal while keeping valuable data. With a Fourier analysis (Figure 5) we see that the frequencies at which more information was read are low frequencies, which is expected since P300 ERP are low frequency phenomena [6]. Based on these results, a filtering procedure was performed with a 5th degree Butterworth bandpass filter, with low and high cut frequencies of 0.1 and 15 Hz, respectively. After this Butterworth filter is applied, we end up with a smoother signal (Figure 6).

Once data have been filtered, a summing process is performed in order to unite all intensifications for each column or row; here an average operation was carried on, which led to a new two dimensional array of dimensions 12×168, where the first dimension corresponds to the 12 rows and columns in the character matrix, and the 168 values of the second dimension are the result of averaging those values from the 15 repetitions.

When performing a sum over the 15 repetitions of a particular stimulus intensification, it is expected that noise values will cancel each other, since these readings are considered to be completely random values, while the signal of interest

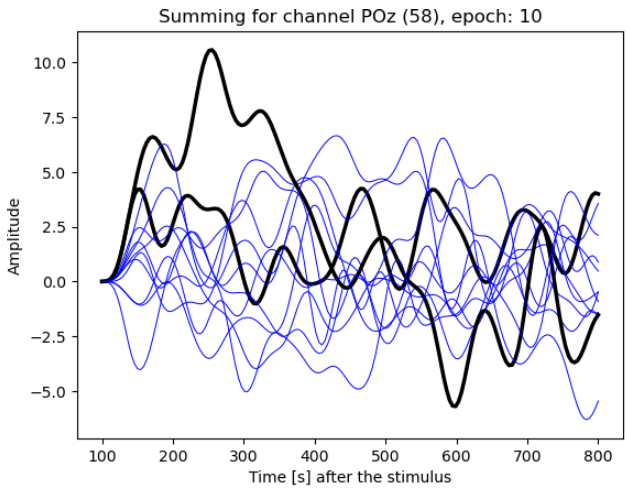

Fig. 7. Resulting signals after summing on channel POz for epoch 10. The thickest signals show the row and column containing the target character.

will be present in each repetition, thus, amplifying the corresponding reading values; the arithmetic mean is computed in order to return the signal amplitude to its initial range of values.

In Figure 7, we show the resulting average values for each row and column in the matrix, here, two signals are depicted as thicker lines, while the rest of the signals appear as thinner series, in the background; the thicker signals correspond to the row and column that contained the target character of the tenth epoch for subject A.

The nex step in the preprocessing stage is to downsample the arrays in order to reduce data dimensionality without losing valuable information; downsampling was carried on via a decimation procedure with a downsampling factor $q = 5$, meaning that every fifth signal value is preserved while the previous four are thrown away thus, reducing the signal dimensionality by a factor of five. This reduction makes each vector for a row or column, to reduce its number of elements from 168 to 34.

When comparing Figures 7 and 8, we see that the average shape of the signals is conserved, hence, downsampling is a valid approach for dimensionality reduction when working with EEG reading signals, since the general structure of the problem is preserved while reducing the number of attributes the model will deal with, demanding less computational resources and providing results in less time; aspects of primary importance in BCI systems.

Then, intersections of each row and column are computed by averaging both signals. In Figure 9, all indexes are shown as represented in the dataset so, for example, the average of column 1 and row 7 represents the signal read for character A, since it is the character at the intersection of such row and column; the average of column 5 and row 10 corresponds to character W, and so on.

This operation results in a new array of dimensions 36×34, the first dimension corresponds to each one of the characters in the matrix, while the second dimension contains the average values of the intersection for a given character. Figure 10 shows

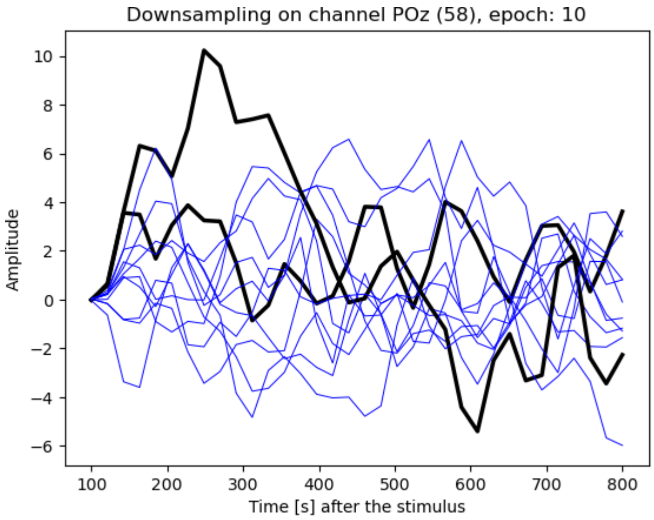

Fig. 8. Downsampling on Channel POz for epoch 10. The thickest signals show the row and column containing the target character.

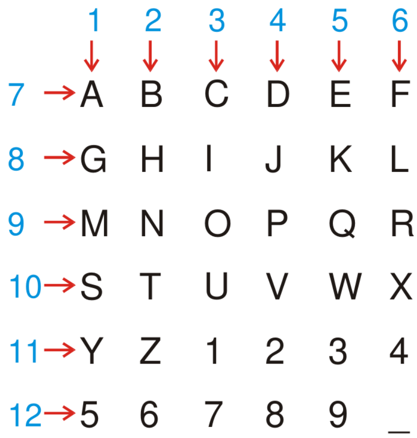

Fig. 9. Indexes of rows and columns in the dataset [7].

the resulting signal values, averaged for each character in the matrix, with the given target character appearing as the thickest signal, for epoch 10.

For this case, the goal of any ML classifier is to perform as best as possible, by trying to assign the thicker signal to the target character class, while assigning the rest of the signals to the non-target character class.

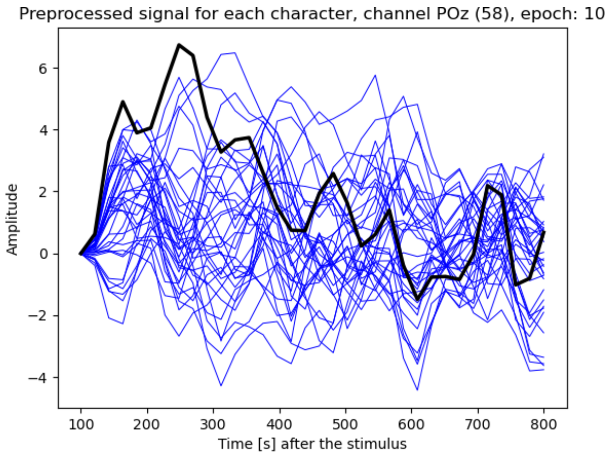

Fig. 10. Signal averaged values for each character in the epoch. The thickest signal corresponds to the target character of the epoch.

4 Conclusions

In this work we presented a summary of the state of the art for P300 Speller BCI, making emphasis on the techniques most widely used in every stage of the system, as well as pointing out some of the deficiencies still present, namely: BCI illiteracy and ML algorithms performance, from a computational perspective.

We also presented the signal preprocessing stage for the given dataset. Although results were only shown for one particular example, i.e., one subject, one epoch and one electrode, this procedure, utilizing the same set of parameters, applies for the whole dataset nevertheless.

It is important to emphazise the validity of the approach presented. The several plots shown indicate that even by reducing the feature vectors dimensionality by five, the main characteristics of the signals are preserved, which should allow for a classifier to perform without issues, within its own limitations.

This aspect results of great importance since two models to be tested in future work are neural network architectures, known for requiring longer times of training when compared to other ML paradigms.

References

1. Akram, F., Han, S. M., Kim, T. S.: An Efficient Word Typing P300-BCI System Using a Modified T9 Interface and Random Forest Classifier. Computers in Biology and Medicine, vol. 56, pp. 30–36 (2015). DOI: 10.1016/j.compbiomed.2014.10.021.
2. Artzi, N. S., Shriki, O.: An Analysis of the Accuracy of the P300 BCI. Brain-Computer Interfaces, vol. 5, no. 4, pp. 112–120 (2018). DOI: 10.1080/2326263X.2018.1552357.
3. Fang, T., Song, Z., Niu, L., Le, S., Zhang, Y., Zhang, X., Zhan, G., Wang, S., Li, H., Lin, Y., Jia, J., Zhang, L., Kang, X.: Recent Advances of P300 Speller Paradigms and Algorithms. In: 2021 9th International Winter Conference on Brain-Computer Interface (BCI), pp. 1–6 (2021). DOI: 10.1109/BCI51272.2021.9385369.

4. Ghazikhani, H., Rouhani, M.: A Deep Neural Network Classifier for P300 BCI Speller Based on Cohen's Class time-frequency Distribution. Turkish Journal of Electrical Engineering and Computer Sciences, vol. 29, no. 2, pp. 1226–1240 (2021). DOI: 10.3906/elk-2005-201.
5. Guger, C., Ortner, R., Dimov, S., Allison, B.: A Comparison of Face Speller Approaches for P300 BCIs. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 004809–004812 (2016). DOI: 10.1109/SMC.2016.7844989.
6. Guger, C., Ortner, R., Dimov, S., Allison, B.: A Comparison of Face Speller Approaches for P300 BCIs. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 004809–004812 (2016). DOI: 10.1109/SMC.2016.7844989.
7. Krusienski, D., Schalk, G.: Documentation Wadsworth BCI dataset (P300 evoked potentials). https://www.bbci.de/competition/iii/desc_II.pdf. (2004)
8. Kundu, S., Ari, S.: P300 Detection with Brain–computer Interface Application using PCA and Ensemble of Weighted SVMs. IETE Journal of Research, vol. 64, no. 3, pp. 406–414 (2018). DOI: 10.1080/03772063.2017.1355271
9. Lu, Z., Gao, N., Zhou, W., Yang, J., Wu, J., Li, Q.: A Comparison of Facial P300-speller Paradigm Based on Famous Face and the Familiar Face. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–6 (2019). DOI: 10.1109/CISP-BMEI48845.2019.8965892.
10. Meshriky, M. R., Eldawlatly, S., Aly, G. M.: An Intermixed Color Paradigm for P300 Spellers: A Comparison With Gray-scale Spellers. In: 2017 IEEE 30th International Symposium on Computer-Based Medical Systems, pp. 242–247 (2017). DOI: 10.1109/CBMS.2017.123.
11. Noorzadeh, S., Rivet, B., Jutten, C.: Beyond 2D for brain-computer interfaces: Two 3D Extensions of the P300-speller. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5899–5903 (2014). DOI: 10.1109/ICASSP.2014.6854735.
12. Noorzadeh, S., Rivet, B., Jutten, C.: 3-D interface for the P300 speller BCI. IEEE Transactions on Human-Machine Systems, vol. 50, no. 6, pp. 604–612 (2020). DOI: 10.1109/THMS.2020.3016079.
13. Oralhan, Z.: A New Paradigm for Region-based P300 Speller in Brain Computer Interface. IEEE Access, vol. 7, pp. 106618–106627 (2019). DOI: 10.1109/ACCESS.2019.2933049.
14. Rathi, K., Gomathi, V.: Human Vision Reconstruction Using Brain Activity Profiles. In: 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1–5 (2018). DOI: 10.1109/CCAA.2018.8777542.
15. Rezeika, A., Benda, M., Stawicki, P., Gembler, F., Saboor, A., Volosyak, I.: Brain–computer Interface Spellers: A review. Brain Sciences, vol. 8, no. 4 (2018). DOI: 10.3390/brainsci8040057.
16. Sarraf, J., Vaibhaw, Pattnaik, P. K.: A Study of Classification Techniques on P300 Speller Dataset. Materials Today: Proceedings, vol. 80, pp. 2047–2050 (2021). DOI: 10.1016/j.matpr. 2021.06.110.
17. Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P. H., Schalk, G., Donchin, E., Quatrano, L. A., Robinson, C. J., Vaughan, T. M.: Brain-computer Interface Technology: a Review of the First International Meeting. IEEE Transactions on Rehabilitation Engineering, vol. 8, no. 2, pp. 164–173 (2000). DOI: 10.1109/TRE.2000.847807.
18. Won, K., Kwon, M., Jang, S., Ahn, M., Jun, S. C.: P300 Speller Performance Predictor Based on RSVP Multi-feature. Frontiers in Human Neuroscience, vol. 13 (2019). DOI: 10.3389/fnhum.2019.00261.
19. Yáñez-Márquez, C.: Toward the Bleaching of the Black Boxes: Minimalist machine learning. IT Professional, vol. 22, no. 4, pp. 51–56 (2020). DOI: 10.1109/MITP.2020.2994188.