Automatic Recognition of Error Potentials in a P300-Based Brain-Computer Interface

G. Visconti, B. Dal Seno[1], M. Matteucci[1], L. Mainardi[2]

[1]Department of Electronics and Information, IIT-Unit, Politecnico di Milano, Italy
[2]Department of Bioengineering, IIT-Unit, Politecnico di Milano, Italy

bernardo.dalseno@polimi.it

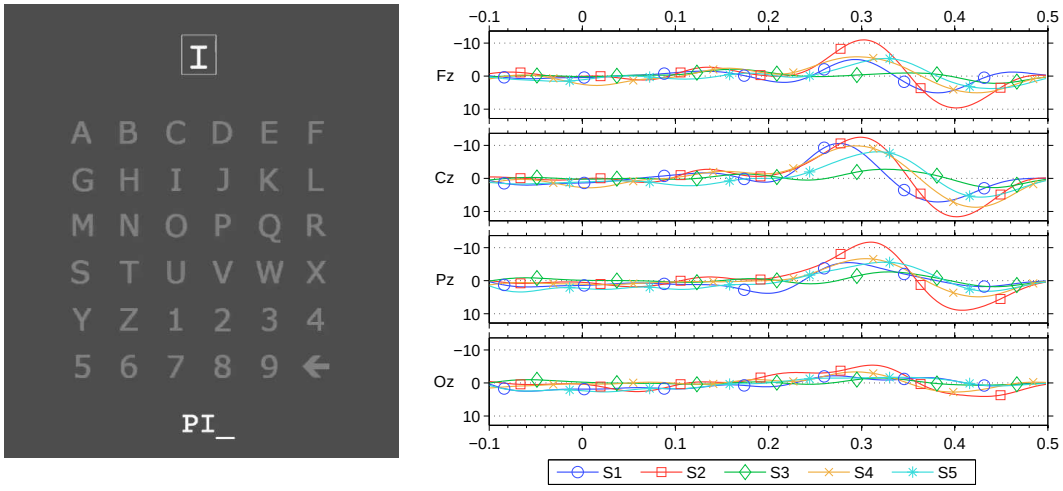
Abstract

An *error potential* (ErrP) is an innate event-related potential generated when a subject makes a mistake, and, more relevant to brain-computer interface (BCI) applications, when the BCI itself behaves differently from the user intent. For this reason, error potentials are nowadays attracting attention in the BCI field and the presence of ErrPs has been studied already in a few BCI paradigms. In this paper we investigate the presence and the detection of error potentials in a P300-based BCI speller similar to the one described in [1], where 36 symbols are disposed on a 6×6 grid, and entire rows and columns of symbols are flashed one after the other in random order. The aim of our research is twofold; first of all, we are interested in developing a method for the automatic detection of ErrPs in a P300 speller, and, secondly, we want to evaluate the real improvement of the performance obtained when ErrP detection is used. Experiments are conducted on five subjects in a controlled scenario, where the outcome of the BCI is actually programmed to generate errors with a 20% probability; users are unaware of this, and they believe to be interacting with a real BCI. Results show that it is indeed possible to recognize an ErrP in an automated fashion when a user interacts with a P300-based BCI, and we also provide a measure of how an automatic error correction based on ErrPs impacts on the overall BCI performance.

1 Introduction

An *error potential* (ErrP) is an event-related potential generated when a subject makes a mistake, and, more relevant to BCI applications, when the machine behaves differently from the user intent. They are known since the late 80s [2, 3] when they were described as a negative shift in the electric potential over the fronto-central region (from Fz to Cz of the 10-20 system) occurring 50–100 ms after an erroneous response (*error negativity* — `Ne` — or *error-related negativity* — ERN) and a subsequent positive shift in the parietal region [2], whose maximum occurs between 200 and 500 ms after the error (*error positivity* — `Pe`). Many experiments have been done, and a high variability in shape, size, and delay of the `Ne` and `Pe` components has been observed. The two components are thought to be the effect of different underlying mechanism, whose nature is not yet certain [4].

Error potentials are obviously attracting attention for BCI applications. In [5] the presence of ErrPs in a BCI paradigm (cursor movement by mu and beta rhythms) was revealed, as a positive peak at Cz 40 ms after the end of erroneous trials. Although the features of this potential are rather different from the ErrP mentioned above, this finding suggests an interesting application: the automatic detection of the errors a BCI makes in recognizing the user's intent. Millán and colleagues worked on this possibility to improve a BCI performance [6]: they made experiments with a simulated BCI, which made an incorrect choice 20% of the times, independently of the user's EEG. They trained a Gaussian classifier to automatically recognize ErrPs reaching an accuracy of about 80%.



(a) (b)

Figure 1: (a) Graphical interface of our P300 speller. (b) Error-minus-correct means for the five subjects; units are seconds and microvolts; time 0 is feedback time.

In this paper we investigate the presence of error potentials in a P300-based BCI speller with two aims: first, the development of a method for the automatic detection of ErrPs in a real BCI, i.e., a P300 speller, and, second, evaluating the possible increase of the real performance obtained when such an ErrP automatic detection is included in the speller. Section 2 of this paper describes our experimental setup, the P300 speller used, and the designed interactions between user and machine; Section 3 describes the methodologies used to identify ErrPs in the experimental data, while the obtained results are shown in Section 4.

2 Experimental Setup

In this study, we used an experimental paradigm similar to the one described by Donchin [1]: 36 symbols are disposed on a 6×6 grid, and entire rows and columns of symbols are flashed one after the other in random order. The grid of symbols (see Figure 1.a) includes letters from the alphabet, digits, and the *backspace* symbol, represented by the small arrow in the right bottom corner.

The spelling of a single letter is divided in *repetitions*, with each repetition being composed of 12 stimulations. A single stimulation is obtained by flashing a row or a column (the intensification lasts 100 ms and the inter-stimulus interval is 100 ms). Each row/column is flashed only once during a repetition, and a series of 5 repetitions is performed. There is no pause between repetitions. At the end of the fifth repetition, the P300 system detects the (hopefully) desired row and column, and selects the letter at the intersection. After a pause of 1 s, the letter is displayed in the rectangular frame visible in the top part of Figure 1.a, and added to the word at the bottom of the screen. The presentation of the letter should elicit an ErrP whenever the letter is different from the user intention. If the error-detection system recognizes an ErrP, it overrides the P300 speller and cancels the last spelled letter. After a 2 s pause, the speller starts a new series of stimulations for the next letter. A single *trial* is therefore composed of 60 P300 stimulations and 1 ErrP stimulation.

In our study, the interaction between user and BCI was conducted in a controlled scenario in order to reach a desired number of errors and easily acquire the experimental *ground truth* without the user knowing it. Subjects were told to pay attention to a given letter at the beginning of each trial, and they were informed that the system would recognize their intention. However, the BCI system was programmed to select the right letter with a probability of 80% and a wrong one with

20% probability, without considering the EEG recordings at all. When a wrong letter was spelled, subjects had to choose the *backspace* symbol in the next trial. An accuracy of 80% was selected because it was considered reasonable for this BCI protocol: this accuracy value is low enough to have a sufficient number of error epochs without frustrating the users or inducing them to think that the BCI was not working.

Five subjects took part in this experiment. EEG data were recorded with an EBNeuro BE Light system at Fz, Cz, Pz, Oz referenced to right mastoid, and EOG with two bipolar electrodes near the right eye, sampled at 512 Hz, in the band 0.1–230 Hz. Each subject participated to three separate recording sessions, separated by a few days or also some weeks. Figure 1.b shows the error-minus-correct means (i.e., the differences between the averages of ErrP epochs and the averages of non-ErrP epochs), filtered in the 1–10 Hz band. Four out of five subjects have a strong `Ne` with a peak at about 300 ms, followed by a proportionate `Pe` some 100 ms after. The response of Subject 3 is rather weak, with peaks of less than 3 µV in absolute value.

3 Signal Analysis and Classification

Recorded data are segmented in epochs ranging from 100 ms before the stimulation instant (feedback onset) to 500 ms after it. Epochs containing strong EOG activity ($> 100\,\mu V$ at any point) are automatically discarded before further analysis. The implication of discarding epochs in a BCI is that in those cases there is no way to correct possibly wrong responses. Obviously, robustness with respect to EOG contamination is a desirable property for a detection algorithm, but we preferred to concentrate first on producing something working, and leave the improving of its robustness for the future. Data are then filtered in the band 1–10 Hz to improve the signal-to-noise ratio by eliminating frequency components extraneous to ErrPs.

On average, a difference between ErrP and non-ErrP epochs is observable only in some intervals of the segmented epoch, and these intervals depend on the subject. For these reasons we developed a way to automatically determine significant intervals in the ErrP for classification. For each channel c and time point t, the signals $s_{c,1}(t)$ from ErrP epochs and $s_{c,0}(t)$ from non-ErrP epochs can be viewed as two sets of random variables. A t-test is used to check if, for any given t and c, the mean of $s_{c,1}(t)$ differs significantly from the mean of $s_{c,0}(t)$; the significance level has been chosen to be 0.01, but much smaller p-values have been often found in analyzing data.

The t-test requires the distributions of the samples under test to be normal with equal variance. The significance of the t-test is not used to classify epochs, but only to find promising time intervals. For this reason, only a preliminary analysis has been done to verify that the t-test preconditions, so as to be sure that the t-test results would be meaningful. Normal probability plots drawn for a subset of time instants have been used to verify normality; although in some cases the tails of the observed distributions are longer than the Gaussian ones, the departure from normality is never dramatic. Equality of variance has been tested by applying the F-test to data from some subjects, and it is verified at a significance level around 0.01.

The points detected by the t-test tend to lie in groups, because the filtered signals have a strong autocorrelation for short lags. However, many intervals of different sizes, with "holes" in between (see the top part of Figure 2 for an example) are usually detected, while we are interested only in finding one contiguous time interval containing all the interesting features of signals. For this reason, a clustering algorithm is run on the time points found by the t-test to fill holes and discard isolated points or small intervals. For this purpose, we used DBSCAN [7], a clustering algorithm based on density, because it clusters together nearby points and ignores outliers, which is exactly what we need.

For our application, a cluster is transformed in an interval by taking the smallest interval containing the cluster; in other words, any gap is filled (see Figure 2). For the DBSCAN parameters, we have chosen $k = 10$ and $\varepsilon = 60$ ms; k defines the minimum number of points in a cluster, so the smallest interval found was $10/(512\,\text{Hz}) \approx 20$ ms. ε defines the minimum distance between two points in different clusters, hence sequences of contiguous points from the t-test less than 60 ms apart are fused together; when `Ne` and `Pe` components are present, they generate (more or less)

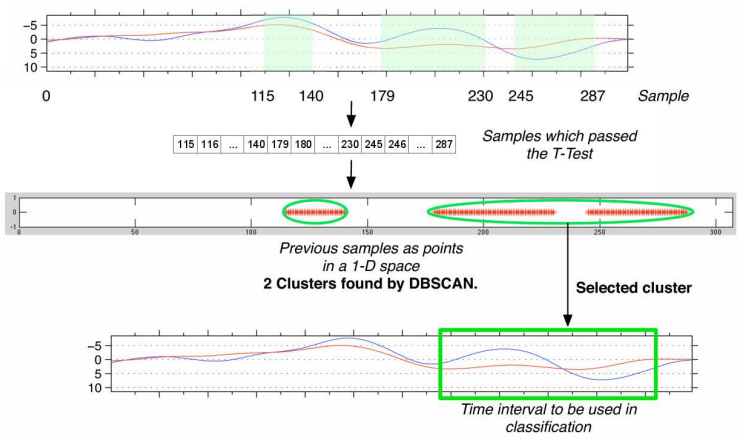


Figure 2: Procedure for the identification of significant intervals. Top: shadowed areas contain the samples that passed the t-test with a p-value of 0.01 or less. Middle: clustering of samples. Bottom: the interval used for classification.

contiguous sequences of points passing the t-test, and these sequences are fused by clustering, as they are closer than 60 ms.

The channels with the most significant interval found by the t-test have been Cz and Fz, in accord with the literature. For this reason, we use only these two channels for automatic epoch classification. As a further simplification, the intervals found by DBSCAN for the two channels are fused together (the minimum interval encompassing both is taken), and used for both channels.

The significant intervals are then used to extract two different kinds of features: raw sample values and coefficients of polynomial approximation. Features of the first kind are just all the samples of the EEG signals falling within the time interval found as previously explained. The second kind of features is computed by fitting (in the least square sum sense) two third-grade polynomials to the EEG signals from the Fz and Cz channels of each epoch; the 8 (4+4) coefficients of both polynomials represent the extracted features. Features are then fed into classifiers.

We used two standard classifiers, linear discriminant analysis (LDA), and k nearest neighbor (k-NN), and two methods from the literature, which were already been applied to detect other event-related potentials: the Bayesian method described in [8], and the SVM-based approach [9]. LDA was applied to both kind of features, while k-NN only to raw samples. The SVM-based method was applied to all the four channels of whole epochs, as a sort of a check that the selection of channels and time interval would not lower the classification performance.

4 Results and Conclusions

Table 1 shows the classification results, as the mean values of recall (fraction correctly classified) for ErrP and non-ErrP epochs obtained with a 3-fold cross-validation scheme applied to the three sessions of each subject. The column labeled *size* shows the number of epochs, either ErrP or non-ErrP, that remained after discarding the most noisy ones followed by their original number (please recall that we discared epochs affected by relevant EOG artifacts). The column labeled *LDA* is for LDA applied to raw samples, while *P. LDA* stands for LDA applied to polynomial coefficients. Except for Subject 3, the best classifiers (SVM and LDA with polynomial coefficients) reach about 80% of recall, which is quite good. The performance for Subject 3, whose responses in Figure 1.b are the weakest, is just slightly better than random.

It should be evident that a BCI elicits ErrPs and it is possible to automatically detect them. The question is, can ErrP detection improve the overall performance of a BCI? We estimated the impact of ErrP detection by modeling how this affects the performance of the BCI used in this study.

Subject		Size	LDA	Bayes	*k*-NN	P. LDA	SVM
S1	ErrP	90 / 92	72%	74%	63%	77%	84%
	N-ErrP	440/450	86%	75%	88%	87%	82%
S2	ErrP	83 / 90	64%	72%	60%	72%	70%
	N-ErrP	384/426	83%	81%	85%	84%	84%
S3	ErrP	56 / 97	50%	46%	50%	54%	52%
	N-ErrP	284/477	68%	55%	64%	67%	56%
S4	ErrP	89 / 92	69%	65%	71%	69%	79%
	N-ErrP	352/450	86%	75%	87%	83%	84%
S5	ErrP	41 / 88	61%	73%	63%	80%	71%
	N-ErrP	245/443	85%	76%	86%	91%	80%

Table 1: 3-fold cross-validated results of ErrP detection in P300-speller tasks

We considered the case where the user selects the *backspace* command in the letter grid to correct all the errors made by the P300 speller, and we computed the expected number of trials needed to correctly spell a letter. We derived a formula for this (the derivation is based on the modeling of the BCI as a simple stochastic process that is beyond the scope of this paper):

$$t_{\mathrm{L}} = \frac{1}{p \cdot r_{\mathrm{C}} + (1-p) \cdot r_{\mathrm{E}} + p - 1}\,, \tag{1}$$

where p is the accuracy of the selection of the letter, i.e., the BCI accuracy, r_{E} is the recall of ErrP epochs, and r_{C} the recall for non-ErrP epochs (i.e., correct letters). To be fair, the recall values in Table 1 cannot be used directly, as they are calculated after throwing away some epochs. If we treat discarded epochs as if they were classified as non-ErrP, then we have

$$r_{\mathrm{E}} = f_{\mathrm{E}}\, r'_{\mathrm{E}} \qquad\qquad r_{\mathrm{C}} = 1 - f_{\mathrm{C}}\,(1 - r'_{\mathrm{C}})\,, \tag{2}$$

where f_{E} and f_{C} are the fraction of epochs kept, and r'_{E} and r'_{C} are the recall values computed on the epochs kept.

The improvement in BCI given by ErrP detection can be measured as the reduction in the number of trials needed to properly spell a letter; Figure 3 shows the expected number of trials needed to spell a letter correctly, as a function of the precision p, for the five subjects of the study. We used Equations (1) and (2) applied to the results obtained with the LDA classifier applied to polynomial coefficients, which is a reasonably simple system with good performance. The dashed red line is a reference that represents the expected outcome with no automatic error correction ($r_{\mathrm{E}} = 0$, $r_{\mathrm{C}} = 1$). While Subject 3 seems not to get any benefit from automatic error detection, the other subjects can benefit from it when the accuracy of the P300 speller is no more than 75%.

We have described a procedure to recognize ErrPs in an automated fashion when a user interacts with a P300-based BCI. Our results are encouraging, but further work is needed to improve the performance. First of all, epochs containing EOG activity were excluded from the analysis, and robustness and validity of our method against EOG contamination should be tested. In addition, we limited the search of ErrPs in a 0.5 s window after the stimulus, but longer windows could be explored as done in previous research [10]. Nevertheless, the obtained results show the feasibility of detecting ErrPs, and we quantitatively show how the automatic error correction based on ErrPs may impact the overall BCI performance.

Acknowledgments

This work has been partially supported by the Italian Institute of Technology (IIT), and by the grant "Brain-Computer Interfaces in Everyday Applications" from Politecnico di Milano and Regione Lombardia. We are thankful to EBNeuro S.p.A. for the use of their BE Light EEG system and their support. We are also thankful to all the participants in our study.

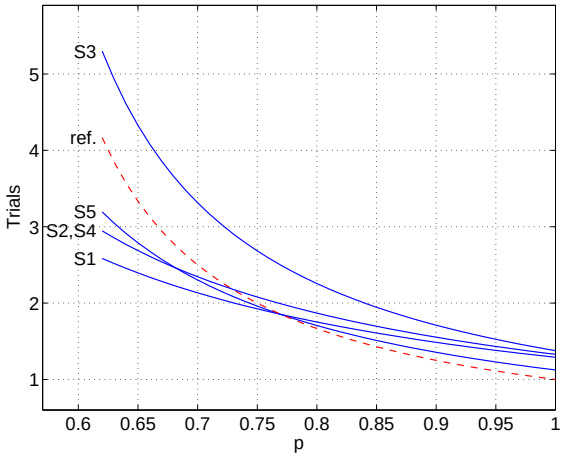


Figure 3: Trials per letter in a P300 speller with error correction vs. accuracy of the base speller classifier. The lines for Subjects 2 and 4 almost coincide and have been collapsed.

References

[1] E. Donchin, K. M. Spencer, and R. Wijesinghe. The mental prosthesis: Assessing the speed of a P300-based brain-computer interface. *IEEE Transactions on Rehabilitation Engineering*, 8(2):174–179, 2000.

[2] M. Falkenstein, J. Hohnsbein, J. Hoormann, and L. Blanke. Effects of crossmodal divided attention on late ERP components. II. Error processing in choice reaction tasks. *Electroencephalography and Clinical Neurophysiology*, 78(6):447–455, 1991.

[3] W. J. Gehring, B. Goss, M. G. H. Coles, D. E. Meyer, and E. Donchin. A neural system for error detection and compensation. *Psychological Science*, 4(6):385–390, 1993.

[4] M. Falkenstein. ERP correlates of erroneous performance. In M. Ullsperger and M. Falkenstein, editors, *Errors, Conflicts, and the Brain. Current Opinions on Performance Monitoring*, pages 5–14, 2004.

[5] G. Schalk, J. R. Wolpaw, D. J. McFarland, and G. Pfurtscheller. EEG-based communication: presence of an error potential. *Clinical Neurophysiology*, 111(12):2138–2144, 2000.

[6] P. W. Ferrez and J. d. R. Millán. EEG-based brain-computer interaction: Improved accuracy by automatic single-trial error detection. In *Advances in Neural Information Processing Systems 20*, pages 441–448, 2007.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon, USA, 1996. AAAI Press.

[8] J. Kohlmorgen and B. Blankertz. Bayesian classification of single-trial event-related potentials in EEG. *International Journal of Bifurcation and Chaos*, 14(2):719–726, 2004.

[9] M. Kaper, P. Meinicke, U. Grossekathoefer, T. Lingner, and H. Ritter. BCI competition 2003 – data set IIb: support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engineering*, 51(6):1073–1076, 2004.

[10] P. W. Ferrez and J. d. R. Millán. Error-related EEG potentials generated during simulated brain-computer interaction. *IEEE Transactions on Biomedical Engineering*, 55(3):923–929, 2008.