EEG Signal Categorization Performance: Influence of ERD position within classification window

N. Boom
University of Twente, Postbus 217, 7500 AE Enschede, The Netherlands
n.boom@student.utwente.nl

ABSTRACT
Much research has been done in the field of cue-paced movement (imagery) classification performance. But cue-paced classification greatly narrows BCI bandwidth and possibilities. This research aims to uncover the extent to which signal classification performance is lost when the Event Related Desynchronisation (ERD) is not placed optimally in the classification window. This also denotes the primary difference between cue- and self-paced BCI. In self-paced situations cues are not known, resulting in windows where ERD placement differs every sample.

Raw electroencephalography data was obtained from one test subject and was categorized for several different window sizes and ERD placements. This was achieved by sliding the classification window backwards and forwards in time. Markers were used as a reference point.

We found that the placement of windows during training has great influence on classification performance. Larger windows seem to result in higher overall performance but are prone for multiple hits per ERD. We also found that classification performance rises steeply when more bandwidth power of the ERD enters the classification window, but rises steeper for smaller windows. When classification windows are slid further past the marker performance starts to drop, but drops slower than it rises.

Keywords
BCI, Self-paced, asynchronous mode, cue-paced, cue-based, synchronous mode, timeframe, time window, signal categorization, categorization performance

1. INTRODUCTION
Brain-Computer-Interaction (BCI) can be seen as an input device. When a living animal thinks, or has the intent to perform a given action there are slight differences in the creatures brain activity. These changes can be measured via non-invasive methods such as electroencephalography (EEG), magneto encephalography (MEG), positron emission tomography (PET) and functional magnetic resonance imaging (fMRI), and optical imaging. [17] Or more invasive methods such as implanted EEG sensors as used in a study of Chapin [3].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

12th Twente Student Conference on IT, January 22, 2010, Enschede, The Netherlands.
Copyright 2010, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

EEG via non-invasive sensors measures brain-potentials that breach the skull. One of the features visible in those signals is an Event Related Desynchronisation (ERD) (bandwidth power decrease) followed by a rebound (Event Related Synchronisation: ERS). Movement imagery is visible around 10Hz and 20Hz frequencies in an averaged frequency spectrum [9] and is invoked when one intends to move. Pfurtscheller showed in 1978 that this phenomenon is visible in most people: “In the current study, mu activity (...) was present in all 10 investigated subjects.” [11]. The human brain emits nearly the same ERD/ERS activity change whilst executing actual or imagery movement, but error rates are lower using actual movement [7]. Lemm [9] and Pfurtscheller [10] showed the ERD is invoked rapidly, but the ERS can take up to several seconds.

“BCI operation could be either in synchronous (or cuedriving) mode where mental states alter according to predefined external stimulus or cue, or in asynchronous (or intermittent) mode where the subjects initiate the intent of control,” [16]. In Figure 2a we see a cue-paced BCI with four cues. Around these cues windows are placed resulting in four classifications. In Figure 2b we see a self-paced BCI. All data is classified and windows often overlap resulting in more classifications. These classifications can be translated to true or false positives and negatives (TP / FP / FN / FN).

Cue paced BCI often incorporates a 3 part epoch: First a stimulus (cue) is given to undertake action. After a few milliseconds (response time) data is checked to classify the movement (See Figure 2a). After the classification there is a refractory period of multiple seconds before the epoch starts again. This gives time for the ERS noise to disappear from the EEG signal. The cues are stored together with the EEG data. Self paced BCI uses many overlapping time windows (See Figure 2b). In each of these windows the algorithm checks if a classification can be made. Here no cues are saved.

This research aims to uncover the extent to which signal classification performance is lost when the ERD signals are not placed perfectly in the window. This also denotes the primary difference between cue- and self-paced BCI. In self-paced situations cues are not known, resulting in windows where ERD placement differs every sample and could lay on the edges of the window.

Figure 1. Fictional recording. A cue-paced algorithm will only categorize the grey blocks. A self-paced algorithm will categorize the entire stream.
Most real-life situations need such rest moments. Scherer [13] which BCI could be used. But 2-class cue-paced BCI cannot be interaction problem and thus extends the set of situations in introducing a 3-class BCI (rest and two different movements cue-less paradigm.

Thus signal categorization development should also focus on a timing and speed of communication are preset by the paradigm. synchronized (or cue-based) operation which means that the task was to perform imagery left hand, right hand, foot or tongue movements according to a cue" [1]. The results of this 4-class BCI can however be seen as 4 states that all deliver an ERD event. The second class in our 2-class BCI is designed to choose left or right based on the data in the window. Whereas the self-paced BCI must not only choose left or right (on basis of which ERD is higher) but also detect if there is no ERD at all. This introduction of the third class changes the way we denote true or false positives and negatives and thus results in incomparable values.

To make self-paced and cue-paced methods comparable we suggest comparing a 2-class cue-paced BCI with a 2-class self-paced BCI with the same classes (movement-class versus rest-class). We do not, however, take the problem of overlapping windows and thus generating multiple hits into account. This is a self-paced BCI problem for which the solution is related to the usage of the input method. Triggering multiple hits will not pose a problem in some usages.

2. PROBLEM STATEMENT

We wish to research if, and to what extent, using classification windows around known cues contribute to signal classification performance. I.e. the influence of "knowing cues" in classification algorithms on overall categorization performance. Especially focused on the location of the ERD within, or the absence of such an event. We expect that the results from this project can improve performance for CSP based BCI classifiers. Qin [15], Want [16] and Kamousi [6] show that a classification performance of 80% can be achieved in a 2-class BCI using non-invasive EEG sensors in a controlled environment. This was achieved with experiments using a visual cue: a stimulus to start the action being classified. The signal classification algorithm then uses a window around this cue as the data to analyze. Many other experiments have used the same technique.

Due to the lack of a rest-class, cue-paced classification methods might not be applicable in the real world. A rest-class gives the user the possibility to withhold action when none is wanted. Most real-life situations need such rest moments. Scherer [13] states "The majority of BCIs used these days are designed for synchronized (or cue-based) operation which means that the timing and speed of communication are preset by the paradigm. This, however, does not represent a natural way of interaction." Thus signal categorization development should also focus on a cue-less paradigm.

Introducing a 3-class BCI (rest and two different movements like left or right hand/foot movement) solves the natural interaction problem and thus extends the set of situations in which BCI could be used. But 2-class cue-paced BCI cannot be compared to 3-class self-paced BCI. In a 2-class BCI classification performance of a random algorithm is 50% if class occurrences are evenly distributed (2 options of equal chance). In a 3-class BCI this random chance is approx. 33%. Furthermore the introduction of the third class changes the way we denote true or false positives and negatives, thus resulting in incomparable values. Also a 2-class classification system is designed to choose left or right based on the data in the window. Whereas the self-paced BCI must not only choose left or right (on basis of which ERD is higher) but also detect if there is no ERD at all. This introduction of the third class changes the way we denote true or false positives and negatives and thus results in incomparable values.

To make self-paced and cue-paced methods comparable we suggest comparing a 2-class cue-paced BCI with a 2-class self-paced BCI with the same classes (movement-class versus rest-class). We do not, however, take the problem of overlapping windows and thus generating multiple hits into account. This is a self-paced BCI problem for which the solution is related to the usage of the input method. Triggering multiple hits will not pose a problem in some usages.

2.1 Related work

Although there has been an extensive search, no research was found comparing experimental cue-paced performance to self-paced performance. Nor have we found classification performance studies involving classification window ERD placement.

Promising research was found about the amount of true positive and false positive classification in asynchronous algorithms. Sherer [13] states that in a population of three healthy test subjects with extensive cue based feedback training: "trained LDA classifiers achieved classification accuracies (10 × 10 cross-validation) of 77%, 84%, and 78% for subjects s1, s2, and s3, respectively." This shows that reliable self-paced BCI classification techniques are currently available. Townsend [14] however shows deferring numbers: True Positives (TP) up to 80% with False Positives (FP) between 10 and 77% for right imagery.

This is further endorsed by the results of a 4-class BCI assignment of the third BCI Competition. 63% to 79% efficiency was achieved. "The task was to perform imagery left hand, right hand, foot or tongue movements according to a cue" [1]. The results of this 4-class BCI can however be seen as 4 states that all deliver an ERD event. The second class in our 2-class BCI must however represent a rest class.

3. METHODS

Simplifying the problem to a 2-class problem can give results that are not applicable in all multi-class self-paced BCI systems. But due to reasons mentioned in Section 2 we will use a 2-class BCI classification algorithm. The classes used are one movement class and one rest class. Comparing the classification performance must be done in a manner that allows for the use of different thresholds.

The primary difference (See Figure 2) between the cue- and self-paced algorithms is that cue-paced algorithms know what part of the data must be analyzed and is thus able to place the window so that the ERD event is placed optimal within it. Self-paced BCI’s must determine themselves if what they see is an entire ERD event, a partial ERD or contains no ERD at all. Allowing alternative thresholds is in favour of classification performance.

Figure 2. In (a) we see a cue-paced classifier. In (b) We see a self-paced classifier with overlapping windows.
3.1 Dataset
Event Related Desynchronisation (ERD), followed by Event Related Synchronisation (ERS) resemble actual or imagined movement and are a voluntarily produced signal which can be detected via EEG. Currently the department of HMI has an ERD/ERS pipeline readily available which can be used. A readily available self-paced 3-class raw EEG dataset was used: "cued motor imagery with 2 classes (left hand, foot)" [4]. The data was retrieved from a single test subject, making cross-referencing impossible. Only the training data of this dataset is used.

"The recording was made using BrainAmp amplifiers and a 128 channel Ag/AgCl electrode cap from ECI. 118 EEG channels were measured at positions of the extended international 10/20-system." [4] The training dataset is a 35 minute long 1000Hz recording with 115 left hand and 115 right foot movement markers. We chose to use the 100Hz set for practical reasons: A 100Hz dataset computes faster. Since movement imagery ERD’s and ERS’s are visible in the 8-30Hz frequency range so no noticeable negative effects related to the 100Hz data should be present in the results. This dataset was generated by taking each 10th sample of the 1000Hz original dataset.

Only a 2-class dataset was needed but this 3-class dataset meets all requirements and is very long. The redundant right foot movement data in the continuous EEG data can be used as a control value, comparing the results of left hand vs. rest to the results of right foot vs. rest.

Movement markers are placed at irregular intervals. The mean is 9.73 seconds. The smallest interval measures 5.23 and the largest measures 71.19 seconds.

For conclusions concerning larger window sizes or the placement of windows relatively far away from its marker the results will be measured over less samples. We do not want to bias results so time is needed for the ERS to disappear from the EEG signals and the length of the required rest data is larger than most gaps are.

In the BBCI dataset no markers are present to mark rest periods. Therefore we added these manually. We will test with several implementations of the rest markers.

(a) 207 rest markers placed exactly between all movement markers
(b) 75 rest markers placed exactly between two markers if the gap is larger than 10 seconds
(c) 31 rest markers placed exactly between two markers if the gap is larger than 16 seconds

The markers are divided evenly over the dataset. The IVb dataset of the third Berlin BCI competition [4] can be found on the Berlin BCI Group website.

3.2 Classification algorithm
The classifier was programmed by B. Reuderink from the University of Twente. The code is written in python 2.6 and is available under the New BSD Licence on googlecode.com. Golemml\(^1\) is a modular machine learning library. Psychic\(^2\) is an extension to Golem specialized in Brain-Computer Interface related Machine Learning Nodes. Both modules rely heavily on NumPy, SciPy and Matplotlib.

First the raw EEG data is extracted from ASCII data files provided by the BCI Competition and loaded into Golem. All 118 channels were band pass filtered using an IIR filter to isolate the 8-30Hz frequency range in which the ERD occurs.

Using spatial filters, variance produced by mental activities such as imagined movement can be extracted from this feature selection [12]. We used the Common Spatial Pattern (CSP) to extract the spatial filters. Linear Discriminant Analysis (LDA) was applied to make a final prediction. All 118 sensors were used, but the spatial pattern probably relied heavily on a few sensors as the movement imagery ERD originates in the motor cortex and sensors closed to that region will react more discriminate.

3.3 Sliding the windows
Knowing when action was taken is one of the most important differences between cue-paced and self-paced BCI paradigms. Knowing these cues enables the classifier to place the classification window over the entire ERD. In the BBCI data we do not have cues, but we do have the true-labels data equivalent of cues: Markers. Markers are vectors that store the time that the movement was undertaken and what type of movement it was. Markers are normally used to verify classification results.

In self-paced settings the windows are not always aligned with the markers since cues are unknown. This misalignment is visible in Figure 2b and Figure 3. Figure 3 illustrates the variables differentiating cue-paced from self-paced BCI.

![Figure 3](image.png)

We wish to research the influence of the window position relative to the ERD event. This means moving the relative position of the ERD by sliding the window in time. This can be achieved by changing the offsetDelta variable as denoted in Figure 3.

3.3.1 Length of window (windowSize)
We constantly look at small subsets of the dataset to classify the data. With our dataset each subset (i.e. window) can result in a

(a) Rest-class classification, or
(b) Movement imagery classification.

We chose to take the length of the window as a variable. As discussed in 3.1 Dataset the minimum time between two movement markers in our dataset is 5.23 seconds. Any window length value larger than that will result in windows containing more than one marker during the sliding. We also need dwell time after each marker to allow for ERS and signal refactoring. Window sizes larger than 3.0 seconds are not tested. Larger windows sizes are not common in self-paced BCI.

3.3.2 Window offset (offsetDelta)
The offsetDelta contains a value in the range

\[
[-\text{MAX}(\text{windowSize}) ; \text{MAX}(\text{windowSize}) + C]
\]

---

1 Code available from http://code.google.com/p/golemml/
where C is a constant that we chose at 2x frame rate. So the range is \([-280; 475]\) frames with a step size of five frames and a frame rate of 100. The result is that the window is placed both before, over and after the ERD, measuring all possible placements. Both movement and rest class windows are slid via this offsetDelta.

3.4 Methods used for data analysis
For each trainOffset, windowSize and offsetDelta an event-by-event analysis is undertaken. The goal is to define the number of true-positives and false-positives generated by the classifier for each combination of the three variables. Townsend [14] showed that such a event-by-event analysis can evaluate the performance of a BCI. Our evaluation matches the criteria Townsend gives as requirements for valid ROC analysis.

Self-paced and cue-paced paradigms use different thresholds, forcing us to use an analysis method that is independent of the used threshold. A ROC shows True Positive Rate vs. False Positive Rate for a binary classifier [5]. When classification thresholds go down, less False Negatives but more False Positives will be generated. Varying the threshold and plotting them in a ROC curve gives insight in the best threshold value for each of the two paradigms. The Area Under the Curve (AUC) determines the performance of the classifier [14].

4. RESULTS

Results are divided into three sections. In the first section we present results related to the placement and size of training windows. In the second part we present results found for test windows. In the third part results regarding the actual sliding of test windows are presented.

All results were found inspecting left hand movement imagery versus rest classes. Afterwards all statements were verified by plotting the same data for the right foot movement imagery versus rest classes of the same dataset. We found that right foot movement imagery results in lower AUC values but are still comparable. Calculated AUC values are 5 times 5-fold cross-validated.

In the following sections we often refer to images in Appendix A. Appendix A can be found on the last page of this paper and shows left hand imagery versus rest classification performance results measured in AUC. The graphs are top views and the red or green colour indicates bad or good performance. Every row of graphs denotes a training offset and shows the performance achieved with the several implementations of the rest markers. AUC values were calculated for window sizes of 0.70, 0.80, ..., 2.70, 2.80 seconds and for window offsets of -2.80, -2.75, ..., 4.70, 4.75 seconds.

Figures A, D, G and J in Appendix A show results for the scenario where rest-markers were added between all movement markers. The AUC values peak at other windowOffsets than the 10s and 16s equivalents do. This is especially visible in Figure D within the highlighted area (d). There we find the highest AUC values at positions where we expect to find the lowest, and vice versa. We therefore believe the data has been corrupted due to the small window sizes and will not be used in further analysis.

In Figures H and K in Appendix A we see a small distortion near the windowSize of 220 frames. This distortion was also found in right hand movement versus rest. The distortion is not visible in all trainOffsets or rest-marker options and we cannot find a relation that explains the occurrence. We expect that these distortions are generated by the participant and would not occur if data was averaged over multiple subjects.

4.1 Training windows
The training uses a window around the marker with respect to the following formula:

\[
\left[ -\frac{1}{2} \times \text{windowSize} + \text{trainOffset} ; \right. \\
\left. \frac{1}{2} \times \text{windowSize} - \text{trainOffset} \right]
\]

This means that if trainOffset equals zero, half of the data is taken before the marker and half is taken after the marker. If only half of the ERD bandwidth power would occur in the specified window this would have drastic consequences in the classification performance. The threshold would be chosen much lower than needed, especially in the smaller windows sized scenarios.

Results of classification performance with various trainingOffsets are shown in Figure 4 and 5. Figure 5a with trainOffset zero shows a classification performance near random for most window sizes. With a trainOffset of +0.5 in Figure 5b much higher AUC values were found.

In Figure 4 we see an upward trend in classification performance until T40 (+0.4 seconds). After T60 (+0.6 seconds) this smoothes out. The results of Figure 4 are plotted in more detail in Figure 5. We see that especially in smaller windowSizes large performance differences are visible between trainOffsets smaller than +0.5 seconds.

4.2 Window sizes
In the used classifier it is necessary to keep the training windows and test windows the same length. Thus training with
smaller or larger windows than the test window size is impossible.

We found that smaller window sizes are more prone for the influence of trainOffsets. Figures B and C in Appendix A show low AUC values for smaller window sizes while the larger window sizes (2 sec. and up) show acceptable values. In Figures G, H and I in Appendix A the differences between small and large windows are less noticeable. Figure A in Appendix A shows AUC values of 0.2. It seems the classifier classified movement as rest and rest as movement. This also explains why results improve when trainOffsets increase.

In Appendix A, Figure 1 where rest markers were added in gaps larger than 16 seconds. Line (b) shows the highest AUC values for each window size. At this windowOffset the ERD is best positioned within the window because the highest performance is achieved. Such a line can also be drawn in E, F, H, L, N and O. We did not find differences between training offsets or window sizes in the placement of (b).

In Figure F of Appendix A we added block C, surrounding a small setback in performance. The distortion is visible in E, H, I, K and L suggesting it is related to window size. In Figure 7 we see AUC's lowered by 0.10. It is unclear what the cause of this performance drop is. Figure 7 shows the minimal (MIN), maximal (MAX) and averaged (AVG) value of AUC's shown in Figure 6. The minimal (RMIN) and maximal (RMAX) values of right foot versus rest show that they give lower performance.

4.3 Sliding test windows

As depicted in Figure 6 and Figure 7 performance reacts to window offsets. Window size influences the shape of the windowOffset vs. AUC plot as shown in Figure 6. Window sizes are denoted by all separate lines in Figure 6, and are shown in 3D in Figure 8.

We see that classification performance rises quickly when the window is placed (partially) over the ERD event. After 150ms performance starts to subside due to the ERS rebound. The shape of the graph is comparable for all window sizes but the window size stretches the shape. Larger window AUC values rise less quickly and remain higher for a longer period of time. Smaller window AUC values rise and subside relatively quickly.

5. CONCLUSIONS

Using an external dataset poses the risk that it was not recorded properly. In this case the dataset was available for the BCI Competition and we believe it was of good quality. The used
dataset did not contain rest class markers and these were added manually. We chose to add them in three different settings depending on the amount of time between two markers. As discussed in Section 3.1 we created three different datasets. the one where rest markers were added between all movement markers will give results where the signal used for classification is polluted by the ERD/ERS of the surrounding movements. Whereas the dataset, where rest markers were added if the time between two movement markers was more than 16 seconds, did not have this problem. Because results were 5 times 5-fold cross-validated this resulted in only 6 rest classes versus 22 movement classes in the test segment. We feel that the dataset that added rest markers in spaces 10 seconds or longer gives best reliable results. Values on the outer edges of window sliding offsets might still be biased.

Found results were compared to values found using the foot movement versus rest case. This data was extracted from the same dataset. We found that this configuration gave lower AUC values. This is due to origins of the foot-movement, which is placed deeper inside the brain and is thus harder to measure. Found results were present in both right foot as left hand versus rest movement.

We found that the placement of windows relative to ERD events is extra important in the training of the classifier using relatively small window sizes of less than ~2 seconds. If these windows are not correctly aligned with markers it has noticeable effects on performance. Overall performance measured by the Area Under the Receiver Operating Characteristic Curve can drop from good to random values when a 1.2 second window is slid -0.5 seconds. Larger windows have less need for accurate markers.

This is particularly interesting for cue-paced BCI settings where one tries to improve bit-rates by creating smaller epochs. A logical step is to use smaller classification windows. Whilst training a classifier, or when classifying offline, it is recommended to experiment with placing all markers tenths of seconds forwards and backwards in time and calculate which value results in the highest classification performance.

With a properly trained classifier we found that AUC optima (highest values) were related to the window sizes. Smaller window sizes result in lower values (up to -0.09). If minimizing latencies is more important than error rates one should consider using a smaller window size. First of all the optimum AUC values are slightly further away from the marker when the window size grows larger. Secondly, smaller windows require less 'future data', resulting in a quicker classification. We found a larger difference between the peak (line b in Appendix A, Figure I) and the after-peak (near 250 in Figure 7) period we expect -but cannot verify- denotes the ERS. This means there is less chance of detecting a ERS as a ERD in smaller windows and minimizes multiple hits.

The ideal window length is interconnected with the BCI-usage. We have shown however that window placement is important both in train and test phases. In Figure 7 we show how our CSP based classifier responds to the changes ERD and ERS have on the bandwidth power.

In self-paced settings windows will overlap to some extent. To minimize the generation of multiple hits a high threshold is needed. Larger window sizes give higher AUC values but they remain at this higher level for a longer period of time. This is depicted by the bell-shaped graph in Figure 7 and will result in multiple hits. Using larger window sizes also strains the computers resources more but we have yet to see a classifier that uses all the resources of a modern computer when used online.

Given an AUC threshold, we can determine the optimal overlap of windows. Figure 8 and Figure 7 show the time AUC values remain above the threshold. Once this value is obtained we suggest window overlap of more than 50%. If windows overlap less than 50% it will not generate enough (True) Positives to achieve the requested AUC value. Since more overlap will result in more classifications of one event an overlap of more than 75% should not be needed.

6. FUTURE WORK

This research takes an in-depth look at one specific classification algorithm and the data used to perform this study was retrieved from only one subject. Further research is needed to show if other participants generate comparable results.

The results and conclusions were based upon a single dataset. Conclusions drawn may be inaccurate for some or most other datasets. Although we did compare the results to the third class (right foot), recording more test data from several participants will improve the quality of the research results. The extra data can confirm if the results found in this study are participant-independent and thus applicable for CSP based BCI in general.

We have detected an interesting deviation in the results which we could not be explained with the current data. This includes the 0.1 AUC performance drop in small windows, approximately 2 seconds after the marker (See Section 4.2 and See Figure F, shape c in Appendix A). We hope future research can show if this appears in all test subjects, or that it is produced by one of the variables. The research can be placed in a larger setting of comparing cue- and self-paced BCIs. This would result in a framework that shows when the different paradigms should be used.

7. ACKNOWLEDGMENTS

We would like to give special thanks B. L. A. van de Laar MSc and D. Plass-Oude Bos for getting us started properly and being available for questions and feedback through the entire timeframe of this research. We would also like to acknowledge B. Reuderink for sharing his in-depth knowledge of signal classification and the used algorithms. He wrote most of the code of the Golem and Psychic projects on which this research is based.

8. REFERENCES


Laar, B.L.A., Oude Bos and D., Reuderink, B. Heylen, D. Actual and Imagined Movement in BCI Gaming. University of Twente.


APPENDIX A: CLASSIFICATION PERFORMANCE

Left hand imagery versus rest classification performance results measured in AUC. Every row denotes a training offset and shows the performance achieved with the several implementations of the rest markers. Graphs are a top view of 3D plots and are slightly distorted to show the 3D effect.