

Detecting Error-Related Negativity for Interaction Design

Chi Thanh Vi

Department of Computer Science
University of Bristol
vi@cs.bris.ac.uk

Sriram Subramanian

Department of Computer Science
University of Bristol
sriram@cs.bris.ac.uk

ABSTRACT

This paper examines the ability to detect a characteristic brain potential called the Error-Related Negativity (ERN) using off-the-shelf headsets and explores its applicability to HCI. ERN is triggered when a user either makes a mistake or the application behaves differently from their expectation. We first show that ERN can be seen on signals captured by EEG headsets like Emotiv™ when doing a typical multiple choice reaction time (RT) task – Flanker task. We then present a single-trial online ERN algorithm that works by pre-computing the coefficient matrix of a logistic regression classifier using some data from a multiple choice reaction time task and uses it to classify incoming signals of that task on a single trial of data. We apply it to an interactive selection task that involved users selecting an object under time pressure. Furthermore the study was conducted in a typical office environment with ambient noise. Our results show that online single trial ERN detection is possible using off-the-shelf headsets during tasks that are typical of interactive applications. We then design a Superflick experiment with an integrated module mimicking an ERN detector to evaluate the accuracy of detecting ERN in the context of assisting users in interactive tasks. Based on these results we discuss and present several HCI scenarios for use of ERN.

Author Keywords

Electroencephalography; EEG; Brain Computer Interface; Error Related Negativity; Flick; User interface

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces;

INTRODUCTION

The availability of off-the-shelf EEG headsets like Neurosky™ and Emotiv™ has opened possibility of exploring a whole range of BCI applications for everyday use. These low-cost (typically ranging from \$100 to \$300) and portable devices are being used in a wide range of application including task classification [16], gaming [13], and recognizing user's intended selection [1]. To date, most of these applications have been based on using P300 (a peak elicited at about 300ms after an event onset) which

can be considered as a composite index of both attention and memory [16]. Another form of signal that could be used but is often ignored in interactive applications is the Error-related negativity (ERN).

An ERN is a form of an Event Related Potential (ERP) that can be triggered in the brain when a user either makes a mistake or the application behaves differently from her expectation. This pattern is produced in a person's brain when she is aware of the obvious error(s) that s/he has made; either through system feedback or individual realization [8]. For example, ERN would be produced when pressing the LEFT key while intending to press the RIGHT key in a multiple choice RT task. It also appears, but with lower amplitude when a person is confused about the decision that s/he has made [3]. Usually these ERN signals appear and peak within 150ms of the committed action [9].

If ERN signals can be detected during an interactive task they can be used in detecting and correcting errors or in augmenting users' experiences in those activities. An ERN detection module provides another medium for HCI designers to access users' intentions, which is intuitive and directly from the users' brain. This has great potential in many types of interactive application such as in gaming, spatial navigation tasks and aiding object selection. For example, when ERN is detected the system can prompt the user to check if the selection is the intended target.

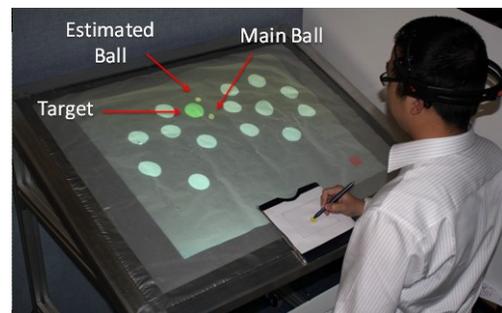


Figure 1. A user performing a Superflick [19] while wearing an off-the shelf EEG headset to see if ERN detection can aide in an object selection task.

However it is difficult to detect a clear ERN pattern due to noisy EEG signals and lack of effective real-time algorithms. Most research in detecting ERN is focused on being able to detect it over an average of multiple trials relying on offline methods (see [3, 17] for examples). As ERN is known to appear when using multiple-choice RT tasks, most research is done on Flanker task [6]. Flanker

task is a visual experiment where the participant has to respond to a central and directed symbol that is surrounded by distracting symbols. There are very few ERN detection algorithms that work in real-time as well as on multiple-choice RT tasks that are not Flanker tasks.

Additionally, most ERN experiments are performed using expensive, non-portable sensing devices such as NeuroScan systems from NeuroScanTM; g.BCIsys from g.tecTM or ActiveTwo from BiosemiTM. These devices have the benefits of not only more sensing channels (up to 256 electrodes therefore can cover all channels in the 10-20 international system), but also higher and user selectable sampling frequencies of up to 16 kHz. These systems also introduce less noise in the signals because of better sensing electrodes and integrated amplifier/ converter module.

In comparison off-the-shelf EEG headsets have at best 14 channels, a sampling frequency of about 1 kHz and introduce more noise due to wireless transfer of EEG signals. Due to these limitations it is not clear if these devices can capture ERN patterns. However, due to their price and portability their use is being increasingly explored within HCI. In this paper we investigate the appropriateness of these off-the-shelf EEG headsets in detecting ERNs and also the effectiveness of real-time algorithms in doing so on a single trial and in real-time.

We first describe an online single trial ERN detection technique that is verified using data acquired from the frontal-central cortex of the human brain. We then show that we can detect ERN online on a single trial in an object selection task. This demonstrates the abilities of harnessing ERN in interactive applications in office conditions. Both our experiments show that we can detect ERNs using the Emotiv Headset with an accuracy of up to 70%.

These rates are indicative of the type of accuracy one can expect from off-the-shelf EEG sets. Improvements in learning techniques can improve this accuracy but it is unlikely that detection accuracy will reach 100%. It is therefore expected that with high detection accuracy users might become over-resilient on a system detecting ERNs which can increase the cost of recovery from the error. In order to examine this issue we conducted a final study where we compare users' error-rates when performing Superflick [19], a modified pointing task, with different potential ERN success rates. The results of the experiment show that ERN detection rates of 65 to 80% are acceptable to interactive applications. We finally discuss the implications of our results to interactive applications.

The contributions of this paper are: (a) we demonstrate that off-the-shelf EEG devices such as the Emotiv headset can measure ERN from channels in the front-central part of the brain; (b) we demonstrate that it is possible to detect these ERN signals online (as opposed to offline) from a single trial in a task that is closer to the types of tasks encountered in HCI; (c) through a final experiment we show that

detection accuracies in the region of 65 to 80% are sufficient to use these techniques in real-time interactive applications.

RELATED WORK

Brain Computer Interface and sensing techniques

There are many methods to detect neural data that can be used in a BCI system. Some notable ones are functional Magnetic Resonance Imaging (fMRI), functional Near-Infrared Spectroscopy (fNIRS) and Electroencephalography (EEG).

fMRI measures the change in blood flow which is related to brain activity by sensing the magnetic field around the scalp. fMRI is an effective technique for brain function imaging in medical science. It can quantify neural activities with high spatial resolution but fMRI machines are big, expensive and produce loud noises [12]. Moreover, this technique is vulnerable to the existence of metal objects and head movements [21], both of which are common when using a computer. For these reasons, it is not favoured by HCI researchers.

fNIRS measures the concentration of blood oxygen using infrared light [24]. This method has been used mostly to measure the user's mental workload [12, 21]. It is comparable to EEG due to its low cost and portability. However, it has lower temporal resolution compared to EEG making it difficult to detect fast responses after an event occurs [12].

EEG records electrical fields produced by neuronal activity [5]. It is a popular non-invasive brain imaging method because of its lower cost (compared to fMRI), portability and high temporal resolution [12]. There are two approaches to measuring EEG signals: invasive (implanting an array of electrodes in the cortex layer of brain) and non-invasive (attaching a number of electrodes to the scalp). Non-invasive approaches have more opportunities for use in everyday use.

Measuring and Processing EEG Signals

Raw EEG signals can be mapped to application control using one of two learning methods: *Operant Conditioning* and *Pattern Recognition*. In *Operant Conditioning*, the users need to be trained to control their own brain activity while they are provided with real time feedback. For example, users train the system such that when they think about right hand movement or a foot movement the application navigates through a virtual street [5]. Alternatively, *Pattern Recognition* uses signal processing and machine learning techniques to reveal the mental states or activities of *untrained* users [5]. The second method has the benefit of making it easier for users and can make a BCI application universal for general people, rather than a specific individual.

Pattern Recognition method has been widely used in neuroscience to discover the neural activity inside a

human's brain. For example, a P300 speller uses pattern recognition to detect the P300 pattern to help spelling/typing letters [7] as well as selecting the intended object [1]. This speller works by flashing rows and columns in a random order of a grid of alphabets. The user concentrates and counts the number of flashes over his/her intended letter. A P300 pattern then appears after the flash over the intended letter and will be clear enough from noise to be detected after a few flashes.

Error Related Negativity (ERN)

ERN is a pattern observed when a user makes an error in a reaction time task. Its shape is a negative deflection which appears in the ongoing EEG right after the time the decision was made. The ERN also appears when users have feedback about their response accuracy [14]. Its amplitude is large when the user is clearly aware of his/her error and is small when user is confused (where errors are caused by data limitation) [3]. However, despite the change in amplitude, ERN latency seems to be consistent (about 100ms after the event). Interestingly, the amplitude of ERN does not depend on the behaviour accuracy itself but the user's perception about it [3, 25].

To date most of experiments involving ERN detection are done offline by averaging over multiple trials. Participants are asked to perform a multiple-choice task in which trials with incorrect responses were used to archive a clear ERN pattern. For example, Gehring et.al [10] used this methods to determine the effect of a speed/ accuracy trade off on different representations of ERN. Scheffers and Coles [3] use this method to conclude that ERN is a manifestation of the ongoing monitoring system in the brain which compares the expected response and the actual response.

However, in an interactive application, ERN will be most useful when it can be detected immediately after it happens. This requires the online detection of the ERN pattern (as opposed to offline) from EEG signals. One way to achieve this is by caching and averaging a small number of patterns in a limited time in order to refine a reasonably clear ERN [20]. This method requires a waiting time to collect sufficient signals and therefore delays the progress of error correction as it goes through multiple trials.

Various attempts have also been made to detect ERN from a single trial. One example is from Ferrez and Millan [8] where they train a Gaussian classifier to recognize error and correct trials on a single trial basis but after the experiment was completed (so considered an offline single-trial approach). Another example is from Dal Seno et.al. [4] who used a Linear Discriminant Analysis (LDA) classifier to cross check with every spelled letter using a P300 speller. The reported performance of the online version varies from 58% to 69%. This result is just better than chance but still encouraging, as this is one of the few attempts at detecting ERN online and from a single trial.

All the above experiments have been carried out on EEG headsets with up to 256 electrodes and with sampling frequency up to 16 kHz. Most importantly these approaches rely on having access to Fz, Cz, Pz and Oz channels. We are not aware of any attempt to detect ERN from a low-cost portable headset where the number of channels and sampling frequency are limited. Hence, we investigate if a low cost, features limited EEG headset can detect ERN and benefit to HCI.

ONLINE ERN DETECTION

The online, single trial ERN detection algorithm we present below is an adapted version of the logistic regression algorithm from Christoforos et.al [2].

If the user makes a decision (i.e by pressing a button) in a Flanker Task at time t we can create an epoch $x(t)$ for a channel around that decision moment. The total length of $x(t)$ is l samples (which is the number of samples in that epoch). We need to design a supervised classifier so that its output:

$$y(t) = \beta^T * x(t)$$

is expected to be maximally discriminated between two cases: either there is ERN or no ERN in that time window. This output $y(t)$ is a real number.

β : coefficient matrix [$l \times 1$] that is unique for each channel.

We tested this algorithm with different signal pre-processing methods. For example, we tested with $x(t)$ as EEG signals from all channels at a time moment t following the method from [18]. This type of input does not give us a good classification. We also tried to design a coefficient matrix for all channels but did not receive a good result.

In our approach $x(t)$ is signal samples of a time windows (combination of two windows: before and after pressing button). Figure 2 presents the procedure of classifying an EEG epoch which is one of two types: ERN and no ERN.

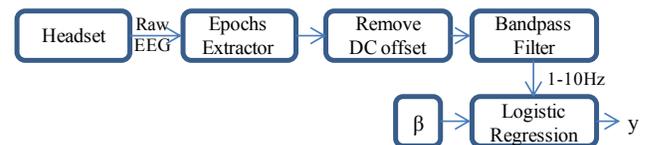


Figure 2. Online Single Trial ERN Detection Procedure

The next step is to find the coefficient matrix to satisfy the expectation of $y(t)$.

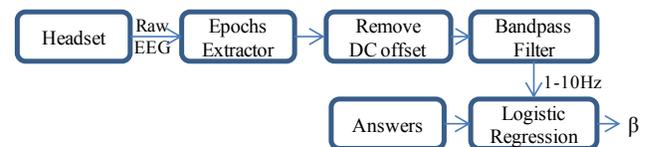


Figure 3. Coefficient matrix extractor

Figure 3 explains the procedure of finding the coefficient matrix β using a supervised logistic regression machine learning technique which is briefly explained in [2]. Once

the coefficient matrix is found, it can be used to classify an EEG epoch into one of two: there is ERN or there is no ERN.

EXPERIMENT 1: FLANKER TASK

To validate the above algorithm on a low cost, portable EEG device, we designed a Flanker Task that is similar to [18]. Twelve local students (7 males) were recruited to participate in this study. Each completed this experiment individually.

To mimic the experimental conditions in [3, 25] for a typical Flanker task experiment and reduce noise in data collection, the participant was seated in front of a screen in a dimly lit room. Participants were told to sit comfortably and minimize eye movement and blink as infrequently as possible while performing the task.

Participants were asked to perform a version of Flanker Task where they had to press one of two keys to specify the direction of a central arrow that was bounded by flanker arrows. There were two types of arrows, each type had two stimuli: congruent stimuli (<<<<<< and >>>>>) and the incongruent stimuli (<<◇<< and >>◇>>). All 4 stimuli were used in our trials in a random order.

For each trial there was a fixation cross in the centre of the screen for 500ms. It was replaced with one of four stimuli. The stimulus was presented for 100ms before the screen was cleared. Participants were asked to respond by pressing one of two buttons corresponding to the direction of the central arrow. At this time a string “-” appeared to mark each interval of 1000ms waiting. After the participant’s response, the screen remains clear for 500ms. All stimuli were presented in white font on a black background. At a viewing distance of around 100 cm, the visual angle of the arrow stimuli was 0.4° vertically and 0.6° horizontally, and between them was 0.3° space. This procedure is similar to the task carried out in [3] and [6]

Participants wore an Emotiv EPOC Neuroheadset that has the ability to capture raw EEG in 14 channels (of the 10-20 international system) from different locations around the human head. At first, participants were given one practice block of 40 trials. After that, they performed 4 blocks of 40 trials in which EEG signals were collected. Participants had 2 minutes to rest after each block. We collected a total of 1920 trials from all participants of the experiment.

SIGNAL PROCESSING

The EEG signals, captured from the Emotiv headset, were divided into 2-second length epochs (1 second before and 1 second after the key press moment). With the sampling frequency at 128Hz (after downgrading from 1024Hz), the length of each epoch is 256 samples. The first 200ms of each epoch were used to remove DC offset following which all epochs were filtered in 1-10Hz to remove components that are not in the ERN frequency bands of that particular epoch. Figure 4 shows examples of single trial ERN in the

two cases when the trial was incorrect (Fig 4a) and correct (Fig 4b). These examples were picked to demonstrate the difference in the ERN pattern from a single trial.

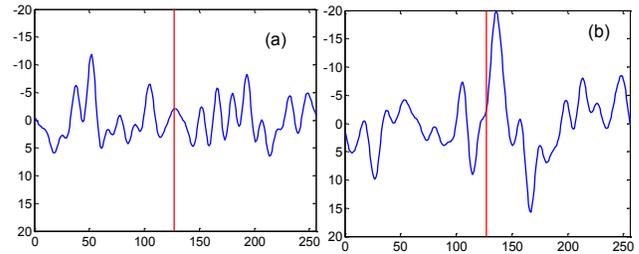


Figure 4. Examples of single trial ERN for two cases: incorrect (a) and correct (b). The x axis is number of samples (1 sec = 128 samples). The y axis is amplitude (uV). The red line is the key press moment (sample number 127).

RESULTS

For each channel, half of the trials (= 80 trials) were used for training via a logistic regression technique and other half were used for testing. We performed a t-test on $y(t)$ (calculated by multiplying $x(t)$ with the coefficient matrix) for each channel per user to check if there was a significant difference between two types of output (ERN and no ERN epochs). Based on the results of the t-test we found that on average F3, F4, F8, FC5, FC6, AF4 were the channels that yielded significant differences in $y(t)$ implying they are best suited to detect an ERN pattern from a signal $x(t)$.

From the test trials we found that using data from F4 channel can discriminate the two types of responses best: 69.7% of correct trials were classified as correct and about 70.3% of erroneous trials were classified as incorrect.

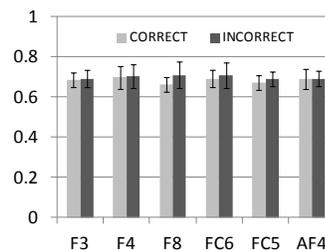


Figure 5. Classifier accuracy on a single trial basis

Figure 5 shows the average accuracy of different sensing channels on which the algorithm works most effectively. It was observed that those channels, which correspond to the frontal lobe, produce better classifications than other channels. This matched the literature about the origin of ERN, which is from the Anterior Cingulate Cortex (ACC) [26]. This also satisfies the fact that ERN has the frontal-central distribution of the human brain.

Figure 6 illustrates the average EEG signals over all epochs belonging to two cases: correct and incorrect.

To investigate the efficiency of the classifier further, we used the Receiver Operating Characteristic (ROC) Analysis [23]. It has two distinct inputs: hit rate (or true positive rate) and false alarm rate (or false positive rate) as two separate

performance measures. ROC analysis has been used in machine learning recently to justify how good a classifier is by evaluating its discriminating power [15]. Figure 7 shows four ROC curves of 4 frontal-central channels. The further the curve is from the diagonal line, the more effective that classifier is. The area under the curve (AUC) gives an indication of how well the classifier is performing. The AUC of 1 indicates a perfect classifier and 0.5 indicates a random chance of classification. Our classifier achieves an average of 0.77 across all channels. The AUC for the best six channels (F3, F4, F8, FC5, FC6, and AF4) are shown in Figure 8.

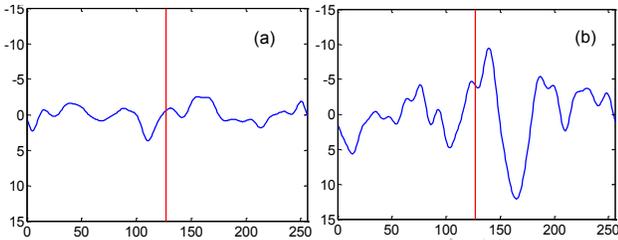


Figure 6. EEG signals at channel AF4 with number of samples in the x-axis (1 sec = 128 samples) and uV in the y-axis. (a) Correct Epochs and (b) Incorrect Epochs. The vertical red line is the moment when key was pressed.

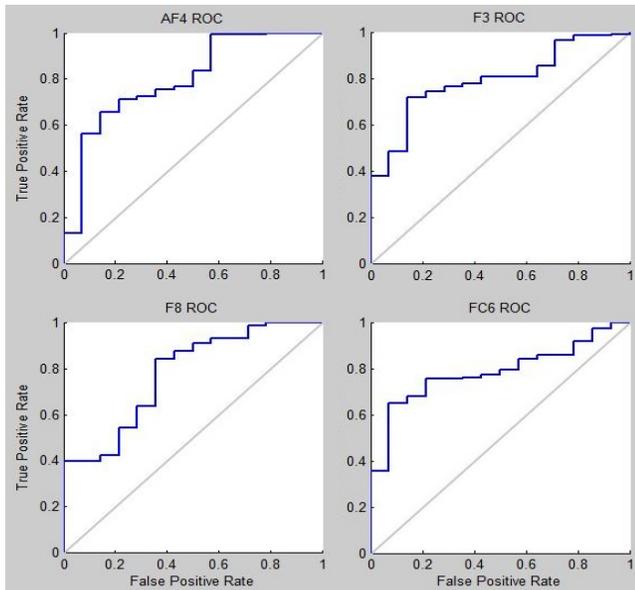


Figure 7. ROC curve for 4 channels: AF4, F3, F8, and FC6

Discussion

The result of the above Flanker Task demonstrate that Emotiv™ device is capable of capturing EEG signals with sufficient quality for a classifier to be able to detect ERN with an accuracy of about 70%. This initial result suggests that we can achieve ERN detection rates that can benefit interactive applications where these can be further improved through manipulating the feedback mechanism such that any likely error (or mistake) is noticed earlier by the user. This can provoke higher amplitude of ERN, further increasing the probability of accurately detecting it.

As a result it will enhance the advantage of using ERN as the system can “sense” that the user made a mistake and interrupt or undo the last action.

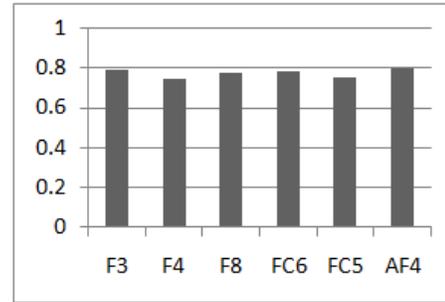


Figure 8. AUC for different channels

EXPERIMENT 2: BUTTONS SELECTION

Compared to Flanker tasks, a visually rich environment, like the one a user might encounter in an interactive application requires more users' mental workload. This will trigger neuronal activities from different parts of the brain; all of which can interfere with ERN signals. Mouse inputs in interactive tasks require free hand movements over longer distance which can cause bioelectrical signals from hand muscle activity. These activities will bring artifacts to EEG signals [22]. For these reasons it is not clear whether ERNs can be detected in a visually rich environment.

The main goal of this experiment is two-fold: first to confirm that an interactive task other than Flanker Task will produce an ERN pattern that can be measured using our portable EEG set and secondly to check whether the pattern can be classified using our online detection algorithm.

We designed a task that retains the main elements of a multiple-choice RT task while at the same time offering a visually rich environment that goes beyond the Flanker task and be closer to what might be expected of an interactive application.

Task: Button selection

In this task the user had to select an object with the size of a 160 x 55 pixel button. In each trial, the user was asked to select a button in a limited time. There are 7 buttons with text from 'Link 01' to 'Link 07' which were arranged in an ascending order from top to bottom. The trial began when user clicked on the START button. A message at the top of the screen showed which button needed to be clicked (Figure 9a). The user then moved the mouse cursor toward the required button. A timer was placed at the top right of the screen to show elapsed time since the trial started. The timer was to put pressure on user to complete the trial as quickly as possible. The user must finish the trial in a limited time (1.4s) otherwise a TIME OUT message would appear and the trial restarted. We chose 1.4s because 1.3s was the average time for an experienced user to complete the task. When the mouse cursor was moving toward the desired button and there was 30 pixels distance remaining to reach, the order of the buttons may change (Figure 9b). The probability for the buttons' order to change was 50%.

Because the position of the desired button was changed when the cursor was very near, user may not have enough time to change their decision to click on the intended button but still aware of the result of their action. The intention was to provoke an ERN in this case. When user clicked on a button, a message notified user if they had clicked on the correct one. There was a 3 second waiting time before a new trial started.

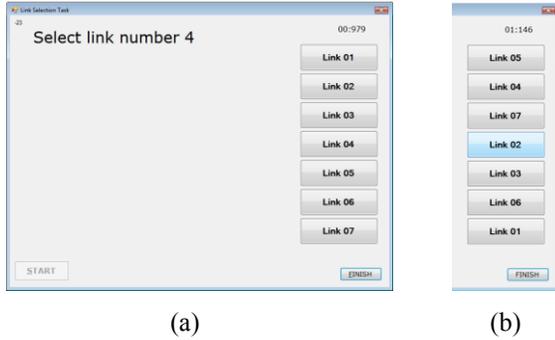


Figure 9. User interface of the task at the moment of (a) start trial and (b) cursor is near the desired button.

Design and procedure

Nine students (6 males) recruited from the local university took part in this experiment. All of them knew about the Emotiv device but had never tried it before. Each of them was given the consent form and information sheet informing them about the task details. They were asked to sit comfortably but alerted to the task. They were also instructed to blink less during the experiment, especially around the deciding moment (mouse click). The experiment took place in an office environment with ambient noise.

Participants were given some practice trials to become familiar with the task until they confirmed they were ready to start the experiment in which data was collected. Each participant performed 4 blocks of 40 trials each and there was 2 minute break between each block.

Data Collection and Analysis

Signals were collected using the Emotiv neuroheadset. They were then divided into 1.5 second epochs around mouse click moments, 0.5 seconds before and 1 second after. The moments that a participant saw the result of their action were also captured as it is assumed that was when s/he realized that s/he has made a correct or incorrect choice. The epochs were then divided into two groups where the first group was used to compute the coefficient matrix (β). The second group was formed with the remaining data.

This coefficient matrix was then used to classify epochs of the second half. There were two cases to classify: participants clicked on the desired button successfully or they performed it unsuccessfully. The classifier performed the classifying task on a single trial basis. Epochs were divided based on the probabilities of having ERN and not

having ERN. The result then was compared with the ground truth of participants' confirmation.

We do the classification and analysis offline but using the online detection algorithm. The primary goal here is to verify the effectiveness of the algorithm not to provide users with real-time feedback in an interactive task.

Result

We visually inspected the EEG signals around the moments when a participant clicked a mouse button to see if ERN patterns can be observed or not. ERN pattern was clearly visible in epochs where participants made an incorrect decision (clicked on a wrong button). As the trial result was displayed immediately after user clicked, the epochs of displaying moments were the same as the epochs around mouse click moments.

Figure 10 shows the mean of EEG signals around the moments when a participant click (select an object).

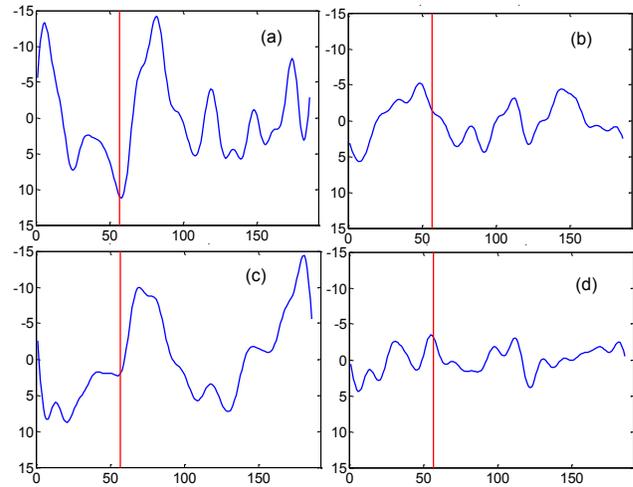


Figure 10. EEG signals around moments where a participant made an incorrect decision (a, c) and correct decision (b, d) at channel F3 (top) and FC5 (bottom). The red lines mark the moment that the mouse button was clicked.

In addition, the Logistic Regression algorithm can classify epochs belonging to those two cases: correct and incorrect responses. The coefficient matrices (β) of the same participant from the first half of data were multiplied with the extracted epochs $x(t)$ to produce output $y(t)$. We perform a t-test on $y(t)$ for each channel per user to check if there is a significant difference between two types of output: correct (user clicked on a correct button) and incorrect (user clicked on an incorrect button). We investigated further based on the t-test result and found that the accuracy of the classifier on incorrect decision moments and correct decision moments were 64% and 67% at channel F3 and 63% and 69% at channel FC5. The results from this experiment are similar in accuracy to the result of the Flanker Task. It is notable that those two channels that have the best classifying accuracy are in frontal-central part of human brain which is consistent with the result obtained from Flanker Task.

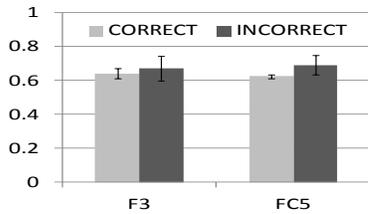


Figure 11. Single trial classification accuracies

Discussion

The result obtained from this experiment confirms the existence of an ERN pattern in a non-Flanker task. Those patterns can be seen clearly in epochs when participants commit an incorrect decision. Moreover, this result confirms that ERN patterns appeared in a normal application in a working condition as long as the ERN triggering condition is designed properly.

The method investigated in this experiment demonstrates the ability to detect an ERN pattern using a off the shelf EEG headset. It also demonstrates the ERN detection on a single trial basis. This is promising for HCI designers as the ultimate purpose is not to develop an effective and robust classifier but to harness this type of ERP into HCI.

While one might argue that we processed EEG data offline, it is worth noting that we used a detection algorithm that is essentially an online detection algorithm that works on a single trial basis. This algorithm can be used as a light weight module and can be run in the background of the system. Thus, it can be easily brought to an online version with the same design and signal processing methods.

The questions that remains is how useful this classification level is when ERN is applied to interactive applications and is 65% to 70% accuracy enough in effectively assisting users. The answer varies depending on the types and purposes of each task. Therefore, we designed the next experiment in order to demonstrate at least one set of applications that benefit from this level of ERN classification accuracy.

EXPERIMENT 3

The goal of the experiment was to determine whether the accuracies of the ERN classifier can benefit interactive applications and improve users' performance. It is unlikely that ERN classifying rate can reach 100% at all time. Therefore we design this experiment to justify the trade-off between ERN accuracies and benefits toward users.

The chosen application for this experiment was similar to Superflick [19]. Superflick is a pointing technique which is based on Flick (sliding/ throwing the object across the table) but adds a correction step. Flick is an open-loop technique, providing fast movement but requires practice to achieve accuracy. Superflick offers a "remote drag-and-drop" correction phase if the object is off the target. In our experiment we integrate a "simulated ERN detection" module into Superflick to test if there are any performance differences between different types of detection accuracies.

There are several reasons why we pick this specific pointing technique to explore performance. The main motivation is that it is easy to establish ground truth to compare the different techniques. Secondly, Flick is a popular interaction technique that has been studied extensively in the HCI literature for both handheld and tabletop environments. Thirdly, the authors have experience with the Flick technique and understand how the Superflick design can be improved through the introduction of an ERN detection module. This allows us to examine the effect of various ERN detection rates on users' performance.

The simulated ERN detection module consists of a random function used to mimic the ERN classifier. There are 4 types of accuracy: 50%, 65%, 80% and 100% to simulate the probabilities that the classifier will detect ERN patterns. The random function was controlled so that accuracies were as precise as required. During the experiment, participants were not aware that the ERN module was simulated.

Task and Technique

Participants had to move an object onto a target. The task started with the animated ball to move (the main ball – the lower yellow circle in Figure 1) located at the middle bottom of the screen. The target was a green circle and was assigned randomly among 15 targets.

The participant used a stylus to flick the object onto the target. As soon as the flick action was completed (by lifting the stylus of the touch surface) a circle appeared at the estimated final position of the main ball (the estimated ball – the higher yellow circle in Figure 1) which was calculated based on the distance and duration of the flick gesture. This instant visual feedback allowed the user to know whether or not the ball would actually hit the target. This was done to help with potentially triggering an ERN (although we do not capture these signals but rely on our controlled random function). At this point the mimicked ERN detection module will be triggered causing either the red (mimicking ERN detected) or the green (mimicking ERN not detected) light to be visible. The main ball then moved from the start point toward the estimated ball with the speed of 1200 pixels per second. When the main ball was moving, the user had a chance to remotely drag-and-drop the estimated ball to the target (as in SuperFlick). During the drag process, the main ball automatically moves toward the estimated ball. The trial finished when the main and the estimated balls met each other for over 200ms. This threshold was set to prevent accidental overlaps of the two balls. If they met inside the target, the trial was marked as successful and unsuccessful otherwise. A new trial started after 3s waiting and the user interface was reset to the original condition (with a new target).

The simulated ERN detection worked as follow: 250ms after the main ball left the start position, the ERN detection light, which was placed at the bottom right corner of the screen, started showing red if ERN pattern was detected (estimated ball was not inside the target) and green if not. In

case of the light indicating red, the main ball's speed decreased dramatically to 15% of the original speed (180 pixels per second). The user then had more time to remotely correct the position.

The task was built in C# for the experiment, and was installed in a bottom-projected tabletop system. The table was 105x88x106cm (w x d x h) with display size of 72.5x60cm and projector resolution of 1280x1024 pixels. Participants used a Wacom CTE 430 tablet (dimensions: 210 x 208, active area: 127.6 x 92.8mm) for input.

Design and Procedure

Nine local students (5 females) with age from 21 to 41 participated in this experiment. All of them heard about the Emotiv headset but never had used it before. They were given a consent form and information sheet explaining about the purpose of the experiment. Participants were not aware that we were simulating ERN detection. They wore the EEG headset, stood comfortably but alerted to the task in front of the tabletop, and were led to believe that their EEG signals were used to assist them in their task. But instead of detecting ERN based on the measured EEG signals, we created a detection module which is a controlled random function. The accuracy of this detection module was controlled to simulate the accuracy rates.

Participants were given 1 block of 50 trials or more to practice until they were familiar with the task and ready to start. After that, they performed 4 blocks (with minimum 50 trials) with ERN accuracies of 50%, 65%, 80% and 100%. A block was finished when number of trials was more than the minimum number of trials and ERN detection accuracy reached the required number.

Result

Figure 12 illustrates the percentage of successful trials (the main ball ended inside the target). We applied one-way ANOVA test with Bonferroni post hoc multiple comparisons on the data and found that there was significant difference between groups ($F = 17.517$, $p < 0.001$). However, there were no significant difference between groups of 65%, 80% and 100% ($p > 0.5$).

We also analyzed the number of unsuccessful trials when there were red and green lights separately. The intention was to check if users made mistakes naturally when the main ball moved slowly (red light); and if there were more unsuccessful trials when the main ball moved with full speed (green light).

Their mean values are shown Table 1. We ran one-way ANOVA test with Bonferroni post hoc multiple comparisons on those data and found no significant difference between groups in case of Red light ($p > 0.5$). It was noteworthy that even there was significant difference between groups in case of Green light ($F = 11.741$, $p < 0.01$), there was no significant difference between groups (50%, 65%), (65%, 80%), and (80%, 100%) ($p > 0.5$).

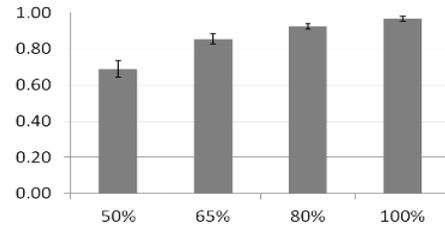


Figure 12. Accuracies of Superflick with different ERN detection accuracies

	50	65	80	100
Red	0.12	0.03	0.06	0.03
Green	0.47	0.35	0.22	0.00

Table 1. Mean % of unsuccessful trials with the two indication light

Discussion

The results obtained from participants show that when integrate ERN detection module with 65% accuracy was as good as 80% accuracy. This was proved as no significant differences were found between two groups in term of overall accuracy and when ERN was not detected (incorrectly). Our results also show that integrating an ERN detection module might not prove beneficial if the detection accuracies are lower than 65%.

The classifier with 65% accuracy may provide higher error rate compared to 80% and 100% accuracies but still benefits interactive tasks. If a system has very high accuracy (less error rate), it can promote hasty commitment to selections [11]. This is because users overly rely on the system and know that there is little cost for making mistakes. In the real world, the cost of an error may be much higher. Therefore, a system with lower classification accuracy may require more attention but has lower cost of recovery hence still offers benefits to the users.

These results confirm that ERN classification accuracy with Emotiv can benefit interactive tasks as good as other expensive devices, yet offer the advantages of portability, low-cost and instantaneous classification.

This experiment also shows an opportunity to assist users in pointing and table top applications using ERN. In these tasks, objects are usually out of range for users to select therefore they need to use several techniques such as drag-and-drop, radar, etc. ERN integration does not replace these techniques but provides better performance and more precise selections for users.

DISCUSSION

ERN has the potential to enrich interactive applications. The results from the studies in this paper can provide guidance on how best to begin harnessing ERN for such interactive experiences.

Employ ERN to Assist Interactive Tasks

The studies in this paper have shown that it is possible to detect ERN patterns using an off-the-shelf EEG headset on an online single-trial basis. If integrated into interactive

tasks, an ERN detection module provides another medium for HCI designers to access users' intentions, which is intuitive and directly from the users' brain. This module can be designed as a lightweight background feature.

An ERN pattern will appear in any multiple-choice RT task, of which the button selection task used in this work is just one example. The ERN pattern can be detected within 150ms of the event onset. This means within this time window the user becomes aware that they have made a mistake; the interactive application can know this and respond. In most cases this type of information is not available without access to the user's EEG signals. Even if knowledge of user error is available in some other way (such as text auto correction, prediction techniques) it may not offer as fast response as ERN detection.

The nature and scope of usage of the ERN signal will depend on the usage context, the creativity of the designer and on the ability of the user to ignore recommendations that are incorrect or inappropriate.

Dealing with incorrect classification

Detecting ERN correctly is a challenge. We can be reasonably sure that even the best classifier will never achieve 100% accuracy in an online single trial system. Consequently, an interruption management system is needed to be integrated into the system. This can reduce the disturbance to users when the system makes suggestions based on incorrectly detected ERNs. For example, if a user chooses to ignore a suggestion that pops up based on a confirmed ERN, the pop up must not prevent the user interacting with the user interface. Designers wishing to use ERN in their applications must be careful to ensure that such ERN-based suggestions do not stop the user from working with the system.

Potential Applications of ERN

As ERN is elicited in any multiple choice RT time task when the user is confused or aware of their accidental action this has great potential in many type of interactive applications. We suggest some of them as follow:

Gaming conditions: ERN can be used to provide users with a new form of experience in gaming. For example, in time critical missions (like shooting a character in World-of-war craft) sometimes network delays and other external factors may affect the overall outcome of the mission or battle. In these cases if an ERN is detected this can be used in systems decision making process either by giving the player another chance or changing the time-stamp of user triggered events to make the outcome seem as the user intended.

Aiding Object Selection: Selecting a static target among a selection of objects is a multiple choice reaction time task. The user starts with deciding on the target then makes an initial open-loop movement followed by a final correction phase where they move the pointer or finger on the target to select it. In this correction phase if the user usually receive visual feedback on whether or not their selection was

successful. This feedback combined with the initiation of the correction-phase movement can trigger ERNs if the movements are fast enough. Thus it should be possible for an ERN detection module to detect errors in users' intention and attempt to correct it. In many instances the cost of recovering from a wrong button press or a pointer selection can be quite high – the application might be launched and the user would have to close it before re-launching the right application. In these circumstances when ERN is detected the system can prompt the user if the target was selected correctly potentially helping the user.

ERN detection can be combined with P300 to reduce the user's mental load and frustration (error-recovery is time consuming and difficult) that is associated with working with P300 data. For example, in a object selection using P300 on a multi-touch table [27], ERN can be used as final confirmation that the object is the user's intended selected object. This will be very useful because it is difficult to confirm using only P300 that the selected object on the table is the one that the user wanted. Moreover, ERN usage can eliminate the trial of hitting the BACK button to de-select an object in P300 spelling which is time consuming and requires high concentration. This can also be applied into object selections on a tabletop for people in working condition so that the time spent in correcting wrong selection will be reduced and efficiency improved.

Mobile Spatial Navigation: Sometimes, a user still needs help in using and navigating using electronic maps on the fly (i.e. Google maps). HCI designers may integrate ERN into the system so that it can detect the confusion and error awareness moments in order to provide appropriate suggestion based on the location. One obstacle is EEG is known to be sensitive to movements such as walking and moving your body. This can make the EEG signal very noisy reducing the accuracy of the ERN detector. Before being fed into the classifier, EEG signals may need to be carefully pre-processed.

Multiple users' applications: ERN use can be extended to multiple users scenarios where a person's ERN is made visible to the entire team so team-support is available when the user is confused. For example if a gamer is confused in navigation or shooting activity, his/her teammates can assist him, or in collaborative table top applications, other people may give help and suggestion to a person whose avatar is being shown as confused or aware that they have made a wrong selection encouraging collaborative peer-learning.

CONCLUSION

The experiments described in this paper offer some valuable guidelines for HCI designers. We show that ERN patterns can be detected using an off the shelf EEG headset on an online single trial basis. Moreover, we apply this model to an interactive task to illustrate that it can work with normal interactive applications in a working environment with ambient noise. We finally show that accuracies in the region of 65 to 80% are sufficient for ERN

to be effectively integrated into HCI applications. In our discussions we suggested novel ways in which HCI applications could benefit from ERN.

ACKNOWLEDGEMENT

We thank Dr. Nick Yeung (Oxford Neuroscience, University of Oxford) for providing access to shared data and useful discussions in the initial stages of the project.

REFERENCES

1. Campbell, A., Choudhury, T., Hu, S., Lu, H., Mukerjee, M.K., Rabbi, M., and Raizada, R.D.S. NeuroPhone: brain-mobile phone interface using a wireless EEG headset, in ACM SIGCOMM. (2010), 3-8.
2. Christoforos, A., Dimitris, T., Niall, A., and David, H. Temporally adaptive estimation of logistic classifiers on data streams. ADAC. 3, 3: (2009), 243-261.
3. Scheffers, MK, and Coles, MG. Performance Monitoring in a Confusing World: Error-Related Brain Activity, Judgments of Response Accuracy, and Types of Errors. Proc. of JEPHPP. 26, 1: (2000), 141-151.
4. Dal Seno, B., Matteucci, M., and Mainardi, L. Online detection of P300 and error potentials in a BCI speller. Computational Intelligence and Neuroscience (2010).
5. Dornhege, G., R.Millan, J.d., Hinterberger, T., J.McFarland, D., and Muller, K.-R. Toward Brain-Computer Interfacing. MIT Press. (2007), 393-408.
6. Eriksen, C. and Schultz, D. Information processing in visual search: A continuous flow conception and experimental results. Attention, Perception & Psychophysics. 25, 4: (1979), 249-263.
7. Farwell, L.A. and Donchin, E. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography and clinical neurophysiology. 70, 6: (1988), 510-523.
8. Ferrez, P.W. and Millán, J.D.R. You are wrong!: automatic detection of interaction errors from brain waves, in Proc. of IJCAI. (2005), 1413-1418.
9. Ganushchak, L.Y. and Schiller, N.O. Brain error monitoring activity is affected by semantic relatedness: An event-related brain potentials study. J. Cognitive Neuroscience. 20, 5: (2008), 927-940.
10. Gehring, W.J., Goss, B., Coles, M.G.H., Meyer, D.E., and Donchin, E. A Neural System for Error Detection and Compensation. Psychological Science: (1993).
11. Gutwin, C. and Cockburn, A. Improving list revisitation with ListMaps, in AVI, ACM. (2006).
12. Hirshfield, L.M., Solovey, E.T., Girouard, A., Kebinger, J., Jacob, R.J.K., Sassaroli, A., and Fantini, S. Brain measurement for usability testing and adaptive interfaces: an example of uncovering syntactic workload with functional near infrared spectroscopy, in CHI. (2009), 2185-2194.
13. Hjelm, S.I. and Browall, C. Brainball - using brain activity for cool competition. NordiCHI: (2000).
14. Holroyd, C. and Coles, M. The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. Psych review, 4: (2002).
15. Honghu Liu, G.L. Testing Statistical Significance of the Area under a Receiving Operating Characteristics Curve for Repeated Measures Design with Bootstrapping. J. Data Science. 3: (2005), 257-278.
16. Lee, J.C. and Tan, D.S. Using a low-cost electroencephalograph for task classification in HCI research, in UIST. (2006), 81-90.
17. Miltner, W.H.R., Braun, C.H., and Coles, M.G.H. Event-related brain potentials following incorrect feedback in a time-estimation task: Evidence for a neural system for error detection. J. Cognitive Neuroscience: (1997), 788-798.
18. Parra, L., Alvino, C., Tang, A., Pearlmutter, B., Yeung, N., Osman, A., and Sajda, P. Linear Spatial Integration for Single-Trial Detection in Encephalography. NeuroImage. 17, 1: (2002), 223-230.
19. Reetz, A., Gutwin, C., Stach, T., Nacenta, M., and Subramanian, S. Superflick: a natural and efficient technique for long-distance object placement on digital tables, in Proc. of GI. (2006), 163-170.
20. Shenoy, P. and Tan, D.S. Human-aided computing: utilizing implicit human processing to classify images, in CHI, ACM. (2008), 845-854.
21. Solovey, E.T., Girouard, A., Chauncey, K., Hirshfield, L.M., Sassaroli, A., Zheng, F., Fantini, S., and Jacob, R.J.K. Using fNIRS brain sensing in realistic HCI settings: experiments and guidelines, in UIST. (2009).
22. Sörnmo, L. and Laguna, P. Bioelectrical Signal Processing in Cardiac and Neurological Applications. Elsevier. (2005).
23. Van Erkel, A.R. and Pattynama, P.M.T. Receiver operating characteristic (ROC) analysis: Basic principles and applications in radiology. European J. Radiology. 27, 2: (1998), 88-94.
24. Villringer, A. and Chance, B. Non-invasive optical spectroscopy and imaging of human brain function. Trends in Neurosciences. 20, 10: (1997), 435-442.
25. Yeung, N., Botvinick, M.M., and Cohen, J.D. The Neural Basis of Error Detection: Conflict Monitoring and the Error-Related Negativity. Psych review: (2004).
26. Yeung, N. and Cohen, J.D. The Impact of Cognitive Deficits on Conflict Monitoring: Predictable Dissociations Between the Error-Related Negativity and N2. Psych Science. 17: (2006), 164-171.
27. Yuksel, B.F., Donnerer, M., Tompkin, J., and Steed, A. A novel brain-computer interface using a multi-touch surface, in CHI. (2010), 855-858.