Design and Evaluation of a Self-Correcting Gesture Interface based on Error Potentials from EEG

Felix Putze
Karlsruhe Institute of Technology
Karlsruhe, Germany
felix.putze@kit.edu

Christoph Amma
Karlsruhe Institute of Technology
Karlsruhe, Germany
christoph.amma@kit.edu

Tanja Schultz
Karlsruhe Institute of Technology
Karlsruhe, Germany
tanja.schultz@kit.edu

ABSTRACT

Any user interface which automatically interprets the user's input using natural modalities like gestures makes mistakes. System behavior depending on such mistakes will confuse the user and lead to an erroneous interaction flow. The automatic detection of error potentials in electroencephalographic data recorded from a user allows the system to detect such states of confusion and automatically bring the interaction back on track. In this work, we describe the design of such a self-correcting gesture interface, implement different strategies to deal with detected errors, use a simulation approach to analyze performance and costs of those strategies and execute a user study to evaluate user satisfaction. We show that self-correction significantly improves gesture recognition accuracy at lower costs and with higher acceptance than manual correction.

Author Keywords

Error-Potentials; Self-Correction; Adaptive Interface; Gesture Recognition; Simulation; User Study

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Intuitive human-computer interaction (HCI) has been a highly active field of research in the last decade. Natural input modalities like speech or gestures have become broadly available. However, while those input techniques are an important step towards intuitive and efficient HCI, there are some important aspects which are still lacking. One major challenge when using machine learning and pattern recognition techniques to interpret user input is the introduction of a substantial error chance compared to traditional input devices like keyboards. Reasons are on the one hand limitations of generalizing from a finite set of training samples for data of high intrinsic variability and on the other hand ambiguities of complex, natural input patterns. Recognition errors often lead to an inefficient and unsatisfying interaction.

CHI 2015, April 18–23 2015, Seoul, Republic of Korea

ACM 978-1-4503-3145-6/15/04$15.00.
http://dx.doi.org/10.1145/2702123.2702184

In this paper, we propose the design of an error-aware gesture interface which is able to pro-actively detect erroneous system behavior in reaction to the input of a user. To detect errors, we exploit the fact that a discrepancy between the expected and the observed system behavior results in characteristic patterns of brain activity. Such a pattern is called Error Potential (ErrP) and can be measured using Electroencephalography (EEG) almost immediately after erroneous system feedback is presented following a user action. Therefore, identifying ErrPs allows the design of systems that proactively recover from errors.

While the general feasibility of ErrP detection from EEG has been established at least in the context of Brain-Computer-Interfaces (BCIs, see next section), there is a lack of investigations concentrating on the design of the actual error recovery behavior and the related usability implications. In this paper, we describe the design of an error-aware gesture recognizer. This recognizer uses ErrP detection and implements different strategies for error correction. Our main focus is to provide a thorough evaluation of the proposed error-aware recognizer, looking at both objective and subjective metrics. For this purpose, we use a comprehensive simulation employing data from multiple user studies.

RELATED WORK

In this section, we give an overview of related work on error recovery. Recovery from error states is a feature which is required for any type of interaction in which the interpretation of user input is error-prone, for example from speech or handwriting recognition. Most mature in that regard are probably spoken dialog systems which often need elaborate recovery strategies. For example, Bohus and Rudnicky [1] described ten distinct strategies to recover from non-understanding situations and empirically evaluated the performance impact of the different strategies. Most of their presented strategies involved one of several alternatives of "reprompting" the user after a non-understanding. Another approach is to identify erroneous sections of a speech recognition result and propose other likely alternatives [16].

Most of those systems make use of confidence scores to estimate the presence of recognition errors [5, 16]. However, when statistical models are unreliable and generate incorrect results, it is unreasonable to expect a very reliable confidence estimate. For example, Vu et al. [18] showed that confidence scores in ASR correlated well to recognition performance for well-trained models but confidence reliability deteriorated for

models which were trained on small data corpora or data which did not match the testing data. This indicates that in order to provide self-correcting behavior for a user interface, we need additional information sources on the presence of an error besides confidence scores. One promising candidate in this regard is the detection of ErrPs.

Unsurprisingly, the idea to use ErrP detection for improving interaction has been first introduced in the context of BCIs, for which the necessary equipment for EEG recordings is already in place. BCIs as input or control device suffer from far-from-perfect recognition rates. A standard technique to remedy this is to always repeat each input several times. This increases robustness but leads to a low transfer rate [8]. The detection of ErrPs allows increasing accuracy and therefore the potential transfer rate. Combaz et al. [2] showed how they can detect ErrPs during operation of a P300 speller BCI. They suggest using the second best recognition result of the BCI in case of a detected ErrP and showed in simulations how this would improve performance. Spüler et al. [15] pursued a different approach and deleted the previously given input in case an ErrP was detected. They then prompted the user to repeat the input command. The authors showed how they can use an online ErrP classifier to significantly increase transfer rate. A similar approach was chosen by Schmidt et al. [13], who showed that they were able to reliably detect ErrPs online and demonstrated a significant increase in communication speed for their gaze-independent "Center Speller" BCI. The work by Margaux et al. [11] is one of the few examples of studies which do not only regard objective criteria but also investigate subjective evaluation of error-aware interfaces. For a P300 speller with second-best correction, the authors showed that a majority of participants "reported a preference in favor of a spelling including automatic correction". However, the authors also noted large individual differences regarding the subjective evaluation of their error-aware BCI. Llera et al. [9] used detected ErrPs to adapt the weight parameters of a logistic regression model for BCI operation to better represent the (assumingly) misclassified trial. They used simulation and offline analysis of data from eight participants to show that this process improved classification accuracy.

The number of studies which transfer ErrP detection to other input modalities – like gestures – is limited. Förster et al. [4] used classification of ErrPs during operation of a gesture recognition system to improve its performance by adaptation of the gesture recognizer. However, their system did not immediately react to the detected ErrPs by error correction. Instead, it focused on improving gesture recognition accuracy by selective online adaptation: Gesture trials which were classified correctly (i.e., did not result in an ErrP) were added to the training data to train a personalized gesture recognizer. This addressed the challenge of unsupervised adaptation that the addition of misclassified trials can result in performance degradation instead of improvement. Vi and Subramanian [17] proposed an ErrP recognition system based on a consumer-level EEG device. They performed person-dependent ErrP classification, using a test set of 80 trials of Flanker Task execution and achieved a classification accuracy of about 0.7. Using a simulation of ErrP classification with different error rates, they also showed that a non-perfect ErrP detection rate between 0.65 and 0.8 was beneficial for the enhancement of interactive systems for detecting user errors in spatial selection with the Flick technique on a touch surface. The authors analyzed accuracy improvements of allowing manual corrections when an ErrP is detected, but did not analyze costs or other usability aspects.

Corpus	Modalities	Size	Used for ...
BCI-ErrP	EEG	20	training of ErrP classifier (already existed)
Gesture-ErrP	EEG+IMU	11	evaluating ErrP classifier on gesture task
Gesture-Sim	IMU	20	evaluating accuracy & cost of recovery strategies
Recovery-Study	IMU	10	evaluating user satisfaction

Table 1. Overview over the different data corpora used in the paper. Size is given in number of sessions.

While there exists a large corpus of usability investigations on gesture-based interaction systems, we are not aware of many studies on the impact and handling of recognition errors. The cited related work on results from error-aware BCIs cannot be directly transferred to other HCI applications (e.g., gesture-based interfaces) for a number of reasons: 1) Errors of a gesture-based interface might generate different or no ErrPs compared to BCI input. 2) The feasibility of error recovery behavior depends on attributes of the input modality (number of classes, input recognition accuracy, error distribution, etc.), which differ between gesture and BCI input. 3) The acceptance of error recovery behavior is specific to the modality, as expectations on system performance, judgment of own performance, frequency of user errors vs. system errors, the costs of recovery behavior and many other factors differ between gesture-based interfaces and BCIs. 4) Some approaches for error recovery are modality specific (e.g., approaches using spatial re-arrangement for gesture interfaces). For those reasons, it is necessary to study the development and evaluation of error-recovery strategies which can make use of ErrP detection outside of a BCI application. We do this for the example of a gesture recognizer. Our main focus is on a thorough (objective and subjective) evaluation of different error recovery behaviors. For this purpose, we use a comprehensive simulation employing data from multiple user studies.

STRUCTURE OF THE PAPER

In this section, we describe how we design and systematically evaluate the necessary components for an EEG-based self-correcting gesture interface. This also defines the structure of our paper. First, we briefly introduce an existing ErrP classifier which is trained on the so-called *BCI-ErrP* corpus of participants operating a simple BCI task simulating feedback errors. Second, we introduce the experimantal setup, consisting of the gesture task, the employed gesture recognizer and the self-correcting interface. For the latter, we define several recovery strategies which are employed to respond to detected gesture recognition errors. Third, we evaluate the ErrP classifier by transferring its trained models to data of participants perceiving correct and erroneous feedback in the actual gesture task. We call this data of simultaneous gesture and EEG

recordings the *Gesture-ErrP* corpus. We then use an additional larger corpus – called *Gesture-Sim* – of 20 participants performing the gesture task without parallel EEG recordings to evaluate the performance and associated costs of a large number of recovery strategy variants in a comprehensive simulation. We finally perform a user study with 10 participants – generating the *Recovery-Study* corpus – using three selected recovery strategies together with simulated ErrP classification to compare user satisfaction between different automatic and manual strategies. Table 1 gives an overview of the different employed corpora.

Most of the presented results use evaluation methods which employ partial simulation of system components. Note that while this is an established method to evaluate error-aware interfaces (see for example [17]), simulation is no complete substitution for the evaluation of a complete end-to-end system. This should be considered when interpreting the results of the paper. On the upside, simulation allows us to generate results which are more general than results for few concrete end-to-end systems.

EEG-BASED ERROR RECOGNITION

In this section, we briefly review the employed methods for ErrP classification. A typical ErrP can be measured at frontocentral electrode positions and occurs in a window of about 150ms to 600ms after a stimulus (i.e., the feedback), with its most pronounced components being a negative peak around 250ms and a positive peak around 350ms [3]. The exact contour and latency of an ErrP varies with tasks and individuals [6]. Figure 1, which is taken from [12] shows a Grand Average of an ErrP pattern as difference between brain activity following error-free and erroneous feedback.

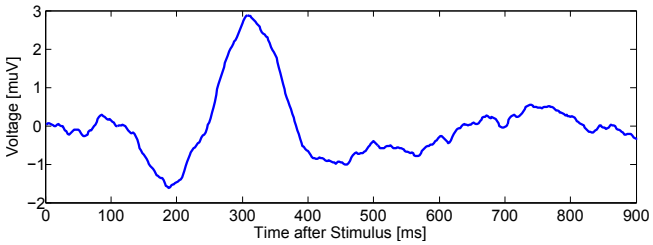

Figure 1. Grand Average of an ErrP (difference of correct and erroneous feedback) after a feedback stimulus in reaction to user input.

For the present investigation, we leveraged an existing ErrP classification system which is described and evaluated in [12]. In summary, this system extracts EEG data from a window of 1s duration after the presentation of a feedback. It preprocesses the data by re-referencing data to a common average and by removing ocular artifacts using Independent Component Analysis (ICA). The signal is then downsampled and the signals from two electrodes (Fz and Cz) used as features for a Support Vector Machine with a radial basis function kernel[1]. For each participant, the system is trained on available training data from other persons as well as a number of available calibration trials recorded from the current test person. This calibration data is also used to select only those training sessions which are closest to the data from the present test person. Additional details on the design of the ErrP classifier can be found in [12].

To validate this classification setup, we tested it on a corpus (originally described in [12]) using a highly controllable and easily executable task for elicitation of ErrPs. The employed paradigm to evoke ErrPs was operation of a mockup-BCI (selecting one of two presented options with mental commands) that simulated an error rate of 30%. For this task, EEG data was recorded at 500Hz using a BrainVision actiCHamp system with 32 active electrodes, of which 23 where used for ICA calculation and feature extraction[2]: Fz, F3, F7, FC5, FC1, C3, T7, CP5, CP1, P3, P7, O1, O2, P4, CP6, CP2, Cz, C4, T8, FC6, FC2, F4, F8. Impedance was kept below $16k\Omega$ for all electrodes. Pz was used as reference electrode and an additional light sensor attached to the stimulus presentation screen was used for synchronization. Using this setup, the *BCI-ErrP* corpus was recorded with data from 20 participants. On this data, the authors of [12] achieved an F-score of 0.86, corresponding to an accuracy of 0.92, using leave-one-person-out cross-validation with 125 calibration trials.

When applying the existing ErrP classifier to a gesture recognition context (see next section), we will use the BCI-ErrP corpus as training data. We use calibration data from the users of the gesture recognizer as calibration data. The rest of the classification setup remains unchanged.

EXPERIMENTAL SETUP

In this section, we describe the main parts of the experimental setup besides the ErrP recognition: The gesture task, the gesture recognition component and the self-correcting interface. To evaluate the potential for an ErrP classifier and corresponding error recovery strategies in a realistic, non-BCI related interaction task, we designed an experiment using a simple gesture recognizer for pointing gestures. This experiment consists of a pointing gesture task, a gesture recognizer and the self-correcting interface.

Gesture Task

In the employed task, participants selected and dragged images presented on a large projected display to a certain spot of a 2x3 matrix, depending on the content of the image (Step 1 and 2 in Figure 2 show the interface of the task). We used data from a wireless sensor wristband equipped with an inertial-measurement-unit (IMU) placed on the participant's right hand to classify the six possible pointing gestures. Figure 3 shows a participant performing the gesture task.

Before execution of the actual experiment, each participant trained a predefined movement sequence: From a resting position, the participant moved the arm to point at the bottom left corner, paused for about a second, moved the arm to the target cell of the matrix in a smooth motion, paused for about a second and returned to the resting position. This schema ensured consistent execution quality across all participants. The

[1]Note that the recording of additional channels is still necessary to perform ICA.

[2]Different subsets of electrodes were used in different setups and therefore not all were available for all sessions.

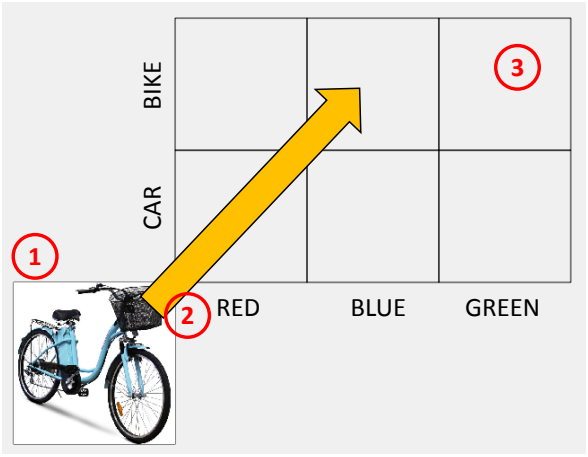

Figure 2. The gesture task: Image is shown (step 1) and moved by pointing gesture (step 2). Then, feedback on the recogized category is shown (step 3).

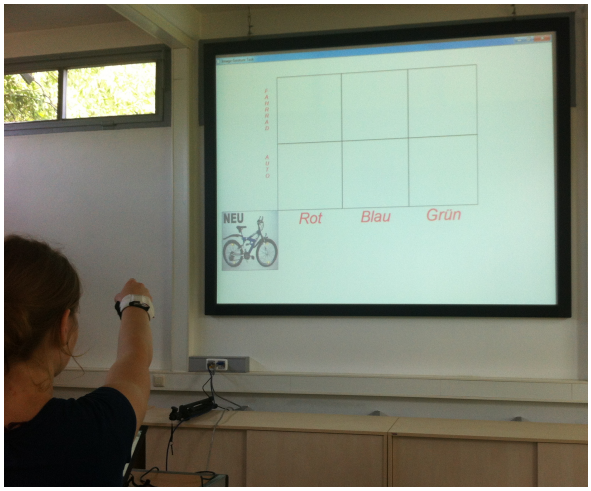

Figure 3. Execution of the gesture task.

task supported the participant in the correct execution by giving acoustic feedback in the form of a beep when a pausing position could be left. When the gesture was completed, the recognition result was displayed as a large overlay showing an abstract pictogram representing the selected category (Step 3 in Figure 2). At this point, an ErrP classifier can evaluate the subsequently recorded EEG to recognize whether an error has occurred. As recognition of ErrPs, and triggered effects in EEG in general, depend on temporal patterns in the range of milliseconds, it is important to provide such a stimulus-locking. Note that the selected window for ErrP classification does not contain systematic movement or system stimuli besides the presented feedback.

Gesture Recognition

We designed a simple person-independent gesture recognition system for recognizing the pointing gestures. The recognizer was designed to discriminate six different classes, corresponding to the six matrix cells of the gesture task. Arm motion was sensed with a sensor equipped wristband. We used a jNode sensor [14], an open research sensor platform which contains a 9 degrees-of-freedom inertial measurement unit (IMU). Sensor data was sampled at 50 Hz and sent wirelessly to a computer. We applied a standard processing chain consisting of segmentation, feature extraction and classification stages. It should be noted that the gesture recognizer was deliberately not optimized towards high accuracy. An almost perfect recognizer would not be of use in our scenario, since we are investigating recovery strategies from errors. We targeted an accuracy of 75%.

We employed a two-stage segmentation process, which first identifies segments of motion and then separates the actual pointing gesture from other hand movements. In the first stage, the motion data is segmented into parts containing motion and parts that are considered to contain no motion (idle). A motion segment is detected whenever the angular rate exceeds a given empirically determined threshold. The motion segment ends if the angular rate is below the threshold and stays below it for at least 200 ms.

In the second stage of the segmentation process, we model the movement sequence with a finite state automaton. Since the movement sequence follows a strict schema, this is a feasible approach for the case of this study. The finite state automaton has four states called `UNDEFINED`, `POINTSTART`, `GESTURE`, and `POINTEND` (see Figure 4). Whenever the segmentation stage detects a motion/idle change, we check for a state transition. The start state is `UNDEFINED`, which captures all motion that occurs between two gestural interactions. The `POINTSTART` state corresponds to the initial pointing on the picture at the bottom left corner of the display. The transition into the state `POINTSTART` is performed if the acceleration in the axis perpendicular to the back of the hand is within a range of $0.98\,\mathrm{m/s^2}$ of an experimentally determined reference value. This means, the orientation of the hand in 3d space is compared to a reference orientation based on the measured earth acceleration. The reference orientation depends on the height and distance of the projected image relative to the user and is therefore dependent on the local environment. The next motion segment triggers transition into the state `GESTURE`, which indicates the execution of the actual gesture. The next idle segment consequently triggers transition into the state `POINTEND`, indicating the pause at the target position. The next motion segment leads to the transition back into `UNDEFINED`. The motion segment corresponding to the state `GESTURE` is used for the actual classification of the gesture.

For the classification stage, we used a Gaussian mixture model (GMM) with five Gaussians per class for maximum likelihood classification. Features were computed on the complete gesture segment associated with the `GESTURE` state in the finite state automaton. Preprocessing consists of a mean subtraction to compensate constant offsets introduced by gravity (mean calculated on previous trials) and signal smoothing with a running average filter of order five. Due to the drift and noise present in inertial sensor readings, there is no established technique to reconstruct the actual trajectory performed in 3d space. As a result, we could not simply compute the direction and length of the performed gesture re-

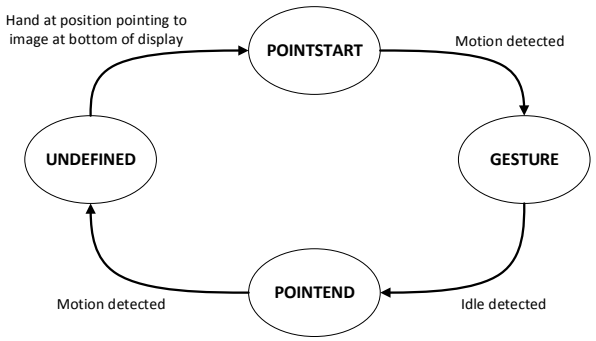

Figure 4. Finite state automaton that controls the second state of the gesture segmentation process.

liably. Instead, we computed the angle of motion in each of the three axes, the L_2-norm of these three angles and the duration of the gesture segment. The angles were computed by integrating the angular rate measurements from the gyroscope over the whole gesture segment. The resulting feature space therefore had five dimensions. The classifier was evaluated in crossvalidation, yielding a person-independent recognition accuracy of 77% (on the *Gesture-Sim* corpus) and therefore the gesture recognizer is suitable for our experiments. We also computed a confidence estimate for the gesture classification: Scores of the GMMs for each feature vector were normalized to a probability distribution across classes. The distance between the normalized score of the highest and second-highest scoring class was used as a confidence estimate of the classification result.

Self-Correcting Interface

For the described scenario, the self-correcting interface is designed as follows: The gesture classifier receives input from the IMU and outputs a probability distribution for six classes corresponding to the six possible matrix cells. The most likely class is used to present feedback on the recognized class to the user. The EEG data following this feedback is analyzed; in case the ErrP detector triggers, a recovery behavior is started to correct the error. The exact nature of this recovery behavior depends on the implementation of the recovery strategy. Different strategies can have different characteristics in terms of recovery accuracy, recovery costs and other factors.

In the evaluation section, we will show how the existing ErrP classifier performs on erroneous feedback during the gesture task. In the experiments comparing different recovery strategies, however, the ErrP component was replaced with a simulated ErrP classifier that receives the ground truth gesture class and the output of the gesture classifier to detect ErrPs with a recall and precision of both 0.8. The results from the isolated analysis of the ErrP classifier justify this simplification of the experimental setup as they show that we are able to achieve this performance from real EEG data. Simulating the ErrP classification allowed us a better control over the distribution of errors and therefore a better comparability between sessions. Furthermore, it reduced the setup time, and thereby allowed us to record data from a larger number of participants. It should be noted that the existing ErrP classifier we describe in the previous section can also be operated in online mode, therefore the results from the experiment can be generalized to an end-to-end system with EEG-based classifier.

Next, we defined the notion for the different recovery strategies we analyzed. The most basic strategy is the `REPROMPT` strategy. `REPROMPT` reacts to a detected error by prompting the user to repeat the input. The initial gesture data is discarded and the second gesture is used as the final classification result. We did not repeat the correction procedure after the first repetition as frequent handling of the same error might lead to unexpected EEG signal patterns. The `2ND-BEST` strategy is a modification of `REPROMPT` in that it does not always reprompt if an error is detected. Instead, it inspects the probability distribution of the remaining classes without the allegedly rejected one and picks the now highest scoring class (i.e., the originally second best). However, this estimate might be unreliable as it is based on a probability distribution that has just been indicated as erroneous by the ErrP classifier. Therefore, we only used the second best class if its re-normalized confidence (i.e., probability mass of the first best class distributed equally across all remaining classes) is above a certain threshold T (which is varied during evaluation). Otherwise, the user was asked to repeat the input. Figure 5 shows the control flow of both correction strategies (`REPROMPT` is a special case of `2ND-BEST` with threshold $T = \infty$).

As we also want to compare the automatic correction strategies with user-triggered correction, we define the `MANUAL` strategy that requires the user to actively report a system error. This could for example be executed with a "manual override" command or a correction gesture. In our experiments, we used a two-handed keyboard command issued by the user as trigger. When triggered, the system performed a reprompt of the last trial. This strategy has the advantage of near-perfect recognition of error events but does impose additional overhead for the user. Note that the `MANUAL` strategy still allows only one correction attempt per trial and can therefore still result in recognition errors. This restriction keeps `MANUAL` comparable to the other strategies that also allow at most one correction attempt. We finally call the recovery strategy that does not provide any mechanism to recover from detected error situations `NONE`.

EVALUATION

In this section, we systematically evaluate several aspects of the proposed approach: First, we show that ErrP classification is feasible in the proposed gesture task. Second, we introduce our simulation approach to evaluate different correction strategies. Third, we present the simulation results on correction accuracy and correction costs. Fourth, we show how the simulation results can be generalized to systems at different performance levels. Fifth, we show results of a user study comparing correction strategies.

ErrP Classification on Gesture Task

To understand how the existing ErrP classifier generalizes for more realistic HCI applications, we investigated how the

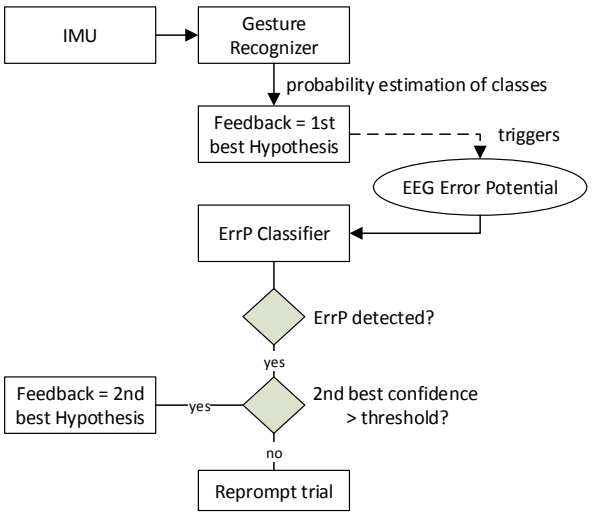

Figure 5. Flowchart of the `2ND-BEST` strategy.

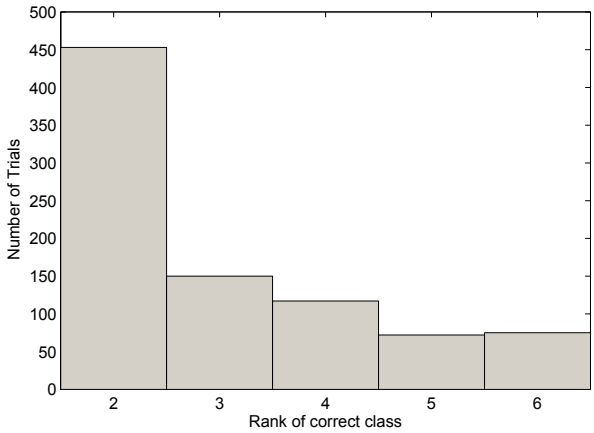

Figure 6. Histogram of ranks of the correct gesture class within the recognition results. Ranks are ordered by assigned probability (only for originally misclassified trials).

trained model transferred to data from the actual gesture scenario. As Iturrate et al. [6] showed, transfer between different tasks may result in performance degradation as characteristics of the ErrP patterns, for example onset latency, change. Therefore, we recorded the *Gesture-ErrP* corpus, which contains data from 11 participants performing the gesture task while wearing the EEG headset. Feedback was presented in the form of a pictogram symbolizing the selected class. It was presented on a fixed position (to avoid systematic eye movement artifacts) as a large, high-contrast overlay and participants were instructed to pay attention to the feedback to improve the recognition accuracy. This procedure was chosen to increase the likelihood and timing precision of the generated ErrP, as participants will not miss feedback or perceive it with a delay. We then evaluated the system trained on the *BCI-ErrP* corpus on the new data set. Using 60 calibration trials, we achieved an accuracy of 78.3%, corresponding to an F-score of 0.69, a precision of 0.72 and a recall of 0.67. This result shows that recognition of ErrPs in the presented real-life gesture scenario is feasible with limited calibration time. While those values are a bit lower than the values assumed for the self-correcting interface, it is known from Putze et al. [12] that accuracy can be improved by adding additional calibration trials.

Simulation-based Evaluation of Recovery Strategies

Using the described setup for the gesture task, we recorded IMU data of a total of 20 sessions from 20 participants. All participants were university students or employees. During the experiments, participants first performed a number of training trials and then three blocks with 35 trials each. Between two blocks was a pause of several minutes for the participant to rest. This is the *Gesture-Sim* corpus which we use to evaluate the baseline gesture classifier and to simulate the effects of different recovery strategies.

First, we describe the gesture classification performance in more detail. Inspecting the average confidence values yielded by the gesture classifier, we see a difference of 0.84 vs. 0.64 for the correct and the incorrect results, respectively. The confidence values are indistinguishable for practical purposes with an average p of 0.158 (two-sided t-test on each fold). This result indicates that the confidence value alone is not reliable enough to detect errors on its own. An additional knowledge source, like the proposed EEG-based ErrP detection, is necessary to reliably identify errors. For all misclassified trials, Figure 6 shows a histogram of the position of the correct label within the recognition results for the corresponding trial sorted by probability. We see that of all wrong classification results, in 52.2% of the cases, the second best estimate is the correct one. This indicates that we can successfully pursue a `2ND-BEST` strategy.

In the following, we evaluate the different recovery strategies for their impact on recognition accuracy of the gesture recognizer and the associated costs in the form of additional gestures. We do this in simulation, where we use the actual results from the gesture recognizer for each trial but simulate the ErrP classification and recovery strategy, including any additional gesture classifications during recovery. This method gives us the opportunity to evaluate a large number of recovery strategies with different parameters to study and compare their effects.

To quantify the effect of the different recovery strategies, we define the *corrected recognition accuracy* to be the fraction of correctly identified gestures after applying the effects of any recovery behavior of the system. Corrected recognition accuracy takes into account the error of the gesture recognizer as well as both types of errors of the ErrP classifier (false positives and false negatives). See the appendix for the equation used to calculate corrected recognition accuracy.

We also assess the costs of each correction strategy to measure system efficiency. As a metric, we calculate the costs of a correction as the average number of additional user inputs (gestures or key presses) necessary to perform the correction. As performing a gesture and receiving the corresponding feedback is the most complex operation in our scenario, this cost measure is highly correlated to task execution time. For `REPROMPT`, this amounts to one gesture for each detected error, including false alarms. Those costs are reduced

Metric	Definition
Corrected accuracy	Gesture accuracy after application of correction behavior.
Correction costs	Number of additional gestures for application of correction behavior.
Cost benefit ratio	Ratio of difference between corrected accuracy and accuracy and correction costs.

Table 2. Overview of employed evaluation metrics.

for 2ND-BEST which in favorable cases corrects errors without any additional costs for the user. For MANUAL, we have no false alarms but additional costs for triggering the correction. We favorably assume that the signal to trigger manual correction is always issued correctly by the user. Correction costs can be estimated in cross-validation by counting the number of (automatically or manually triggered) reprompts.

Simulation Results

To assess the performance of the different strategies using the above metrics, we performed a block-wise leave-one-out crossvalidation of the gesture recognition system and calculated a number of statistics on errors and confidence values. The crossvalidation simulates online recognition, i.e., normalization parameters for each trial are only calculated from the data previous to this trial in chronological order. For each fold, we evaluate the gesture classifier on the testing block and estimate corrected recognition accuracy for the different correction strategies. Table 3 summarizes the results, averaged across all folds. We see that for all correction strategies, the corrected accuracy exceeds the baseline accuracy (i.e., NONE) and therefore improves the overall performance of the gesture classifier. As expected, MANUAL yields the highest improvement, followed by REPROMPT. Still, 2ND-BEST ranks only 3.1 to 9.9% worse than REPROMPT (depending on the selected threshold, see below) and provides a statistically significant improvement over NONE (block-wise one-sided paired t-test, $t = 4.38$, $p = 0.002$). A caveat is that while 2ND-BEST fixes a number of erroneously classified gestures, it also reacts to a number of false positives with no chance of recovery (in contrast to REPROMPT which can still generate a valid final result in case of a false positive). To some degree, this is mitigated by the confidence threshold t applied to the second best result: With $T = 0$, the corrected accuracy of 2ND-BEST is 71%, i.e., below the raw accuracy of NONE. Using a confidence threshold of 0.7, the strategy suggested the second best result for 53% of all error trials. This yields a tuning parameter with which the designer can select between different trade-offs between accuracy and correction costs: Table 3 lists results for three different parameter settings to demonstrate this.

Strategy	Corr. Accuracy	Corr. Costs
NONE	77.0%	0
MANUAL	93.0%	0.46
REPROMPT	86.8%	0.34
ROW-COL-REPROMPT	76.4%	0.34
SELECTIVE-REPROMPT	88.1%	0.34
2ND-BEST (T=0.5)	78.2%	0.14
2ND-BEST (T=0.7)	81.4%	0.19
2ND-BEST (T=0.9)	84.1%	0.27

Table 3. Performance measures of different error correction strategies.

It is surprisingly difficult to beat the simple 2ND-BEST strategy (and its special case REPROMPT) in terms of corrected accuracy. We explored a number of other strategies that use elaborate mechanisms to improve the recovery process. For example, the ROW-COL-REPROMPT strategy tries to estimate the correct row or column (by identifying the row or column with the highest cumulative confidence after removing the first best result) from the initial gesture and only reprompts this reduced set. Indeed, corrected accuracy of the gesture recognizer rises to 88% when only the one missing dimension (i.e., row or column) has to be estimated from the reprompt. However, errors in the automatic selection of the correct row or column, which inevitably prevent a successful correction, lead to a non-competitive corrected accuracy of 76.4%, which is worse than NONE. An alternative which performs better is SELECTIVE-REPROMPT. It also limits the number of reprompted items, but selects the three best classes from the initial gesture input, including the one marked as incorrect by the ErrP detector. This leads to a corrected accuracy of 88.1%, reducing the number of errors by 9.8% relative to REPROMPT. However, we achieve this error reduction only by re-arranging those items into one row[3]. Thus, we pay for this benefit by the fact that we are required to re-arrange the matrix for the reprompt (e.g., moving the three candidates to a joint row) to actually benefit from a simplified pointing and classification task. Informal user tests showed that this is highly confusing to users.

Generalizing Simulation Results

Up to this point, we analyzed the benefit of error recovery using the empirically determined gesture recognition accuracy of 77.0% (on average) yielded by the employed gesture recognizer. Similarly, we used performance values for the ErrP classifier which were plausible for the system described in [12]. However, the benefit of error recovery depends on the performance of both components, gesture recognition and ErrP detection: A very high gesture recognition accuracy demands a high ErrP classification accuracy. Otherwise, nearly all detected errors will be false alarms. In case of rather low gesture recognition accuracy, even an ErrP detector with comparably low accuracy can lead to substantial benefits. For this reason, we analyze the system for different performance levels of gesture recognizer and ErrP classifier in Figure 7. For every configuration, we calculate the *benefit cost ratio*, which is the ratio of absolute improvement of correction accuracy to the corresponding correction costs. See the appendix for how to calculate benefit cost ratio.

We can make a number of noteworthy observations: Unsurprisingly, the benefit-cost ratio increased with performance of the ErrP classifier for a fixed accuracy of the gesture recognizer. In contrast, benefit-cost ratio did not develop monotonously for a fixed ErrP classification accuracy. This is because for very low gesture recognition accuracy, the costs of frequent reprompts are high, regardless of ErrP classification performance; for very high gesture recognition accuracy, the ErrP classifier has to be very precise to actually detect

[3] other options like simply discarding the other items or training new models with a-priori probabilities from the original trial lead to worse recognition accuracy.

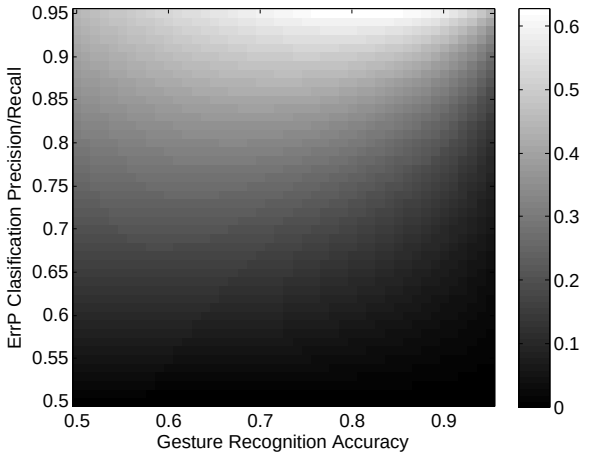

Figure 7. Benefit-cost ratio of `REPROMPT` over `NONE` for different performance levels of gesture recognizer and ErrP classifier.

the few misclassified gestures. Highest benefit-cost ratio can be achieved for gesture recognizers in the accuracy range of 70% to 85%. This is an encouraging result as this is a plausible range for many real-world gesture recognition tasks. Figure 7 can be used to decide whether the application of a self-correcting interface is beneficial for a concrete gesture recognizer and a concrete available ErrP classification accuracy.

In the previous analysis, we treated precision and recall identically. Relaxing this constraint enables the trade-off between precision and recall as an additional design parameter: As false positives and missed errors are not symmetrical in their influence on the performance of a recovery strategy, we now look at the impact of different precision/recall values of the ErrP classifier. For example, it may be favorable to tune the classifier towards a higher precision at the cost of reduced recall, to avoid false alarms which might invalidate correct gesture inputs. In [12], Putze et al. showed that it was possible to achieve a precision of 0.96 by modifying the class balance of training data. Simultaneously, this step reduced recall to 0.76. While this resulted in a slightly lower F-Score of 0.84 compared to the optimal system, it is the number of false positives (which depends on the precision of the classifier) that causes the most trouble for automatic recovery; this is especially true for `2ND-BEST`, which cannot recover successfully in case of a false positive. To investigate the impact of the trade-off between both performance metrics, we adjusted precision and recall of the ErrP detector by $+0.1$ and -0.1, respectively. As a consequence, relative corrected recognition accuracy improved by 3.5% for `REPROMPT` and 6.3% for `2ND-BEST`. The overall ranking of recovery strategies stayed the same, although the gap to the `MANUAL` strategy was reduced, as it did not benefit from the adjustment.

User Satisfaction of Recovery Strategies

The results up to this point indicate that is is possible to develop a self-correcting interface that significantly improves the accuracy of the employed gesture recognizer. However, usability of such a system does not only depend on efficiency and effectiveness and we still have to investigate whether users will accept the different correction strategies. In [7] Jameson systematically analyzed a number of undesired side-effects of introducing such an adaptive interface. He discussed the cause and the effects and ways to remedy them. For application, the most relevant side-effects are "Inadequate Control over Interaction Style" and "Inadequate Predictability and Comprehensibility", as the user has no control over when a correction is triggered and has no way to predict when an error is detected. Another important aspect is "Imperfect System Performance" (of the ErrP classifier), which may lead to negative user experience when a correct interpretation is discarded. To investigate whether those challenges affect user satisfaction, we recorded the *Recovery-Study* corpus, a user study in which we compared `REPROMPT` and `2ND-BEST` (with $T = 0.7$) to the `MANUAL` correction strategy. This strategy selection allowed a comparison of recovery strategies with different levels of autonomy. After a training phase with the gesture recognizer, each participant performed 35 trials for each of the three strategies in random order and filled out a usability questionnaire. Averaged across all participants, raw recognition accuracy was 76.2%. Table 4 describes the actual performance of the different correction strategies. Compared to the simulation results, we can conclude that in simulation we made satisfactory predictions on accuracy and correction costs. Table 5 summarizes the items of the questionnaire and the results. Items were presented with a 5-point Likert-scale, with 1 indicating no agreement and 5 indicating the highest possible agreement. In the following, we analyze the corresponding questionnaire responses. Given the limited sample size, not all results are significant, but the tendencies give a good impression on the perception of the different strategies.

Strategy	Corr. Accuracy	Corr. Costs
`MANUAL`	91.7%	0.52
`REPROMPT`	84.0%	0.37
`2ND-BEST` (T=0.7)	79.7%	0.18

Table 4. Performance measures of the different error correction strategies for the *Recovery-Study* corpus.

Questionnaire Item	`REPROMPT`	`2ND-BEST`	`MANUAL`
felt supported	3.6	3	3.1
system reacts proactively	4.4*	3.5*	2.1
errors corrected reliably	3.1	3.2	2.6
system strenuous	2.9	2.4	2.8
correction tedious	3.0	2.4*	3.5
system predictable	3.4	3.1	2.5
system impedes user	2.5	2.4	2.9
system confusing	2.2*	2.9	3.0
system intuitive	4.6*	4.3	4.1
user has control	4.0*	3.4	3.3
felt observed	1.3	1.8	1.8
pleasant experience	3.3	3.7	3.0

Table 5. Subjective evaluation of correction strategies. 1 = strong rejection, 5 = strong agreement. An asterisk denotes a significant difference (two-sided, paired t-test, $\alpha = 0.05$) between `MANUAL` and the respective automatic strategy.

We see that users are not daunted by the self-correcting interfaces. By tendency, the self-correcting systems are evaluated more positively compared to the system with man-

ual correction regarding all presented questionnaire items. However, there was a difference between REPROMPT and 2ND-BEST in which items they more distinctively differed from MANUAL. Due to the reduced number of manual interventions necessary, automatic correction by the 2ND-BEST strategy was perceived as less strenuous, less tedious and more pleasant than manual correction. Comparing both automatic strategies regarding ease-of-use, users preferred the REPROMPT strategy. REPROMPT was evaluated as the least confusing, most predictable, most intuitive strategy. For REPROMPT, there also was a stronger perception of proactive behavior compared to both other strategies. The reason we see for this difference between both self-correction strategies is that 2ND-BEST provided the more complex behavior as it adds new elements to the interaction flow, while REPROMPT only repeats elements which are already familiar to the user. Both behaviors also differ in their handling of false positives. While reprompting a user after presented feedback can be interpreted as a confirmation of an unreliable classification, the 2ND-BEST behavior explicitly discards the initial classification result and gives the user no opportunity to counter this behavior.

Interestingly, REPROMPT and 2ND-BEST were both perceived as (slightly) more reliable in terms of error correction than MANUAL (3.1 and 3.2 vs. 2.6). This result is in contrast to the higher corrected accuracy of MANUAL. Based on post-hoc interviews with the participants, we postulate that this is because participants did not attribute the high corrected accuracy of MANUAL to the system, but to their own additional effort to report errors.

CONCLUSION

In this paper, we showed that it is possible to improve accuracy of a gesture recognizer using ErrP classification to enable pro-active recovery from recognition errors. We discussed all components necessary for an end-to-end error-aware interface and evaluated different recovery strategies, looking at both objective and subjective evaluation metrics. We showed that an error-aware interface can achieve corrected recognition accuracy close to manual correction, while maintaining much lower correction costs. We compared different recovery strategies, showing a trade-off between achieved corrected accuracy and recovery costs as well as differences in subjective assessment between strategies.

In this paper, we did not only report results for one specific self-correcting gesture recognizer but for a whole family of recognizers with different (gesture and ErrP) accuracy levels and correction strategies. In the discussion of related work we argued that the concrete design of error recovery strategies depends on the employed modalities and recognizers. The methods that we employed in this paper enable us to assess the benefit of different strategies in different situations: The simulation-based evaluation of recovery strategies can be easily extended to other strategies and recognizers. In addition to the simulation, the user study in the *Recovery-Study* showed a number of general criteria (e.g., intuitiveness, perceived performance, control) by which different recovery strategies can be evaluated for user satisfaction. When transferring the results of this paper to another modality or domain, one may have to adjust the recovery strategies and consequently the evaluation of Equation 1. Another aspect that needs to be adapted to new application areas is the employed cost model. It may require adjustments as requesting additional user input may be more or less costly (e.g., finger flicks vs. full-arm movements) or measured on a completely different scale (e.g., time, as some gestures take longer to perform than others) than in this paper. This may also change the user preference for different recovery strategies.

One limitation of the described approach is that it relies on time-locked evaluation of EEG data relative to feedback presentation. This works well for situations were the system can give instantaneous feedback in an unambiguous way, for example global feedback on a graphical user interface. Detection of error states becomes more challenging when feedback is given more locally, is not obviously erroneous or is spreading across longer periods of time. In such cases, the ErrP classification has to rely on additional information sources (e.g., eye tracking to estimate when a presented feedback was perceived) or become less reliant on temporal alignment (e.g., by using methods by Marathe et al. [10]). Another limitation is induced by the fundamental challenges of BCIs: Long setup time, low signal-to-noise ratio, requirement for additional equipment. Those challenges may restrict the applicability to scenarios in which such characteristics are acceptable and are outweighed by the benefits of the error correction. Finally, the work needs to be extended to a full end-to-end system. Such a system would feature additional interactions between the different components, for example potential changes in ErrP patterns in reaction to system error recovery.

APPENDIX

This appendix gives the equations for the calculation of corrected accuracy and benefit cost ratio when employing different correction strategies. The *corrected accuracy* can be calculated as follows:

$$\begin{aligned}\hat{a} =& a \cdot (1 - p_{cor}) + a \cdot p_{cor} \cdot a_{FP} \\ &+ (1 - a) \cdot p_{incor} \cdot a_{TP}\end{aligned} \tag{1}$$

In Equation 1, p_{cor} is the probability of detecting an ErrP if the gesture input was classified correctly (i.e., the false positive rate of the ErrP classifier) and p_{incorr} is the probability of detecting an ErrP if the gesture input was classified incorrectly (i.e., the precision of the ErrP classifier). a is the raw accuracy of the gesture recognizer and can be estimated during crossvalidation. a_{TP} and a_{FP} are the probabilities of a successful recovery when an ErrP was identified correctly (i.e., true positive) or incorrectly (i.e., false positive) for the initial gesture input. For REPROMPT, we simply have $a = a_{TP} = a_{FP}$, as every detected ErrP leads to an additional unmodified gesture trial. For a pure 2ND-BEST strategy (i.e., $T = 0$), we have $a_{TP} = p_{2ND}$ and $a_{FP} = 0$, where p_{2ND} is the probability that the second best estimate is correct, given the first one was already excluded. p_{2ND} can be estimated from the histogram in Figure 6 as the fraction

of second best results of all wrongly classified gesture trials (52.2% in our case). For `2ND-BEST` with $T > 0$, we have:

$$\begin{aligned} a_{TP} &= P(c_{2ND} \leq T) \cdot a + P(c_{2ND} > T) \cdot p_{2ND} \\ a_{FP} &= P(c_{2ND} \leq T) \cdot a \end{aligned} \tag{2}$$

In Equation 2, c_{2ND} is the re-normalized confidence of the second best recognition result (i.e., the probability mass of the first best recognition result is distributed proportionally across the other results). Again, the necessary probabilities can be estimated from the histogram in Figure 6 by counting only the trials with high confidence.

To quantify the efficiency of a correction strategy, we define the *benefit cost ratio* b as the ratio between absolute improvement in accuracy $\hat{a} - a$ and the correction costs c. b can be interpreted as the expected improvement in gesture recognition accuracy per necessary additional gesture. To simplify the analysis, we set $q = p_{cor} = 1 - p_{incorr}$, i.e., we used identical values for precision and recall of the ErrP classifier. Using the definition of b, c and q in Equation 1, we can calculate the benefit-cost ratio as follows:

$$b = \frac{\hat{a} - a}{c} = \frac{(2q-1)(a-a^2)}{a(1-q)+(1-a)q} \tag{3}$$

REFERENCES

1. Bohus, D., and Rudnicky, A. I. Sorry, i didn't catch that! In *Recent Trends in Discourse and Dialogue*, no. 39 in Text, Speech and Language Technology. Springer Netherlands, 2008, 123–154.
2. Combaz, A., Chumerin, N., Manyakov, N. V., Robben, A., Suykens, J. A. K., and Van Hulle, M. M. Towards the detection of error-related potentials and its integration in the context of a p300 speller brain–computer interface. *Neurocomputing 80* (2012), 73–82.
3. Ferrez, P. W., and José del R. Millan. Error-related EEG potentials generated during simulated brain computer interaction. *IEEE Transactions on Biomedical Engineering 55*, 3 (2008), 923–929.
4. Förster, K., Biasiucci, A., Chavarriaga, R., Millan, J. d. R., Roggen, D., and Tröster, G. On the use of brain decoded signals for online user adaptive gesture recognition systems. In *Pervasive Computing*, no. 6030 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, 427–444.
5. Große, P. W., Holzapfel, H., and Waibel, A. Confidence based multimodal fusion for person identification. In *Proceedings of the 16th International Conference on Multimedia* (New York, USA, 2008).
6. Iturrate, I., Chavarriaga, R., Montesano, L., Minguez, J., and Millan, J. d. R. Latency correction of error potentials between different experiments reduces calibration time for single-trial classification. *Proceedings of Annual International Conference of the Engineering in Medicine and Biology Society. 2012* (2012), 3288–3291.
7. Jameson, A. D. Understanding and dealing with usability side effects of intelligent processing. *AI Magazine 30*, 4 (2009), 23–40.
8. Krusienski, D. J., Sellers, E. W., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R. Toward enhanced p300 speller performance. *Journal of Neuroscience Methods 167*, 1 (2008), 15–21.
9. Llera, A., van Gerven, M. A. J., Gómez, V. M., Jensen, O. K., and Kappen, H. J. On the use of interaction error potentials for adaptive brain computer interfaces. *Neural Networks 24*, 10 (2011), 1120–1127.
10. Marathe, A. R., Ries, A. J., and McDowell, K. A novel method for single-trial classification in the face of temporal variability. In *Foundations of Augmented Cognition*, no. 8027 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, 345–352.
11. Margaux, P., Emmanuel, M., Sébastien, D., Olivier, B., and Jérémie, M. Objective and subjective evaluation of online error correction during p300-based spelling. *Adv. in Hum.-Comp. Int. 2012* (2012).
12. Putze, F., Heger, D., and Schultz, T. Reliable subject-adapted recognition of EEG error potentials using limited calibration data. In *6th International Conference on Neural Engineering* (San Diego, USA, 2013).
13. Schmidt, N. M., Blankertz, B., and Treder, M. S. Online detection of error-related potentials boosts the performance of mental typewriters. *BMC Neuroscience 13* (2012), 13–19.
14. Scholl, P. M., van Laerhoven, K., Gordon, D., Scholz, M., and Berning, M. jNode: A sensor network platform that supports distributed inertial kinematic monitoring. In *Proceedings of the Ninth International Conference on Networked Sensing Systems* (2012).
15. Spüler, M., Bensch, M., Kleih, S., Rosenstiel, W., Bogdan, M., and Kübler, A. Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a p300-BCI. *Clinical neurophysiology: official journal of the International Federation of Clinical Neurophysiology 123*, 7 (2012), 1328–1337.
16. Stoyanchev, S., Salletmayr, P., Yang, J., and Hirschberg, J. Localized detection of speech recognition errors. In *Proceedings of the Spoken Language Technology Workshop* (2012), 25–30.
17. Vi, C., and Subramanian, S. Detecting error-related negativity for interaction design. In *Proceedings of the Conference on Human Factors in Computing Systems* (New York, USA, 2012).
18. Vu, N. T., Kraus, F., and Schultz, T. Multilingual a-stabil: A new confidence score for multilingual unsupervised training. In *Proceedings of the Spoken Language Technology Workshop* (2010), 183–188.