

# Simulating the Feel of Brain-Computer Interfaces for Design, Development and Social Interaction

Melissa Quek,<sup>1</sup> Daniel Boland,<sup>1</sup> John Williamson,<sup>1</sup> Roderick Murray-Smith,<sup>1</sup> Michele Tavella,<sup>2</sup> Serafeim Perdikis,<sup>2</sup> Martijn Schreuder,<sup>3</sup> Michael Tangermann<sup>3</sup>

<sup>1</sup>School of Computing Science  
University of Glasgow  
Glasgow, Scotland  
{melissa, jhw,  
rod}@dcs.gla.ac.uk  
email@dannyboland.com

<sup>2</sup>School of Engineering  
École Polytechnique Fédérale de  
Lausanne  
Lausanne, Switzerland  
{michele.tavella,  
serafeim.perdikis}@epfl.ch

<sup>3</sup>Machine Learning Laboratory  
Berlin Institute of Technology  
Berlin, Germany  
{schreuder,  
michael.tangermann}  
@tu-berlin.de

## ABSTRACT

We describe an approach to improving the design and development of Brain-Computer Interface (BCI) applications by simulating the error-prone characteristics and subjective feel of electroencephalogram (EEG), motor-imagery based BCIs. BCIs have the potential to enhance the quality of life of people who are severely disabled, but it is often time-consuming to test and develop the systems. Simulation of BCI characteristics allows developers to rapidly test design options, and gain both subjective and quantitative insight into expected behaviour without using an EEG cap. A further motivation for the use of simulation is that ‘impairing’ a person without motor disabilities in a game with a disabled BCI user can create a level playing field and help carers empathise with BCI users. We demonstrate a use of the simulator in controlling a game of Brain Pong.

## Author Keywords

BCI, HCI, simulation, usability, social interaction, disability, motor imagery, SMR

## ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous

## General Terms

Design, Performance, Human Factors

## INTRODUCTION

A Brain-Computer Interface (BCI) is a system that harnesses a person’s brain activity to interact with a computer or other electronic device. For people who are severely paralyzed, a BCI might provide the only means of communication; more commonly it potentially serves as an additional channel of input for those with a very limited control of movement.

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

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

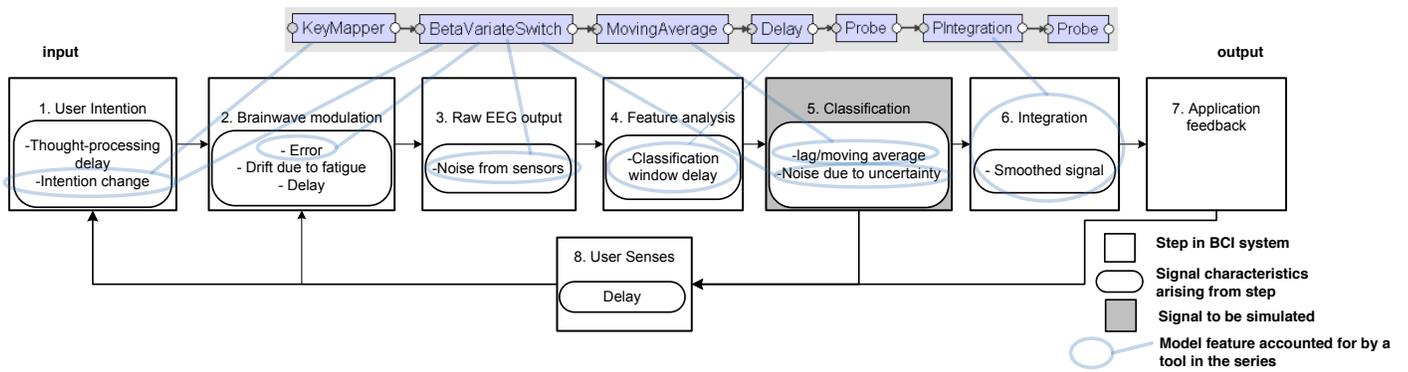
Copyright 2011 ACM 978-1-4503-0267-8/11/05...\$10.00.

A problem with testing and debugging BCI applications is that it can take a long time – up to an hour – to set up and calibrate a BCI. The input is also slow compared to most other methods and thus time consuming to run user trials: the highest performance reported for motor-imagery BCI so far is 40 bits/min [5], while a fast speller yields 7 chars/minute [12]. This note describes how modelling the classifier output of a motor-imagery BCI can be used to simulate interaction characteristics such as error rate, delay and noise. We aim to use the simulator to speed up and improve the development process, to enhance social interaction and to aid understanding of the constraints and skill involved in using a BCI.

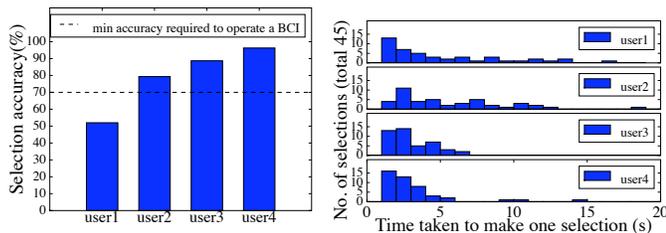
## BACKGROUND AND MOTIVATION

The electroencephalogram (EEG) is a technique used to measure electrical brain activity arising from a large number of neurons near the surface of the brain. It has been used for brain-computer communication in recent years [10]. Typically, a set of electrodes is gelled to the scalp with the help of a positioning cap. Certain *mental states* that a BCI user produces at will are recognized through machine learning techniques and mapped to interface controls. For example, a person might continuously do complex sums in one’s head (the mental state) in order to move a ball to the left of the screen (the control). The current simulator is modelled on BCIs driven by *motor-imagery*, which refers to the imagining of moving a body part [10]. The feet and hands are the most commonly used body parts in current BCI literature.

**Characteristics of a motor-imagery based BCI.** A typical BCI system goes through the general process shown in Figure 1. The user intends an action (step 1), for example, to select left. ‘Brainwave modulation’ (step 2) refers to an attempt to produce a mental state that can be detected by the classifier. The raw EEG signals are recorded (step 3), features of the signals are selected (step 4) and subsequently *classified* (separated), into two or more mental states, or *classes* (step 5). Two classes are most commonly used allowing for binary input, although multi-class BCIs have also been demonstrated [5]. The noisy output of the classifier is usually integrated (smoothed), to lessen errors (step 6). Finally, the application presents the system’s belief about the user’s intention (step 7).



**Figure 1. Top series (shaded background):** Screen shot of part of the GUI used to create a simulator model, showing the tools used in the current model. **Bottom:** Stages in a BCI control loop giving rise to low-level characteristics of the BCI, some of which are modelled in the current simulator.



**Figure 2. Individual differences in BCI control.** Left: Bar chart of selection accuracy for 4 users (num correct trials ÷ total trials). Right: Histogram showing distribution of times taken to make a single selection (i.e. move the cursor to the left or right) for the same 4 users.

Individual differences in motor-imagery based BCI control are pronounced, with an estimated 15–30% of subjects being unable to operate such a BCI [4]. Figure 2 shows two control characteristics from the calibration data used in our current model, giving some examples of the error rate and variability in selection time. The minimum accuracy required for being able to communicate using a BCI is generally taken to be 70% [10]. False positives, where the system makes a selection despite the user not wishing to select anything, are a problem for current development of BCI applications because of the ‘continuous’ production of the brain signal. There is a delay between a user ‘switching’ mental states and its effect being displayed on the screen, and users also differ in their ability to sustain a mental state (continue imagining the movement) such that it can be accurately classified.

**Simulation for design and understanding impairment.** Designing for disabled people can be difficult for those who do not have the same disabilities. Simulation can help designers to understand what a particular impairment feels like in order to provide solutions to problems. For example, the Third Age Suit [6] is a wearable outfit which simulates the restricted movement experienced by the elderly for the purpose of design. In the context of BCI, low-level simulation can potentially enable people to experience the frustration of using a BCI without needing to wear an EEG cap. On the other hand, learning to achieve control of a BCI can be fun and challenging, so in the same way that flight simulators are entertaining for people who might never become a pilot, understanding the skill required to control a BCI through using a simulator could bring respect for a trained BCI user.

**Simulation to predict usability.** Simulation is useful for improving user interfaces for Assistive Technologies (ATs) because of the large variability in the population of users. Simulators incorporating perceptual, cognitive and motor models to predict usability of interfaces have been developed (e.g. [3]), while simulating motor disability by introducing delays into keyboard input produced useful guidelines in a word selection experiment [7]. However, the BCI characteristics mentioned previously require different models than those for input devices that are commonly investigated. The few simulation and modelling efforts in BCI focus on the number of decisions to make in order to reach a target (e.g. [2]), however in BCI this may not necessarily correspond to the time taken to make them. In addition, this method does not allow analysis of interfaces based on continuous interfaces, such as hex-o-spell [12]. Low-level characteristics of BCI can be simulated to explore these at a lower cost.

**Simulation for testing and evaluating interfaces.** Conducting user trials with BCI is expensive: it can be difficult to find people who have been sufficiently trained for testing the interfaces effectively, and it is undesirable to allow disabled people to sit for long periods of time with an EEG cap on, which can become uncomfortable. Since a BCI may be much slower and more error-prone than people with disabilities may be used to, it is important to obtain feedback about applications using BCI input. By using a simulator, user evaluation and testing can take place early on in the development cycle, affording BCI design the rapid prototyping methods that are taken for granted in mainstream HCI.

**Simulation to support social interaction.** Through interviews and focus groups carried out within the TOBI Project<sup>1</sup>, disabled people have indicated a significant interest in interacting more with other people through playing games. One way of achieving this might be to ‘impair’ the non-disabled player in order to establish a level playing field. For example, the keyboard or mouse input could be modified to emulate the BCI user’s input, with similar error rate and timing. In addition to making the games fairer for both parties, this could help the non-disabled player to empathize with the disabled player.

<sup>1</sup><http://tobi-project.org>

## SIMULATOR MODEL AND ARCHITECTURE

The current simulator was developed using data and information from BCI researchers at École Polytechnique Fédérale de Lausanne (EPFL) [8]. Data from 5 users was taken from the usual procedure used to train a user, which is briefly described as follows. A user is asked to imagine moving a limb (e.g. their right hand), in order to move a screen ‘cursor’ to the right, and to imagine moving another limb to move it to the left. At a rate of 16Hz, EEG features are computed and classified into the left or right classes. The output of probabilities at each time step indicate the system’s belief that the person is imagining movement of the body part(s) assigned to these classes ( $\sim 1$ =right,  $\sim 0$ =left). This noisy signal is integrated to give a smoother signal. The horizontal position of the cursor correlates with the value of the integrated classifier: when this reaches a user-dependent threshold, the bar reaches the corresponding side and a selection is made. This last integrated value is the output of the simulator since our aim is to simulate the subjective feel of the interaction.

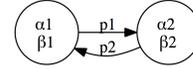
We developed a simulation toolkit in Python where signal generation or modulation blocks, or *tools*, are linked together in a series. Tools can be replaced, added or removed easily. This is useful for improving the model as well as to simulate different BCI systems that use different techniques. Each modulation tool receives a signal from the preceding tool, modifies it and passes it along to the next tool in the series. The value of a signal at a particular point in the sequence can be captured using the *Probe* tool. The output of real BCIs is replaced seamlessly by sending these discrete values over the network to the application, using the same protocols as the real BCI. A GUI was developed in order to easily change the tools used in a model, their parameters and order in the series. We now describe in more detail how the tools work and relate to the current model, following the tool series in Figure 1 (top) from left to right.

The *KeyMapper* tool assigns a keyboard key to a particular state, for example the left arrow key to the ‘left’ state. To change the properties of the signal, the parameters for other tools in the model are updated when the state of the system changes. The *BetaVariateSwitch* tool is the core tool in the current simulator model which generates both the signal and noise in this system. Histograms of the classifier values indicate that at a given time step, the classifier believes that the features belong more strongly to one class or the other (i.e. the output of the classifier ranges from 0 to 1, and is biased towards either 0 or 1). Samples are drawn from a Beta distribution whose probability density function is given by

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} [1]$$

Essentially, the drawn sample can be biased more or less strongly towards 0 or 1, e.g. for low  $\alpha$  ( $< 1.0$ ) and high  $\beta$  ( $> 1.0$ ), most of the values drawn will be closer to 0 than 1, and vice versa. To obtain a noisy signal, we switch between two sets of opposing  $\alpha$  and  $\beta$  values. A Markov chain is employed to influence the probability of and time taken for the integrated signal to reach the desired class threshold. The probability of switching from one set of values to another

depends on the current set (Figure 3); adjusting these 6 parameters enables simulation of a wide range of performance.

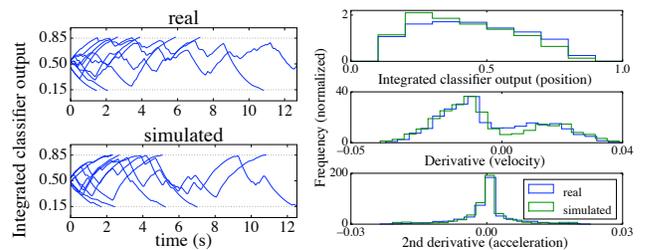


**Figure 3. Finite state machine showing the Markov chain used in the *BetaVariateSwitch* tool: the probability of switching from one set of  $\alpha$ ,  $\beta$  parameters to another is dependent on the set currently employed.**

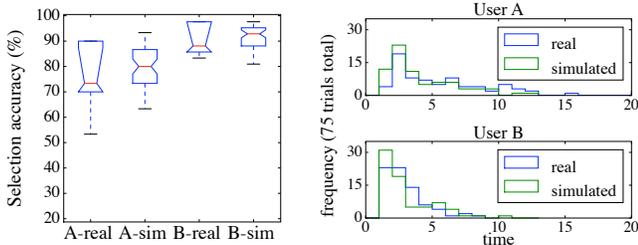
Next, the *MovingAverage* tool implements an  $n$ th order moving average which arises from the feature extraction stage. The *Delay* tool accounts for the time delay due to the feature extraction and classification process. This was estimated to be 0.5s, as the output at each time step takes into account the last 1s of data; it seems reasonable that a change in state would be seen from around half this time. A Gamma distribution was used to model the delay that occurs when switching from one mental state to another, as a large variance was expected. The initial parameters were chosen by a reasonable estimation and optimized automatically. Finally, the *PIntegration* tool integrates the classifier output that has been generated. This is a direct implementation of the integration method used at EPFL [9].

Keyboard input was used rather than mouse as it is more appropriate to generate the classifier output signal based on discrete input to represent the switching between mental states, rather than a continuous input. In this first approximation, some BCI characteristics not accounted for are the ability to hold a mental state once the threshold is reached, the closed-loop interaction between the subject’s thoughts and the influence of feedback, and the longer term deterioration in performance that occurs due to fatigue or stress [4]. However, the flexibility of the simulator allows additional tools to be added as the model is improved.

**Optimization of parameters.** The parameters of the model are optimized for the subjective feel of the BCI. Based on the assumption that the 2nd derivative of the signal (i.e. acceleration of the cursor) corresponds closely to subjective feel of movement, we measure the difference between the histograms of the 2nd derivatives of the time series for real and simulated data. This is added to the difference in selection accuracy,  $x^{difference}$ , where  $x$  is determined by trial and error. The result is used as a cost function to be minimized using Powell’s method. The time series in Figure 4 shows that the integrated classifier output generated for one user using this method subjectively matches the real data; com-



**Figure 4. Comparison of real and simulated BCI signals for one user. Left: Time series of the integrated classifier output. This is reset to 0.5 at the start of every trial; a selection is made when the output reaches a threshold (0.85 or 0.15, dotted lines). Right: Comparison of the average position, velocity and acceleration for real and simulated signals for correct left trials, showing a close match between the two signals.**



**Figure 5.** Selection accuracy (left) and histograms of time to make a selection (right) for real and simulated trials for 2 users A and B, indicating a reasonable approximation of the distributions for both users.

parison of the histograms of the 1st and 2nd derivatives verifies this. Figure 5 also shows that the selected parameters allow a match of the selection accuracy and time distributions, justifying the model and our implementation choices.

### EXAMPLE APPLICATION

Although many BCI applications currently employ discrete selection as input, modelling the continuous control of a BCI is important as novel interaction techniques might improve performance. For example, the fastest motor-imagery BCI speller to date, hex-o-spell, uses the time taken to switch between mental states [12]. We used a version of the simulator adapted for a BCI research group at the Berlin Institute of Technology as input to *Brain Pong*, a BCI version of the classic game *Pong*. This directly replaced the BCI input expected by the application to control a paddle (Figure 6). The game used the Pythonic Feedback Framework [11], a BCI application platform used by several BCI labs. Expert BCI users asserted that the behaviour and subjective feel (i.e. delay, noise and error) of controlling the paddle seemed sufficiently similar to that when using a real BCI to be used for design purposes. This preliminary finding provides a credible basis for carrying out a more rigorous validation. Using a simulator adapted to match BCI characteristics in real time, a non-disabled person could play games either face-to-face or over the Internet with a BCI user, which could help to provide fairer competition.



**Figure 6.** Left: Brain Pong with a real BCI. Right: Brain Pong with keyboard input modified by the simulator.

### CONCLUSIONS

We have built a simulator that models the essential characteristics of motor-imagery based BCIs that are relevant to the design and development of BCI applications. As far as we know, this is the first attempt to model the low-level characteristics of a BCI for interface design purposes. Performance models for mouse, keyboard and single switches are mostly used to automatically evaluate interfaces (e.g. [3]), while usability studies which employ real time simulators to gain information about task performance also focus on standard inputs [7], and thus far do not appear to match the subjective

feel of the input with accurate performance characteristics. Objective and subjective evaluation of the output of our simulator provide a basis for using it to predict usability of interfaces both offline through simulation and online with user studies. Simulation can allow interaction designers to explore BCI characteristics with little effort, and afford end user engagement on issues that arise from the input constraints. We also proposed that the simulator be used to create a more level playing field between disabled and non-disabled people, and validated the feasibility of this with the simulator-controlled game of Brain Pong.

### ACKNOWLEDGEMENTS

Andrew Ramsay for the initial GUI. This work is supported by the European ICT Programme Project FP7-224631 and the PASCAL2 EC Network of Excellence. This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

### REFERENCES

1. J. Banks, B. Nelson, and D. Nicol. *Discrete-event system simulation*. Prentice Hall, 2009.
2. M. Bensch, A. Karim, J. Mellinger, T. Hinterberger, M. Tangermann, M. Bogdan, W. Rosenstiel, and N. Birbaumer. Nessi: an EEG-controlled web browser for severely paralyzed patients. *Comp. Intelligence and Neurosci.*, 2007:71863, 2007.
3. P. Biswas and P. Robinson. Automatic evaluation of assistive interfaces. In *Proc. 13th Int. Conf. on Intelligent User Interfaces*, pages 247–256. ACM, 2008.
4. B. Blankertz, C. Sannelli, S. Halder, E. Hammer, A. Kübler, K. Müller, G. Curio, and T. Dickhaus. Neurophysiological predictor of SMR-based BCI performance. *NeuroImage*, 51(4):1303–1309, 2010.
5. G. Dornhege. *Increasing information transfer rates for brain-computer interfacing*. PhD thesis, University of Potsdam, Potsdam, 2006.
6. D. Hitchcock, S. Lockyer, S. Cook, and C. Quigley. Third age usability and safety—an ergonomics contribution to design. *Int. J. Hum.-Comput. Stud.*, 55(4):635–643, 2001.
7. J. Mankoff, H. Fait, and R. Juang. Evaluating accessibility by simulating the experiences of users with vision or motor impairments. *IBM Syst. J.*, 44(3):505–517, 2010.
8. J. Millán, F. Renkens, J. Mouriño, and W. Gerstner. Brain-actuated interaction. *Artificial Intelligence*, 159(1-2):241–259, 2004.
9. S. Perdakis, H. Bayati, R. Leeb, and J. Millán. Evidence accumulation in asynchronous BCI. *Proc. TOBI workshop*, 2010 (to appear).
10. G. Pfurtscheller, C. Neuper, and N. Birbaumer. Human brain-computer interface. *Motor cortex in voluntary movements*, pages 367–401, 2005.
11. B. Venthur, S. Scholler, J. Williamson, S. Dähne, M. Treder, M. Kramarek, K. Mäller, and B. Blankertz. Pyff—A Pythonic Framework for Feedback Applications and Stimulus Presentation in Neuroscience. *Front. Neurosci.*, 4(179), 2010.
12. J. Williamson, R. Murray-Smith, B. Blankertz, M. Krauledat, and K. Müller. Designing for uncertain, asymmetric control: Interaction design for brain-computer interfaces. *Int. J. Hum.-Comput. Stud.*, 67(10):827–841, 2009.