Available online at www.sciencedirect.com
ScienceDirect
Int. J. Human-Computer Studies 67 (2009) 827–841
International Journal of Human-Computer Studies
www.elsevier.com/locate/ijhcs

Designing for uncertain, asymmetric control: Interaction design for brain–computer interfaces ☆

J. Williamson[a,*], R. Murray-Smith[a,b], B. Blankertz[c], M. Krauledat[c], K.-R. Müller[c]

[a]*Department of Computing Science, University of Glasgow, 18 Lilybank Gardens, G12 8QQ Glasgow, UK*
[b]*Hamilton Institute, National University of Ireland Maynooth, Maynooth, Ireland*
[c]*Berlin Institute of Technology, Berlin, Germany*

Received 30 September 2008; received in revised form 11 May 2009; accepted 23 May 2009
Communicated by A. Steed
Available online 17 June 2009

Abstract

Designing user interfaces which can cope with unconventional control properties is challenging, and conventional interface design techniques are of little help. This paper examines how interactions can be designed to explicitly take into account the uncertainty and dynamics of control inputs. In particular, the asymmetry of feedback and control channels is highlighted as a key design constraint, which is especially obvious in current non-invasive brain–computer interfaces (BCIs). Brain–computer interfaces are systems capable of decoding neural activity in real time, thereby allowing a computer application to be directly controlled by thought. BCIs, however, have totally different signal properties than most conventional interaction devices. Bandwidth is very limited and there are comparatively long and unpredictable delays. Such interfaces cannot simply be treated as unwieldy mice. In this respect they are an example of a growing field of sensor-based interfaces which have unorthodox control properties. As a concrete example, we present the text entry application "Hex-O-Spell", controlled via motor-imagery based electroencephalography (EEG). The system utilizes the high visual display bandwidth to help compensate for the limited control signals, where the timing of the state changes encodes most of the information. We present results showing the comparatively high performance of this interface, with entry rates exceeding seven characters per minute.

Keywords: Brain–computer interaction; EEG; Text entry

1. Introduction

For many years, interaction with a computer has been focused on a few well-understood modalities. Instruments such as the keyboard and mouse, and visual displays have been dominant. In the last decade, however, the potential of unconventional communication channels has become increasingly apparent. A wide variety of sensors and display mechanisms have been experimented with. It is still difficult, however, to reason about the different properties of interfaces with these different channels. There are certain fundamental defining characteristics of human–computer interfaces which can be measured or estimated, and can be used as a consistent framework for design. Among these, the most elementary are bandwidth, dimensionality and delay. This paper explores how these considerations can be used as the basis for rational interface design. From these principles, a state-of-the-art text entry system has been developed. This manages some of the highest entry speeds for brain–computer interfaces of the non-evoked[1] electroencephalography (EEG) type. It is

☆The studies were partly supported by the Bundesministerium für Bildung und Forschung (BMBF), FKZ 01IBE01A, by the SFI (00/PI.1/C067), by EU Project TOBI: Tools for Brain–Computer Interaction, and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

*Corresponding author. Tel.: +44 1413303339; fax: +44 1413304913.
E-mail addresses: jhw@dcs.gla.ac.uk (J. Williamson), rod@dcs.gla.ac.uk (R. Murray-Smith), blanker@cs.tu-berlin.de (B. Blankertz), matthias.krauledat@first.fhg.de (M. Krauledat), klaus-robert.mueller@tu-berlin.de (K.-R. Müller).

[1]Generated by internal thought alone, without external stimuli such as flashing visual indicators.

1071-5819/$ - see front matter
doi:10.1016/j.ijhcs.2009.05.009

designed to cope with the temporal characteristics and noise properties of the EEG control signals rather than being a variant of an existing metaphor.

1.1. Motivation

Direct brain-to-computer interfaces have advanced significantly in the recent past. Non-invasive, EEG-based systems with relatively fast interaction are being developed in several labs throughout the world. Such interactions have almost exclusively been based upon mapping existing metaphors (cursors and button-pushing being the most prominent of these) to the signals that are measured from the brain–computer interface. The question arises as to whether there are better ways of dealing with the curious properties of a brain–computer interface (BCI), ways in which users can express their intentions by reliably controlling the state of the system with which they interact. This question in turn provokes a second: what do experiences with designing interfaces for brain-control reveal about the fundamentals of human–computer interaction?

The purpose of this work is thus twofold: to discuss how brain–computer interaction can be improved by better design and to examine how the interactions should be designed in light of the issues raised. Many elements of HCI consist of carefully adapted special-purpose mechanisms for particular purposes, but given a new control problem it is difficult to utilize these except by *ad hoc* modification. It would be useful to consider, given some knowledge of the control behaviour of an interactor, what the properties of suitable interfaces might be, and what design processes could be used to develop practical systems from these properties.

1.2. The challenges to conventional human–computer interaction

Although proof-of-concept BCIs were demonstrated decades ago (e.g. Elbert et al., 1980), several major challenges remain. One of the most pressing is to develop BCI-driven applications which take the specific characteristics of BCI communication into account. Apart from being prone to error and having a rather uncontrolled variability in timing, the bandwidth of EEG-based brain–computer interfaces is heavily unbalanced: BCI users can perceive a high rate of information transfer from the display, but have a low-bandwidth communication in their control actions (Fig. 1). This asymmetry is unusually extreme in EEG-based BCI, but such imbalance between the inputs a human can apply and the responses a system can make is present in many interfaces. Different interaction styles are suited to different balances of the input and feedback channels.

Where feedback channels are broad in comparison to input channels, more sophisticated models can be simulated within the system and displayed to the user. The system can respond differently to the same control inputs at different times; the mapping from input response can be adapted to disambiguate the user's intention as efficiently as possible. Such systems rely less on user training than where the feedback channels are narrower, where the user

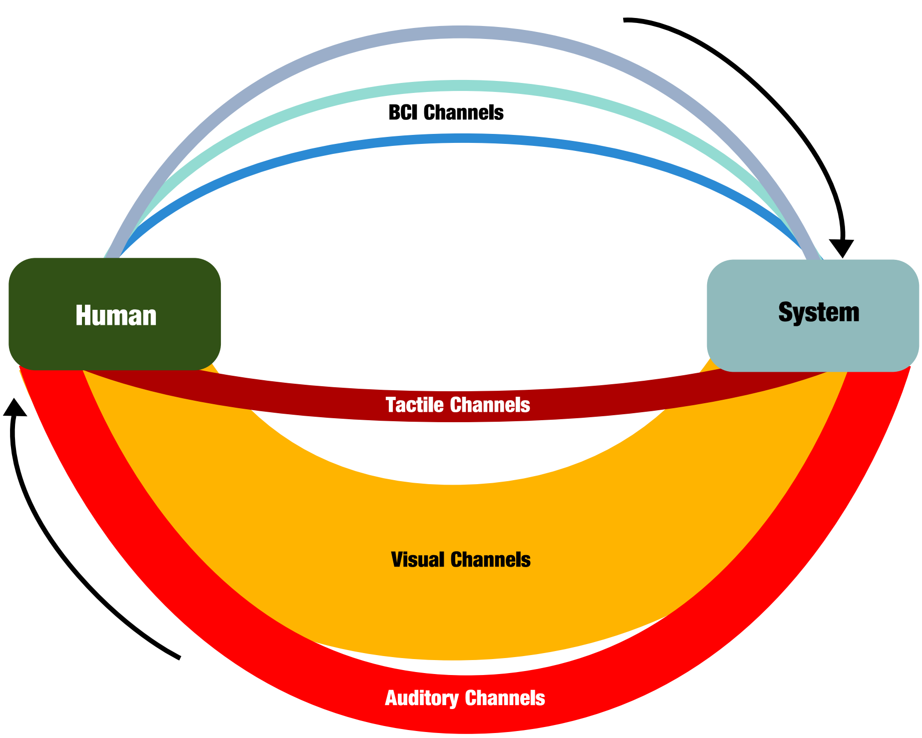

Fig. 1. Asymmetry of BCI communication. The channels available for input are much slower than the channels available for feedback.

must learn the properties of the system in order to be efficient. Typing is an example of an input-bound system, where the feedback is very limited (simple tactile response, possibly with some visual element) and extensive training is required to achieve effective interaction. However, the lack of dependence of feedback means input can be extremely rapid, so long as interaction conditions are constant.

1.3. Background: brain–computer interfaces

Brain–computer interfaces translate brain signals into control signals without intermediate motor action. The measured brain signals contain elements which can be consciously modulated. The resulting signals may be used for a desktop-based computer application or for controlling a neuroprosthesis. There is a variety of BCI systems being developed that use signals recorded from the scalp, the surface of the cortex or from inside the brain. It has been shown that invasive BCI systems enable monkeys, and recently also humans, to operate a robotic arm (Hochberg et al., 2006; Carmena et al., 2003). Furthermore, it was demonstrated that non-invasive BCI systems enable healthy subjects as well as patients to control an internet browser or simple word processing software (Kuebler et al., 2001; Wolpaw et al., 2002; Dornhege et al., 2007a). BCIs have also been successfully used in the restoration of grasp function using functional electrical stimulation (e.g. as described in Mueller-Putz and Pfurtscheller, 2008) and in wheelchair navigation (Millan, 2008).

This has obvious benefits to those physically incapacitated, opening up a channel of control for even the most severely disabled patients. In practice, it has been applied to enable patients suffering from amyotrophic lateral sclerosis (ALS) who have become totally locked-in (i.e. they have no functioning motor control) to communicate with the outside world (Birbaumer et al., 1999). Direct brain control is of interest even for those who are able-bodied; the idea of a direct channel between thought and machine is long-standing technological dream. Current technology limits the richness of the communication, but the potential for tightly bound brain–computer systems is enormous.

2. Designing for uncertain control inputs with unusual dynamics

2.1. Closed-loop interfaces

A computer interface facilitates *control*. It provides a set of mechanisms by which a human can drive the belief of a system about a user's intentions towards a desired state over a period of time. Control requires both display to the user and input from the user; computers feedback state to a user, who modifies his or her actions to bring about the required change of state (Fig. 2). This control operates at multiple scales, forming a hierarchy of loops, from internal muscular control loops to long-term progress towards abstract goals.

The familiar mouse-pointing metaphor is a simple example, where motor control is used to bring two objects (a pointer and a target) into alignment, thus communicating the user's intent to the computer. This superficially trivial mechanism is carefully engineered by interface designers, who manipulate the dynamic response curves of the pointer motion, design the spatial arrangement of targets and shape the hardware to fit with the restrictions of human muscle activity (see for example, Barrett et al., 1995; Isokoski and Raisamo, 2004). Much of this has evolved unconsciously, as designers refine products to subjectively improve user experience. There is surprisingly little systematic analysis of these properties.

2.1.1. Separating communication and control

One of the difficulties in applying information-based measures in interaction design is the confusion between *communication* and *control*. In order to operate a system, the operator must be able to drive the system into the states that the operator wishes the system to enter. Regardless of the potential bandwidth of the input and output channels (how much information can be conveyed in a period of time), the system will function poorly if they cannot be used to reliably steer the system towards desired states. Independent measures of channel capacities only give upper bounds on performance. Introducing a delay in display, for example, does not affect the throughput of the

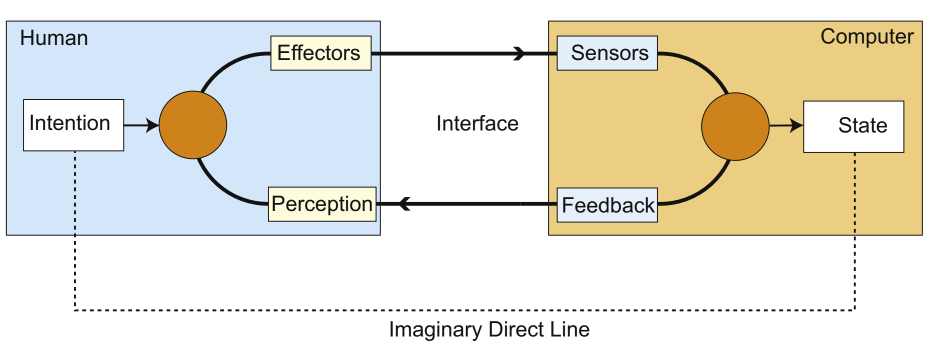

Fig. 2. Interaction as a closed loop control process, with input and feedback channels. Note that in BCIs, effectors are the neuronal assemblies in the cortex, rather than muscles as usual.

feedback channel, but it can significantly degrade control performance.

2.2. Control in a BCI context

New ways of interacting must be devised to make BCI more practical. Twisting existing metaphors to fit the signal properties that occur in brain–computer interfaces is a quick and easy way of building something functional, but it falls far short of the potential for rich brain computer interfacing. The challenge to HCI is to take what is known about the specific control properties of EEG-based BCI and construct a concrete interface from those fundamentals.

There are several important differences between BCI and most other extant interfaces. There is the aforementioned asymmetry of information flow; the response dynamics of the EEG-based BCI are exotic compared to the simple responses of mice, keyboards or touchscreens; and control involves significant levels of uncertainty. EEG-based BCI can be characterized as a slow, noisy, low-dimensional channel with long and variable delays induced by the BCI classification processes and fluctuations of the user itself (Shenoy et al., 2006).

One unusual consideration is the very direct effect of stress and mental load upon the performance of users operating a BCI. Performance is observed to rapidly degrade as mental load increases, possibly due to interference from brain patterns resulting from the stressful situation. It is therefore important to maintain a calm interface which does not result in frustration or require constant, concentrated attention.

2.3. Linking uncertainty and dynamics

Uncertainty and dynamics are intimately linked. Interfaces have dynamics so that communication is spread over time, at a pace that suits the constraints of the human and the system. This distribution is necessary because information cannot be passed instantly through a channel of limited capacity. Furthermore, the comfort of the interacting user is strongly affected by the design of the dynamics; a physically demanding system may be less pleasant to use than the one with calmer or better-matched dynamics. Computer interfaces are unusual in that they have pseudo-physical responses (a pointer moving on screen in response to mouse motion is mapped via some transfer function, for example). These responses, unlike hardware systems, can be manipulated continuously to improve control. This is the basis of enhancements such as semantic pointing (Blanch et al., 2004; Blanch, 2005).

On a fundamental level, the system can link its internal response dynamics to its uncertainty about the user's intent. As the measured signals from the user diverge from patterns characteristic of those associated with functions the system provides, the system can dampen responses. The handling qualities of interface can be manipulated so that when the system is certain, it behaves like a high-performance vehicle, swerving and navigating at the slightest touch. When uncertainty creeps in, the response can gradually calm down, requiring more evidence to effect action. The issues of the "cost" of action and the relation between action, control and belief are discussed in detail in Williamson (2006).

3. Hex: a text entry system for tilt control

These ideas were originally developed while examining the problems of gestural interaction on mobile devices, where problems with noisy sensing and poor user understanding of what movements sensors are actually measuring makes interface design difficult. "Hex" (Williamson and Murray-Smith, 2005) was developed as an example of a system which links the dynamics of a cursor to a probabilistic model. The result is a tilt-based gestural text entry system which adapts the response to sensor input as text is entered.

Hex uses accelerometers to measure tilt, and text is entered by maneuvering a cursor through a hexagonal tessellation, crossing edges which demarcate letters (Fig. 3). Hex is unusual in that it feeds the output of a language model (predicting the most likely future characters) to a dynamic system, altering the dynamic response of the system to make likely things easier, and make *a priori* unlikely text require additional evidence in the form of greater control effort.

Hex models the cursor as a ball rolling over a changing landscape (an example is shown in Fig. 4). As the language

Fig. 3. Hex, running on a PocketPC, controlled by tilt.

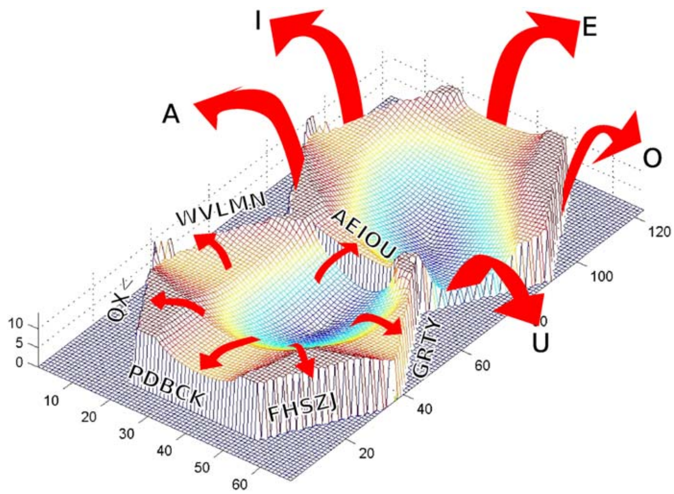

Fig. 4. The surface model for one hexagon during an interaction with Hex. A "Q" has just been entered, and the path to the vowels and then towards "U" has become much more likely. A valley has thus formed, to guide the user towards that path.

model's predictions of future characters change, the landscape forms gentle valleys along likely trajectories, and gentle inclines against unlikely ones. These soft "buffers" mitigate the cumulative effect of small deviations, and so permits higher performance maneuvers to be used reliably.

Hex is an example of a system designed to cope with interaction properties that afflict mobile interaction. It supports noisy inputs by modulating the pseudo-physics underlying the interface. Mobile interfaces are one common example where noisy, intermittently varying sensing is combined with a disparity between input and feedback channels. EEG interaction moves into even more extreme territory, where common assumptions about how interfaces should respond break down. The following sections illustrate how the basic principles of Hex were adapted to deal with the constraints of brain–computer interfaces.

4. Text entry with BCI control

4.1. Classification and its properties

The classification process is able to extract useful control signals from the extremely noisy high-dimensional data recorded from the EEG electrodes (Blankertz et al., 2007b). In the case of the Berlin brain–computer interface (BBCI), this transformation takes 128 scalp electrode readings, and maps them to a single scalar variable (Dornhege et al., 2007b; Blankertz et al., 2008a, 2007a, 2006b). The quality of this signal is, however, quite limited. In particular, noise levels are very high, and the windowing process for the classifier induces delays of a minimum of some hundred milliseconds. In practice, the output of the classifier is passed through an integrator, which smoothes out the signal, but induces further delay. However, despite this, the controllability of the interface is significantly improved by integration.

4.1.1. Timing-based interaction

One of the limitations of current EEG based control is the very limited number of states that can be reliably distinguished. Although some subjects are able to activate up to three or even more distinguishable mental states with practice, only two states are normally viable for most subjects. The classification process is not necessarily binary; it can produce continuous values, as is done in the classifier used in the Berlin BCI. However, accurate control of the level of activation of mental states is not achievable.

To transform these signals into useful symbols, the signals must be coded in some way. A simple threshold could be applied to produce binary values, and sequences of these could be used to navigate a binary tree (this is the approach used in earlier incarnations of the Berlin BCI, Dornhege, 2006). Coding strategies from BCI signals are discussed in Dornhege (2006). The alternative is to use the timing of transitions to control the interface. The bandwidth of the channel is then limited by the uncertainty in the timing of events. Timing based interfaces have been successfully applied to text entry before; the most obvious example is Morse code, which efficiently codes text using a single binary state and small set of discrete transition times. Versions of Dasher (Ward et al., 2000) have been implemented using one or two buttons to control a continuous one-dimensional cursor. The cursor moves constantly in one direction, and button pushes in the reverse direction of movement. Dasher's efficient coding algorithm partitions the space in such a way that less timing accuracy is needed for higher probability targets (Wills and MacKay, 2006; Felton et al., 2007). Dasher has been successfully implemented with EEG based control. However, its continuously paced nature has the potential to become complex when control is subject to the uncertain delays and variability that plague BCI.

Timing-based interaction can be enhanced with appropriate feedback. In order to effect the transitions accurately, users need to be able to predict when to time the next transition with sufficient warning to compensate for the delays in the control loop. If the interface can provide such cues, the quality of control can be improved. In self-clocked systems like Morse code this is unnecessary, as sequences can be completed open-loop (or rather, the timing control loop is internal to the user). However, because users generally have very limited models of how their mental activity affects the system in a BCI situation, closed-loop control is required.

4.1.2. State change transients

The dynamics of state changes are essential constraints in constructing a timing based interaction. Fig. 5 shows the typical outputs of the classifier as a user enters an imagined motor activity from a rest state. The transient has

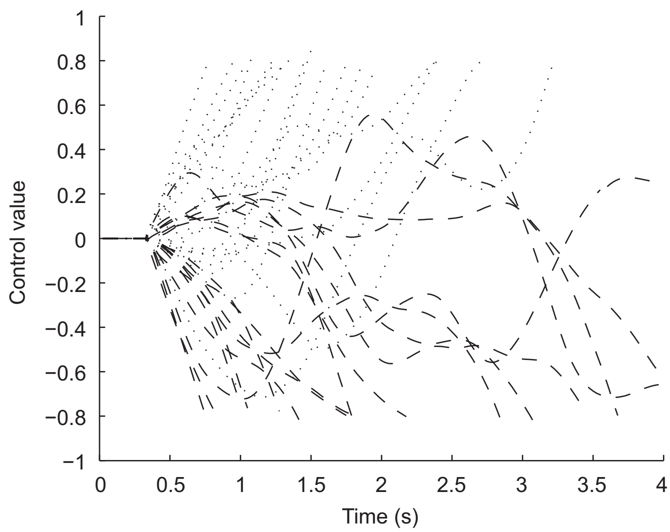

Fig. 5. Typical transients in the Berlin BCI EEG interface in a controlled target acquisition task. The trace of a one-dimensional cursor is plotted against time. The user is attempting to move an onscreen cursor either to right hand side (in the traces with dotted lines), or the left hand side (dashed lines). The y-value indicates the x-position of the cursor; the targets are at −0.8 and 0.8. This value is the filtered, clipped, integrated classifier output. The 400 ms region at the start where the trace reads 0 is where the target is shown before movement begins. The complex, lagged response of the control system is apparent.

noticeable delays which vary significantly over the course of an interaction. These classifier trajectories make control with conventional metaphors frustrating and error prone.

4.2. Criteria of quality: bit-rate, perfection and frustration

Traditionally, text entry systems have been benchmarked on input rate. An interface that allows users to input the largest number of correct symbols in the least time is considered superior. This, while an easy to measure metric, is not necessarily the most appropriate in the awkward, error-prone and slowed down world of BCI. For some users, producing perfect text may be highly desirable; others may wish to communicate rough meaning without particular regard to the precise words used. Anecdotal evidence from several BCI labs suggest that there are severely disabled users who have a strong desire to interact on an equal basis with others–rather than communicate via stock phrases and abbreviations–who strive to achieve letter-perfect text despite the process being exceptionally painstaking.

The comfort of the interface is also important. Interfaces which continuously load the user with fine control actions may be unpleasant to use because of the relentless cognitive load. Allowing intermittent interaction, or stable interaction better paced with the control the user can exert may result in interfaces which are subjectively preferable even though the character throughput is lower than in alternative interfaces.

4.3. Language modeling

Introducing prior models can dramatically reduce the bandwidth requirements for communication. In a text entry application, a language model determines the probability distribution on all symbols, given the symbols that have been produced so far ($P(c|\text{prefix})$) and any other observable evidence which might bias the selection of text (time of day, current tasks, etc.).

For standard English text, which has a very low entropy (1–3 bits per word, Brown et al., 1992) compared to the maximum entropy of the characters which make up the text, were fully independent (5–6 bits per *character*), there is a large potential gain from the introduction of a good language model. Such a language model can ease the use of text entry systems, but only if it is incorporated in an intuitive way, without wresting control from the user. Hex, and the its BCI successor, Hex-O-Spell, are examples of how this can be achieved under different control conditions.

5. Hex-O-Spell

5.1. The Berlin BCI

The Berlin brain–computer interface is an EEG-based BCI system which operates on the spatio-spectral changes during different kinds of motor imagery, i.e. the changes which occur when movements are imagined but not executed. It uses machine learning techniques to adapt to the specific brain signatures of each user, thereby achieving relatively good control with only a single training session (Blankertz et al., 2007b, 2008a).

The decoding of mental states from brain activity as used in the Berlin brain–computer interface system is described in Blankertz et al. (2008b, 2007a, b, 2006b)).

The output from the classification is a continuous control signal. Typically this is a graded classifier output which discriminates two motor imagery classes (for example, imagined movement of the left hand versus imagined movement of the right foot).

Bit rates (measured during one dimensional, synchronous cursor control target acquisition task) range between 6 and 35 bits per minute (Dornhege, 2006). The intention-to-control delay is difficult to quantify. The reaction time from stimulus presentation to significant BCI control is between 750 and 1750 ms with a large intra-subject trial-to-trial variability (compared to 300–450 ms in a two alternative forced choice task with finger movement responses to visual stimuli).

It should be noted that there is a non-negligible percentage of the population for which BCI control does not work well enough for any stable control to be achieved. Since this phenomenon is reported from all BCI laboratories it seems not to be a data analysis problem but rather be an inherent neurophysiological property. This is currently an area of intensive investigation in BCI research.

5.2. Alternative control techniques

There are BCI systems that are based on the detection of potentials that are evoked by external stimuli rather than endogenously altered mental states. Most prominent is the approach proposed by Farwell and Donchin (1988) using the P300 component. Here, characters are presented in a 6×6 matrix. The symbol on which the user focuses her/his concentration can be predicted from the brain potentials that are evoked by random flashing of rows and columns. The role of directing the gaze to the desired letter is not currently understood. Further developments (e.g. Kaper and Ritter, 2004; Krusienski et al., 2006) suggest that high spelling rates can be achieved using this approach. In the online experiments that have been reported so far, many repetitions of the stimuli have been used in order to increase the signal-to-noise ratio for P300 detection. Accordingly the spelling speed could not exceed about 6 characters per minute (cpm) even at 100% classification accuracy. However, offline analyses show that, in principle, fewer averages could be used, such that up to 15 characters per minute could be possible. This, however, still remains to be demonstrated. Although these techniques might offer slightly higher performance, their design is inherently visually noisy. It is difficult to imagine such techniques integrated into an interface suitable for a general market. Endogenously altered control offers far more potential for sleek and well-designed interactions. If BCI is to leave the lab and become widely accepted, the interaction must be designed so it is palatable to those for which it is not their only hope of communicating. This is one area where HCI could make a huge contribution to the development of BCI. Indeed, for those who depend upon BCI as their only portal into the world, it could be argued that making the interaction a pleasurable—or at least not uncomfortable—experience should be a priority. Distracting, ugly and painstaking interfaces should not be the default simply because the obvious alternative is no communication at all.

6. Implementation

6.1. Adaptation from Hex

The original Hex system was designed to deal with continuous, steering-style control. The user drove the system into desired states by generating trajectories in tilt space. Each character has a particular ideal tilt trajectory, and these can be sequenced together to form text. The amount of deviation tolerated in the trajectory is determined by the language model. When a particular sequence is likely, more of the tilt space is allocated to that sequence. This is achieved by continuously modulating the response of the system to alter the mapping of the input tilt movements and the on screen cursor movement.

In the EEG control situation, the response of the system when users attempt to control mental states is not well suited to continuous control. Users are more capable of switching between states than maintaining precisely balanced levels between them. The timing of these transitions is relatively free, however, and it is these timings which are used to control the text entry interface, known as "Hex-O-Spell" (Hex-O-Spell was originally described in Blankertz et al., 2006a). The original hexagonal tessellation is maintained, but now letters are rearranged after each transition, so that the most likely ones require the shortest transitions. Because the input is very slow, it is more effective to completely alter the display than to modulate the underlying dynamics as in Hex, where communication is less asymmetric.

In Hex, the language model was directly linked to the interface dynamics. For BCI control, that approach is difficult to apply because of the inherently slow system dynamics. The delays in control mean that adaptations to the dynamics are out of sync with user expectations. Instead, the dynamic layout adaptation algorithm is applied, resulting in more predictable behaviour.

This design decision is based upon the observed asymmetry in the BCI system. Because of the massive display bandwidth available compared to the very limited input channel, it makes sense to increase the complexity of the system model and permit varying response to control inputs. These consequences can be displayed to the user in real-time, rather than having the user learn and internalize a model. A constant one-to-one mapping of particular motion or thought pattern to an action within the system is less able to optimally use the degrees-of-freedom the interactor has to effect their intentions. In the original Hex system, where input and output were better balanced, the system supported the user, but left the fundamental trajectories for text sequences constant. This allowed the basic elements to be learned so that they can be executed quickly with minimal feedback. As the asymmetry shifts towards feedback-dominated control, the complexity of the model is transferred from the user's mind to the system. This makes the user more dependent on feedback, but requires less training and more efficient use of the input available. The continuous adaptation of response dynamics in Hex requires that the user pay more attention in return for a more productive mapping of the tilt input to the selection space.

6.2. Interaction design

The text entry system is controlled by the two mental states: *imagined right hand movement* and *imagined right foot movement*. The initial configuration is shown in the leftmost plot of Fig. 6. Six hexagonal fields surround a circle. In each of them five letters or other symbols (including "<" for backspace) are arranged. For the selection of a symbol there is an arrow in the centre of the circle. By imagining a right hand movement the arrow turns clockwise. An imagined foot movement stops the rotation and the arrow starts extending. If this foot imagination persists, the arrow touches the hexagon and

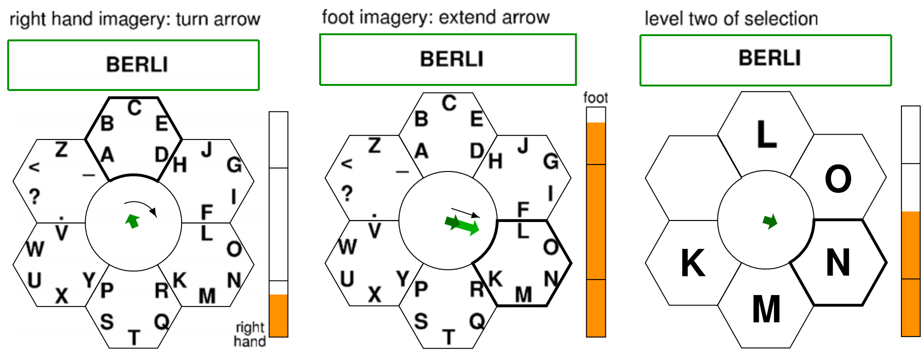

Fig. 6. The mental text entry system "Hex-o-Spell'. The two states classified by the BBCI system (bar on the right in each screenshot) control the turning and growing of the grey arrow, respectively (see also text). Letters can thus be chosen in a two step procedure. If the classifier output is undecided (orange bar between the thresholds), the arrow maintains its direction and its length diminishes continuously to minimum. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

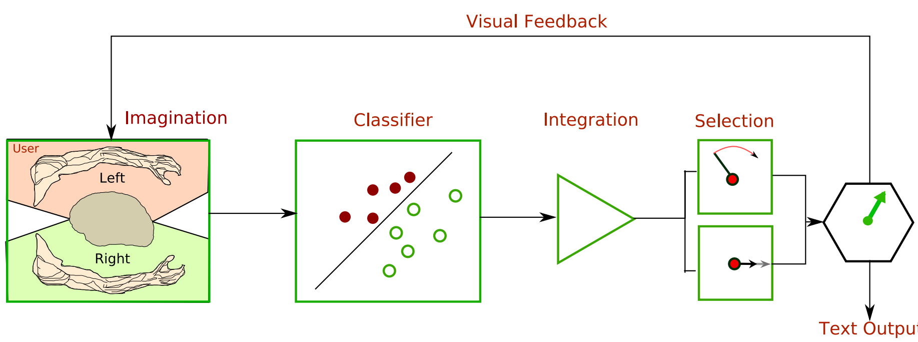

Fig. 7. The structure of the control loop in Hex-O-Spell, indicating the transformation of a discrete user intention into a continuous variable which is fed back to the user, while simultaneously generating discrete symbols.

thereby selects it. Then all other hexagons are cleared and the five symbols of the selected hexagon are moved to individual hexagons as shown in the rightmost screenshot of Fig. 6. The arrow is reset to its minimal length while maintaining its original direction. Now the same procedure (rotation if desired and extension of the arrow) is repeated to select one symbol. Note that there are only five symbols for choice in the second step, cf. rightmost screenshot of Fig. 6. Choosing the empty hexagon makes the application return to the first step without selection. This transition allows a sort of limited undo. Misspelt characters can be erased by selecting the backspace symbol "<".

6.3. The design of Hex-o-Spell

Hex-O-Spell is unusual in that the user applies binary control to produce discrete output, but does so through a continuous control process. Control is effected by imagining one of two distinct motor movements; but these are based upon the feedback from the interface, which has a continuously changing state. This state is the result of integrating the output of the classifier identifying the imagined movements, which is integrated and then thresholded to into a decision between rotation/forward motion with fixed speeds. Fig. 7 shows the structure of this control loop.

Hex-o-Spell is effectively a *timing*-based interface. The time at which the transition from the rotation state to the forward state occurs determines the letter which is selected. The rate of communication is bounded by how accurately the user can make these transitions, given the noise properties, delays and unfamiliarity of interaction present in an EEG interface. The time to traverse 60° should be calibrated against the reaction time of the user and the system; if the traversal time is much shorter than the reaction time, selection will become impossible. In Hex-O-Spell, the timing was roughly adjusted according to empirical measurements to optimize the entry speed. Before the session recording began, users were asked to try the rotation/extension mechanism with various speeds until they felt comfortable with the level of control offered. A more extensive tuning methodology might improve performance.

The language model, which adapts the layout, acts to minimize the time required for a selection, trading-off the time required to rotate to the appropriate position against the time required to visually scan the new layout and find the new locations of symbols. The "calmness" of this adaptation strategy means that the user is not always in a tightly coupled loop with the system; rather than being a flight-style control task, the interaction is broken into smaller chunks which the user can proceed through at their own pace.

6.4. Language model

The implemented language model is a modified partial predictive-match (PPM) model (Bell et al., 1984; Cleary et al., 1995), which comes close to the maximum possible compression for English (Teahan and Cleary, 1996). A tree of probabilities is stored, giving $P_{\text{PPM}}(X_N|X_{N-K}, \ldots, X_{N-1})$ as probability for the N-th letter, given the K previous letters. In the Hex-o-Spell implementation this PPM model (with $K = 2$) was combined with a modified PPM where the prefix is variable length, and runs from the start of the word. I.e. we used the probability $P_{\text{vPPM}}(X_N|X_1, \ldots, X_{N-1})$ of the N-th letter in a word, given all the previous letters of that word. These two probability models are combined by a relative weighting that depends on the relative position of the letter in a word. The relative weights for P_{vPPM} decrease linearly from 1 for the first letter to 0.5 for the sixth and all subsequent letters. The language models have been trained on a large corpus of German newspaper articles and some novels.

The right part of Fig. 8 illustrates the language model (here trained for the German language as described above) during the writing of the word “BERLIN”. In the very first step the language model reflects the prior distribution of letters beginning a word. The second row shows the situation when the second letter is selected. According to the language model, the letter “E” is the most probable second letter in a word starting with “B”. After selection of “B” the arrow is reset pointing to the hexagon containing the “E”, and the “E” is placed in “straight ahead” direction. In this way, mere continuation of the previous mental state—the least effort condition—leads to the selection of “E”. In the example the probability of the backspace symbol was chosen to be 0.1. In practice this value is set according to the control capability of the user.

6.5. Implementation details

Hex-O-Spell, originally developed in C, was ported to a native Matlab script. This ran on a separate machine from the classifier. Classifier output was streamed to the workstation on which the text entry system ran via UDP. An image of the set-up is shown in Fig. 9.

7. A case study–Hex-O-Spell in practice

7.1. Trial run details

On two days in the course of the CeBIT fair 2006 in Hanover, Germany, live demonstrations were given with two subjects simultaneously using the BBCI system. These demonstrations turned out to be BBCI robustness tests *par excellence*. All over the fair pavilion, noise sources of different kinds (both electric and acoustic) were potentially jeopardizing the performance. A low air humidity made the EEG electrode gel dry out and the subjects were under significant psychological pressure to perform well, for instance in front of TV cameras or in the presence of the German minister of research. The preparation of the experiments started at 9:15 a.m. and the live performance

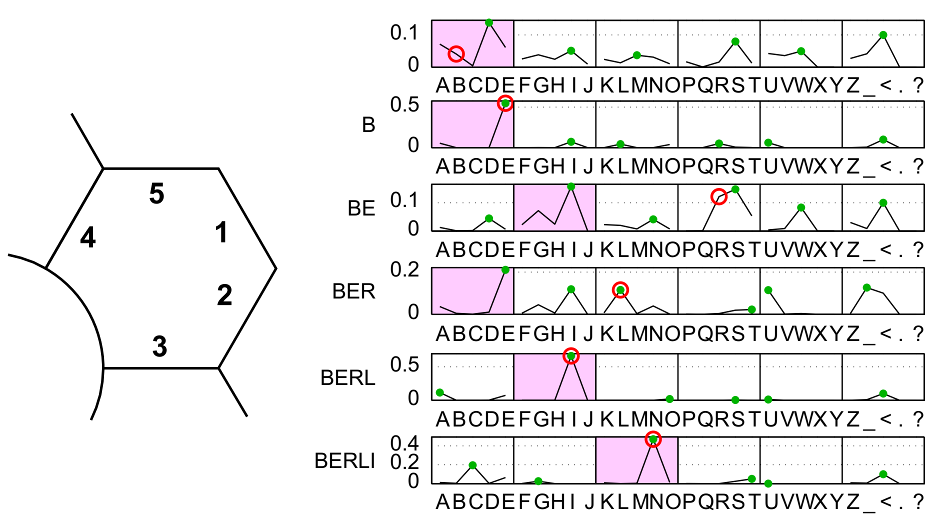

Fig. 8. Language model at work. The sketch on the left shows the ranking of the positions within one hexagon. The position that can be reached in straightforward direction from the centre is the easiest to go to (no state switch needed). Then the ranking proceeds clockwise. The figure on the right illustrates how the language model influences the operation during the spelling of “BERLIN”. Each row corresponds to the selection of one letter. Groups of five subsequent characters are placed within one hexagon, see leftmost screenshot of Fig. 6. Red shading indicates the group containing the most probable letter, i.e. these characters are in the hexagon to which the arrow initially is directed to. The ranking of probabilities within each group determine the relative position of the corresponding characters within one hexagon as indicated in the left subplot. A red circle marks the letter that actually needs to be written. (The probabilities were extracted for the German language.) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

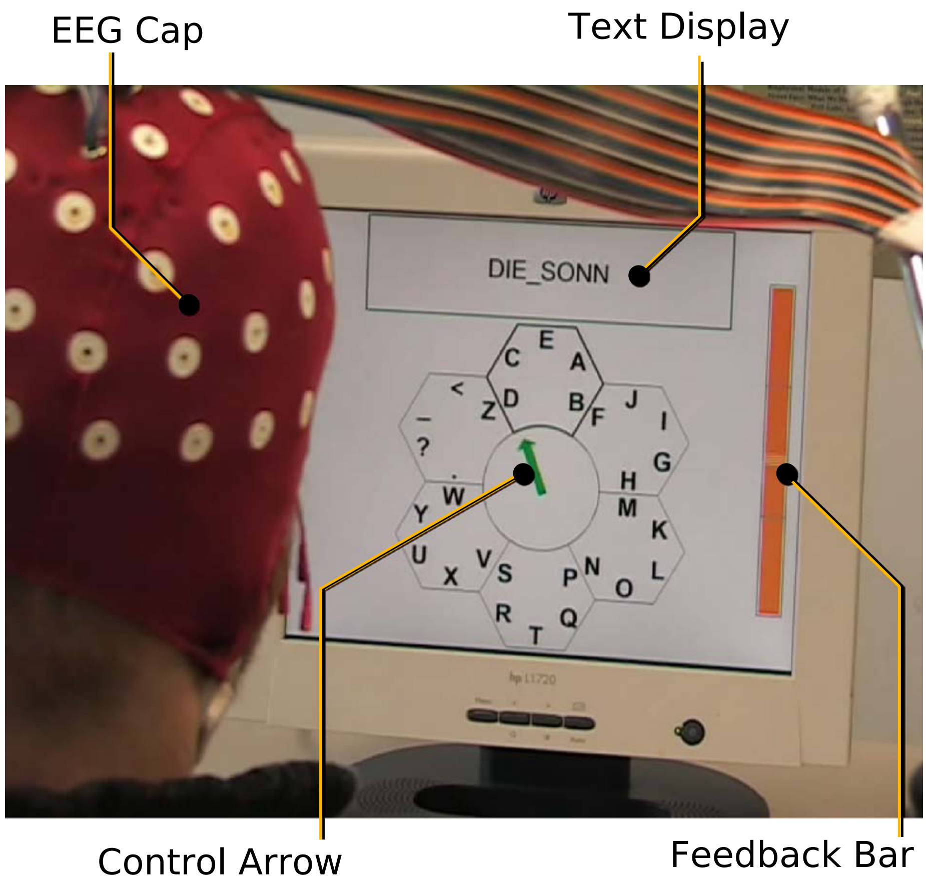

Fig. 9. The basic Hex-O-Spell set-up. The participant wears the EEG cap shown at the left. On screen, the layout of letters can be seen, along with the currently entered phrase, the state of the selection arrow, and the feedback bar which shows the filtered output of the classifier directly.

at 11 a.m. The two subjects were either playing "Brain-Pong" against each other or writing sentences with Hex-o-Spell. Except for short breaks and a longer lunch break, the subjects continued until 5 p.m. without degradation of performance over time—a demonstration of great stability. The typing speed was between 2.3 and 5 characters per minute for one subject and between 4.6 and 7 characters per minute for the other subject. This speed was measured for error-free, completed phrases, i.e. all typing errors that have been committed had to be corrected by using backspace. The total number of characters spelled in error-free phrases was over 500 per subject per day. Although this might appear slow, it is a very high speed for non-evoked EEG and is at a level where it would be of enormous utility to those with total paralysis.

7.2. *From EEG to text*

A detailed time series analysis of one particular entry sequence is given below. This shows each of the steps involved in transforming raw classifier signals into sequences of text. The time series is broken into five stages (Fig. 10):

(a) the raw classifier signal (distance from separating hyperplane);
(b) the bandpass filtered, clamped and scaled values, which are passed to the interface;
(c) the orientation and length of the pointing arrow (solid = angle, dotted = length);
(d) the transitions through edges in the hexagonal plane and
(e) the entered characters against time.

This illustrates the different processing levels that occur during the interaction. There are visual feedback mechanisms for each of these levels save the first. The moving sidebar at the right hand side of the screen shows the filtered classifier signal, the moving arrow cursor shows the pointer direction, the change of layout indicates hexagon transitions, and the text currently entered is displayed on screen.

These multiple feedback loops facilitate hierarchical control, from the base level of mental activity control to the editing of text (Fig. 11).

7.3. *Overall performance*

Results for all of the text entry sequences performed during the session are given in Table 1, and summarized in Figs. 12 and 13. Figures are in characters per minute. The average text entry speed was 4.24 cpm, with a standard

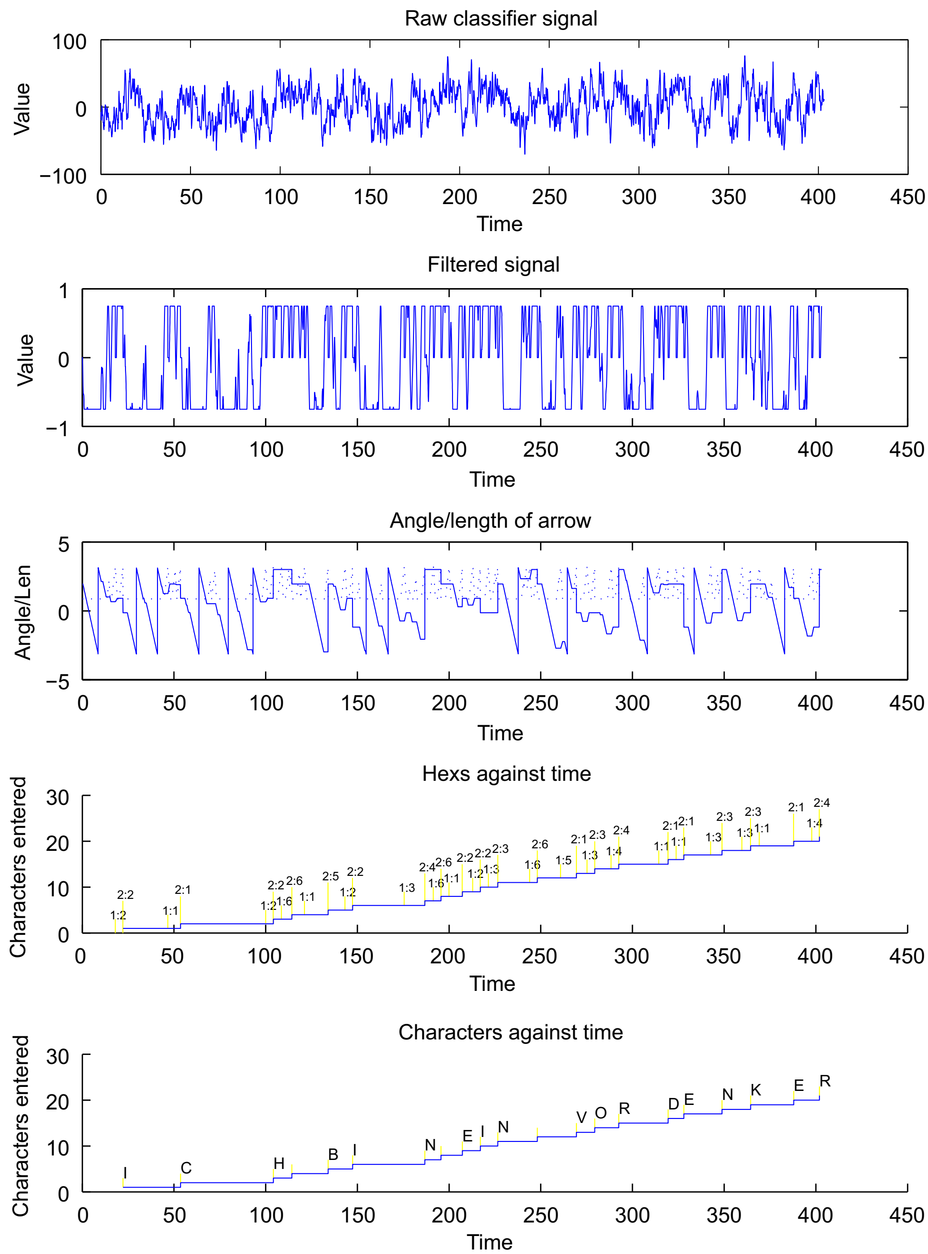

Fig. 10. From top to bottom: (a) Raw classifier output. (b) Filtered and clamped signal. Values >0 represent "extend arrow" operations; values <0 represent "rotate" operations. The alternating pattern of control becomes visible here. (c) Length and angle of the pointer arrow. The angle of rotation is shown in red; the length of the arrow is shown in green. The arrow length ranges between about 0.9 and 3.2. Changes in length and orientation take place alternately. (d) Hex transitions. Each transition is either a first step transition (1:*x*) or a second step transition (2:*x*), at which point a letter is generated. The blue staircase plot shows the number of characters entered against time. (e) The entered characters. The blue staircase shows the number of characters at each time point. The letters entered can be read above. Downward movements in the blue graph indicate a backspace. This example has been chosen for the clarity of the plots rather than the speed of entry. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

deviation of 1.638 cpm. Fig. 12 shows some further examples of character entry.

7.3.1. Discussion

For a BCI driven text entry system not operating on evoked potentials this is a very competitive spelling speed, especially taking into account the environment and the fact that the subjects did not have significant training with the use of the BBCI text entry interface: the subjects used Hex-O-Spell only twice before (although both subjects were experienced users of the Berlin BCI set-up and were familiar with the use of motor imagery control). There is noticeable variability in the time taken to enter characters. This suggests that the one-dimensional rotation control occasionally introduces significant penalties when targets are overshot and a lengthy re-rotation is required.

Hex-O-Spell is an effective speller for two-class EEG interaction. Its performance is limited by the ability of the

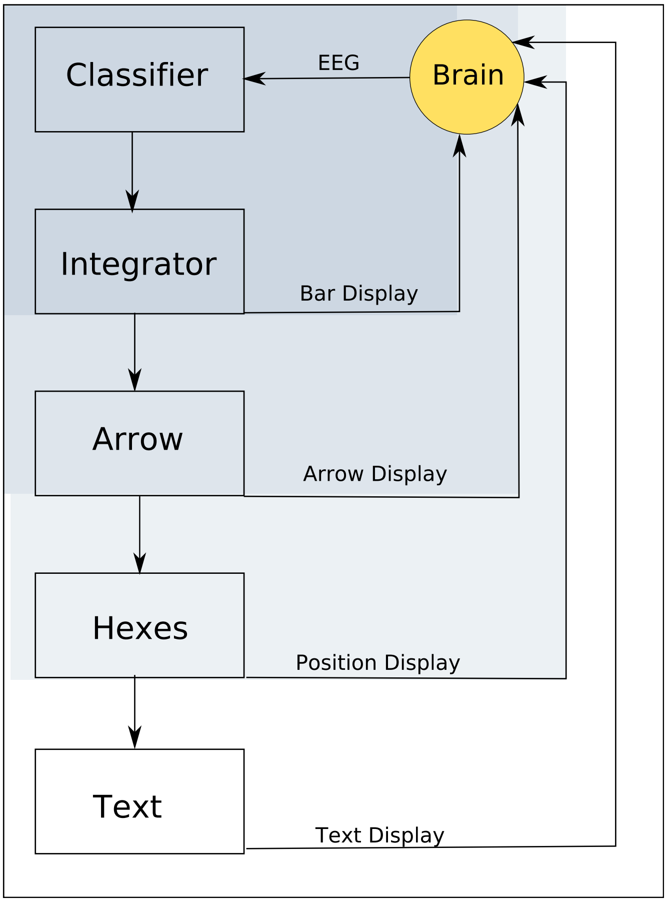

Fig. 11. Hierarchy of control loops in Hex-O-Spell. Each of the different stages of control is fed back to the user. As control moves downwards, timescales increase.

user to switch mental states with appropriate timings and the temporal accuracy with which the classifier detects these changes. There is little literature which gives detailed figures for the rate at which subjects are able to switch motor imagery, however, it seems that there is likely to be wide subject variation in timing ability. Training techniques which emphasize speed of imagery might improve performance with systems such as Hex-O-Spell, as might classifiers which are designed to rapidly respond to transients. Most training is currently focused on reliably maintaining a state; different strategies may be more effective with transient based interfaces. Other timing based structures could be designed which would reduce the worst case delays (for example, in Hex-O-Spell where overshoot occurs). Such structures could, for example, have a more explicitly rhythmic structure. Auditory feedback would be in an interesting mechanism in such cases; there is a very strong link between motor movement and rhythmic audio.

7.3.2. *Improving Hex-O-Spell*

Hex-o-Spell could also be modified to work as a T9-style system, with only a single transition for each character rather than a pair. Given that PPM models can compress

Table 1
Results for the CeBIT testing session.

Text	Mean cpm	Std. dev. cpm
ICH BIN EIN VORDENKER	3.28	5.91
ICH BIN EIN VORDENKER	3.13	5.37
ICH BIN EIN VORDENKER. DIE MENTALE SCHREIBMASCHINE IN AKTION	2.85	3.30
HALLO SUESSE	1.13	1.24
ICH BIN EIN VORDENKER	3.17	4.09
ICH BIN EIN VORDENKER	2.84	6.39
THE GADGET SHOW	2.07	4.48
NEUES AUF DREISAT	1.68	1.54
ICH BIN EIN BERLINER. MENS SANA IN CAMPARI SODA	2.66	3.28
ICH DENKE ALSO BIN ICH?	2.69	2.69
IN GEDANKEN SCHON VIEL WEITER	3.18	3.70
WO EIN BEGEISTERTER STEHT IST DER GIPFEL DER WELT. E.DORFF	1.27	0.33
DER GEIST IST KEIN ELEFANT	4.43	5.93
DIE GEDANKEN SIND FREI	3.63	8.35
ICH BIN EIN FRAUNHOFER VORDENKER	2.30	2.77
MIT UNS KOENNEN SIE RECHNEN	2.83	4.16
DIE SONNE IST VON KUPFER	3.33	5.29
GROSSHIRNRINDE AM STEUER. DURCH GEDANKENKRAFT. DAS DENKEN VERSTEHEN	6.30	11.41
DIE GEDANKEN SIND FREI. ICH DENKE ALSO SCHREIBE ICH	4.54	11.26
DER WUNSCH IST DER VATER DES GEDANKEN	7.05	9.64
DER MENSCH DENKT DAS HIRN LENKT. DAS PFERD FRISST KEINEN GURKENSALAT	5.42	9.74
BERLIN BRAIN COMPUTER INTERFACE. BERNSTEIN ZENTRUM BERLIN. ICH BIN EIN CURSOR. HOLT MICH HIER RAUS	6.43	15.78
BERLIN BRAIN COMPUTER INTERFACE. CAN YOU IMAGINE? MY BRAIN HURTS	4.46	10.83
BERLIN BRAIN COMPUTER INTERFACE. DAS DENKEN VERSTEHEN. PAUSE?	5.85	9.93
SPITZENFORSCHUNG GEFOERDERT VOM BUNDESMINISTERIUM	6.22	13.65
MENTALE SCHREIBMASCHINE IN AKTION. BERLIN BRAIN COMPUTER INTERFACE	5.05	11.23
BERLIN BRAIN COMPUTER INTERFACE. DAS DENKEN VERSTEHEN	6.44	17.24
STEUERUNG DURCH GEDANKENKRAFT. DIE GEDANKEN SIND FREI	5.35	10.79
UND MIT GEISTESSTAERKE TU ICH WUNDER AUCH	5.46	8.29
MENTALE SCHREIBMASCHINE ENTWICKELT VOM BBCI TEAM	5.88	13.77
DAS PFERD FRISST KEINEN GURKENSALAT. BERLIN BRAIN COMPUTER INTERFACE	5.63	14.45
BMBF MENSCH TECHNIK INTERAKTION. FRAUNHOFER FIRST UND CHARITE BERLIN BBCI	5.38	12.02
DEM DENKEN ZUSEHEN. WORUEBER MAN NICHT SPRECHEN KANN DARUEBER SOLL MAN SCHWEIGEN	5.28	11.30
DER KOPF IST RUND DAMIT DAS DENKEN DIE RICHTUNG WECHSELN KANN	6.68	15.43
ICH BIN MUEDE. FEIERABEND?BERLIN BRAIN COMPUTER INTERFACE	5.42	11.85

Speeds are given in characters per minute, along with the standard deviation of the character rate.

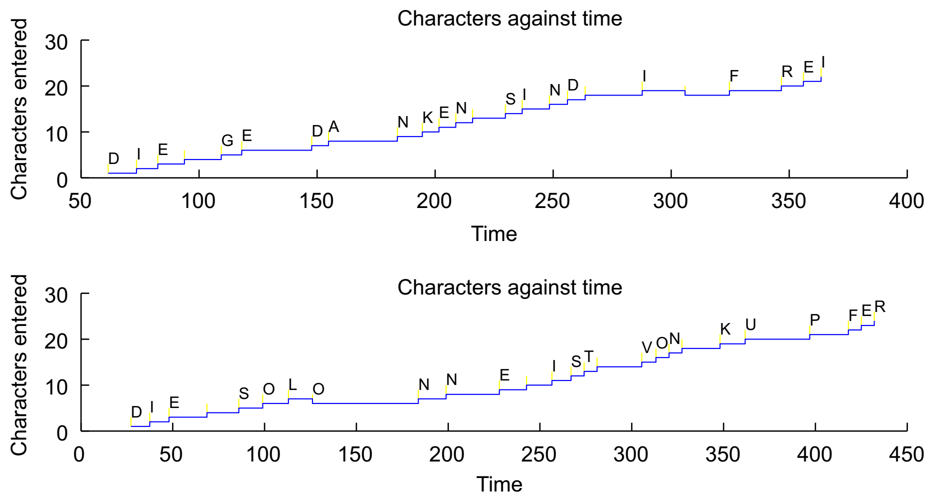

Fig. 12. Entered characters for another two trials. Average character rate of 4.57 and 3.70 cpm, respectively. Correction steps can be seen as downward movements in the staircase plot.

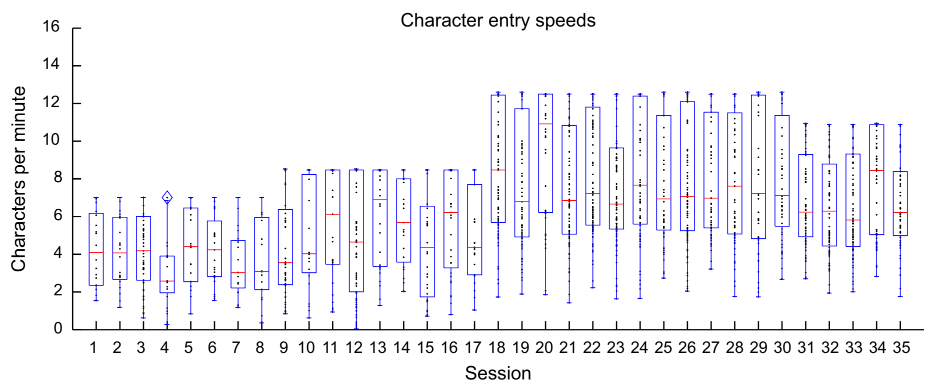

Fig. 13. Box plot of overall text entry speeds. Values are characters per minute (60.0/character time in seconds). Each single point represents one character. The upper limit reaches a saturation point because the time taken to perform selection cannot be less than the minimum time to extend the arrow. Subject 1 performed sessions 1–17 and subject 2 performed sessions 18–35. The rotation rate for subject 2 was higher than for subject 1, resulting in higher rates. Rotation rates were adjusted until the user was comfortable with the level of control achieved.

English to approximately 2 bits per character and choosing one from six transitions gives ~ 2.585 bits, this should be feasible. Dunlop (2004) describes a functioning entry system using only four transitions followed by a decoding step. Despite the potential increase of typing speed, it has to be explored whether BCI users are interested in this form of predictive text entry. There are anecdotal reports of patients who preferred a slower spelling system than using a system which suggested word completions based on a probabilistic model. The assistance of the interface was perceived to be intrusive; the patients desired communicate on the same terms as those around them, painstakingly working over phrases to make them perfect, even though rough meaning could be communicated far more quickly.

Some layout optimization could also be performed, for example, placing backspace in a easier-to-reach position. The original Hex paper Williamson and Murray-Smith (2005) described how layouts can be numerically optimized to penalize certain behaviours. In Hex, jerk was minimized; in Hex-O-Spell the inter-transient time to could be regularized so that changes of mental state could be made more predictable.

Hex-O-Spell could also potentially benefit from directly linking the interface dynamics to the language model. The rotation speed of the arrow could be dynamically adjusted to as to move less quickly when passing more likely options. This introduces a level of control similar to that in the original Hex system.

8. Conclusions

Interfaces of the future will involve a vast array of sensing and display technologies. The existing metaphors for interaction cannot be simply transplanted to these new contexts. In order to take full advantage of the potential of the available hardware, interface designers need to consider how the properties of the channels available affect control, how control can effect communication and how communication can drive action. Low-level signal properties of the control *loop* can be used as starting points for designing interfaces; such properties include delays, transient behaviour and bandwidth asymmetry. The BCI example is a particularly extreme case, where unusually long delays, high levels of uncertainty, and massive display–control asymmetry are present. Hex-O-Spell demonstrates how effective interfaces for exotic sensing hardware can be constructed by examining these properties. The text entry speeds (4–7 characters per minute) are very competitive for non-invasive BCI, and are at a level that are viable for use for those without other means of communication. Hex-O-Spell also demonstrates how interaction design can benefit those developing BCIs; specifically how designing an interface whose dynamics are sympathetic to those characteristic of a BCI can result in effective interfaces, and how integration of predictive models can be used to reduce bandwidth requirements in a natural way. For interaction designers, it illustrates some of the challenges in designing interaction where the familiar properties of devices like mice and keyboards are no longer valid. Those designing for mobile devices, where control properties are in constant flux as users activities change, should take into account the varying imbalances in the control loop and design their metaphors to cope fluidly with transitions between control modes. Interfaces designed to work with users with special needs which preclude the use of conventional control also face the challenge of designing for asymmetric control. The methods outlined in this paper suggest that powerful interface can be designed by identifying key characteristics of the control properties, and then working from these fundamentals to develop metaphors for interaction.

Acknowledgements

The authors acknowledge the great work of Guido Dornhege (former member of the IDA group at Fraunhofer FIRST) in the development of the BBCI and Hex-o-Spell.

References

Barrett, R.C., Selker, E.J., Rutledge, J.D., Olyha, R. S., 1995. Negative inertia: a dynamic pointing function. In: CHI '95: Conference Companion on Human Factors in Computing Systems, pp. 316–317.

Bell, T., Cleary, J., Witten, I., 1984. Data compression using adaptive coding and partial string matching. IEEE Transactions on Communications 32 (4), 396–402.

Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., Flor, H., 1999. A spelling device for the paralysed. Nature 398, 297–298.

Blanch, R., 2005. Architecture logicielle et outils pour les interfaces hommes–machines graphiques avancées (Software architecture and tools for advanced computer–human graphic interaction). Ph.D. Thesis, Université Paris XI, Orsay.

Blanch, R., Guiard, Y., Beaudouin-Lafon, M., 2004. Semantic pointing: improving target acquisition with control–display ratio adaptation. In: CHI '04: Proceedings of the SIGCHI Conference on Human factors in Computing Systems. ACM Press, New York, NY, USA, pp. 519–526.

Blankertz, B., Dornhege, G., Krauledat, M., Kunzmann, V., Losch, F., Curio, G., Müller, K.R., 2007a. The Berlin brain–computer interface: machine-learning based detection of user specific brain states. In: Towards Brain–computer Interfacing. MIT press, Cambridge, MA.

Blankertz, B., Dornhege, G., Krauledat, M., Müller, K.-R., Curio, G., 2007b. The non-invasive Berlin brain–computer interface: fast acquisition of effective performance in untrained subjects. Neuroimage 37 (2), 539–550 URL ⟨http://dx.doi.org/10.1016/j.neuroimage.2007.01.051⟩.

Blankertz, B., Dornhege, G., Krauledat, M., Schroder, M., Williamson, J., Murray-Smith, R., Muller, K.-R., 2006a. The Berlin brain–computer interface presents the novel mental typewriter Hex-o-Spell. In: 3rd International BCI Workshop and Training Course.

Blankertz, B., Dornhege, G., Lemm, S., Krauledat, M., Curio, G., Mueller, K.R., 2006b. The berlin brain–computer interface: machine learning based detection of user specific brain states. Journal of Universal Computer Science 12 (6), 581–607.

Blankertz, B., Losch, F., Krauledat, M., Dornhege, G., Curio, G., Müller, K.-R., 2008a. The Berlin Brain–computer interface: accurate performance from first-session in BCI-naive subjects. IEEE Transactions on Bio-Medical Engineering 55 (10), 2452–2462 URL ⟨http://dx.doi.org/10.1109/TBME.2008.923152⟩.

Blankertz, B., Tangermann, M., Popescu, F., Krauledat, M., Fazli, S., Dónaczy, M., Curio, G., Müller, K.-R., 2008b. The Berlin brain–computer interface. In: Zurada, J.M., Yen, G.G., Wang, J. (Eds.), WCCI 2008 Plenary/Invited Lectures. Lecture Notes in Computer Science, vol. 5050. Springer, Berlin, Heidelberg, pp. 79–101. URL ⟨http://dx.doi.org/10.1007/978-3-540-68860-0_4⟩.

Brown, P.F., Pietra, V.J.D., Mercer, R.L., Pietra, S.A.D., Lai, J.C., 1992. An estimate of an upper bound for the entropy of English. Computational Linguistics 18 (1), 31–40.

Carmena, J.M., Lebedev, M.A., Crist, R.E., ODoherty, J.E., Santucci, D.M., Dimitrov, D.F., Patil, P.G., Henriquez, C.S., Nicolelis, M.A., 2003. Learning to control a brain–machine interface for reaching and grasping by primates. Public Library of Science Biology E42.

Cleary, J., Teahan, W., Witten, I., 1995. Unbounded length contexts for PPM. In: DCC-95. IEEE Computer Soc. Press, Silver Spring, MD, pp. 52–61.

Dornhege, G., 2006. Increasing information transfer rates for brain–computer interfacing. Ph.D. Thesis, University of Potsdam.

Dornhege, G., del R. Millán, J., Hinterberger, T., McFarland, D., Mueller, K.R., 2007a. Towards Brain–computer Interfacing. MIT Press, Cambridge, MA.

Dornhege, G., Krauledat, M., Müller, K.-R., Blankertz, B., 2007b. General signal processing and machine learning tools for BCI. In: Dornhege, G., del R. Millán, J., Hinterberger, T., McFarland, D., Müller, K.-R. (Eds.), Toward Brain–computer Interfacing. MIT Press, Cambridge, MA, pp. 207–233.

Dunlop, M.D., 2004. Watch-top text-entry: can phone-style predictive text-entry work with only 5 buttons? In: Mobile HCI 2004. Lecture Notes in Computer Science, vol. 3160. Springer, Berlin, pp. 342–346.

Elbert, T., Rockstroh, B., Lutzenberger, W., Birbaumer, N., 1980. Biofeedback of slow cortical potentials. i. Electroencephalography and Clinical Neurophysiology 48, 293–301.

Farwell, L., Donchin, E., 1988. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography and Clinical Neurophysiology 70, 510–523.

Felton, E., Lewis, N., Wills, S., Radwin, R., Williams, J., 2007. Neural signal based control of the Dasher writing system. In: Proceedings of the 3rd International IEEE EMBS Conference on Neural Engineering. Capri 2007, pp. 366–370.

Hochberg, L., Serruya, M., Friehs, G., Mukand, J., Saleh, M., Caplan, A., Branner, A., Chen, D., Penn, R., Donoghue, J., 2006. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature 442 (7099), 164–171.

Isokoski, P., Raisamo, R., 2004. Speed and accuracy of six mice. Asian Information-Science-Life 2 (2), 131–140.

Kaper, M., Ritter, H., 2004. Generalizing to new subjects in brain–computer interfacing. In: Proceedings of the 26th Annual International Conference IEEE EMBS, pp. 4363–4366.

Krusienski, D.J., Sellers, E.W., Cabestaing, F., Bayoudh, S., McFarland, D.J., Vaughan, T.M., Wolpaw, J.R., 2006. A comparison of classification techniques for the p300 speller. Journal of Neural Engineering 3 (4), 299–305.

Kuebler, A., Kotchoubey, B., Kaiser, J., Wolpaw, J., Birbaumer, N., 2001. Brain–computer communication: unlocking the locked in. Psychological Bulletin 127 (3), 358–375.

Millan, J., 2008. Brain-controlled robots. IEEE Intelligent Systems 23, 74–76.

Mueller-Putz, G.R., Pfurtscheller, G., 2008. Control of an electrical prosthesis with an SSVEP-based BCI. IEEE Transactions on Biomedical Engineering 55 (1), 361–364.

Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P.N., Müller, K.-R., 2006. Towards adaptive classification for BCI. Journal of Neural Engineering 3 (1), R13–R23 URL ⟨http://dx.doi.org/10.1088/1741-2560/3/1/R02⟩.

Teahan, W.J., Cleary, J.G., 1996. The entropy of English using PPM-based models. In: Data Compression Conference, pp. 53–62.

Ward, D.J., Blackwell, A.F., MacKay, D.J.C., 2000. Dasher—a data entry interface using continuous gestures and language models. In: UIST 2000. ACM, New York, pp. 29–137.

Williamson, J., 2006. Continuous uncertain interaction. Ph.D. Thesis, University of Glasgow.

Williamson, J., Murray-Smith, R., 2005. Dynamics and probabilistic text entry. In: Proceedings of the Hamilton Summer School on Switching and Learning in Feedback systems. Lecture Notes in Computer Science, vol. 3355. Springer, Berlin, pp. 333–342.

Wills, S., MacKay, D., 2006. DASHER–an efficient writing system for brain–computer interfaces? IEEE Transactions on Neural Systems and Rehabilitation Engineering 14, 244–246.

Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M., 2002. Brain–computer interfaces for communication and control. Clinical Neurophysiology 113 (6), 767–791.