



UNIVERSITY
of
GLASGOW

Blankertz, B. and Krauledat, M. and Dornhege, G. and Williamson, J. and Murray-Smith, R. and Müller, K-R (2007) A note on brain actuated spelling with the Berlin brain-computer interface. *Lecture Notes in Computer Science 4557/2007*:pp. 759-768.

<http://eprints.gla.ac.uk/3673/>

A Note on Brain Actuated Spelling with the Berlin Brain-Computer Interface

B. Blankertz¹, M. Krauledat^{1,2}, G. Dornhege¹, John Williamson³, Roderick Murray-Smith^{3,4}, and K.-R. Müller^{1,2} *

¹ Fraunhofer FIRST (IDA), Berlin, Germany

² Technical University Berlin, Berlin, Germany

³ University of Glasgow, Glasgow, Scotland

⁴ Hamilton Institute, NUI Maynooth, Ireland

Abstract. Brain-Computer Interfaces (BCIs) are systems capable of decoding neural activity in real time, thereby allowing a computer application to be directly controlled by the brain. Since the characteristics of such direct brain-to-computer interaction are limited in several aspects, one major challenge in BCI research is intelligent front-end design. Here we present the mental text entry application ‘Hex-o-Spell’ which incorporates principles of Human-Computer Interaction research into BCI feedback design. The system utilises the high visual display bandwidth to help compensate for the extremely limited control bandwidth which operates with only two mental states, where the timing of the state changes encodes most of the information. The display is visually appealing, and control is robust. The effectiveness and robustness of the interface was demonstrated at the CeBIT 2006 (world’s largest IT fair) where two subjects operated the mental text entry system at a speed of up to 7.6 char/min.

1 Introduction

Brain-computer interfaces (BCIs) translate brain signals into control commands. The measured brain signals reflect, to some extent, the intentions of a subject. The control commands may be used for a computer application or 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 ([3,4]). Furthermore it was demonstrated that noninvasive BCI systems enable healthy subjects as well as patients to control an internet browser or simple word processing software ([5,6,7]).

Since a principle motivation of the development of BCIs is to provide paralyzed patients with independent communication tools, BCI-driven spelling devices are an important topic in BCI research. The Tübingen BCI group developed a system that could be operated by patients suffering from amyotrophic lateral sclerosis ([8]). Binary decisions of the BCI were used to select letters in a procedure where the alphabet was

* The studies were partly supported by the *Bundesministerium für Bildung und Forschung* (BMBF), FKZ 01BE01A, by the SFI (00/PL.1/C067), 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. This paper is builds upon [1,2], but also presents new material.

iteratively split into halves. The achieved spelling rate was about 0.5 char/min. With a similar front-end but a different BCI approach, a spelling application of the Graz group could be operated by one patient suffering from severe cerebral palsy at about 1 char/min ([9]). In [10] a spelling application is proposed that is based on a three-class BCI. While one class can be used to scroll through the alphabet which is presented on two ‘assembly lines’ left and right of the cursor, the other two classes are used to select the character from either the left or the right line. Since scrolling is uni-directional, missing the desired character necessitates scrolling through the whole alphabet for another chance. Two out of three BCI-trained users are able to operate the device at spelling speed 2.35 resp. 1.62 char/min (average for 5 words).

Furthermore there are BCI spelling devices that are based on the detection of potentials that are evoked by external stimuli rather than endogeneously altered mental states. Most prominent is the approach proposed by Donchin et al. ([11]) using the P300 component. Here all 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 so far not investigated. Further developments (e.g., [12,13]) 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 char/min even at 100% classification accuracy. Nevertheless offline analyses show that in principle fewer averages could be used, such that up to 15 char/min could be possible. Still, this has to be shown in practice.

Although the proof-of-concept of BCI systems was given decades ago (e.g. [14]), several major challenges are still to be faced. One of those challenges is to develop BCI 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, its bandwidth 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, cf. Fig. 1 (a).

The Berlin Brain-Computer Interface (BBCI) is an EEG-based BCI system which operates on the spatio-spectral changes during different kinds of motor imagery. It uses machine learning techniques to adapt to the specific brain signatures of each user, thereby achieving high quality feedback already in the first session ([15]). The mental text entry system *Hex-o-Spell* which is presented here adapts modern, dynamic text entry methods into a suitable form for brain-computer control.

The idea for Hex-o-Spell was taken from the Hex system ([16,17], see Fig. 1 (b)), which was designed for use on mobile devices augmented with accelerometers, where tilt control was used to maneuver through a hexagonal tessellation. The system adapted the response dynamics in order to make control behaviour for likely actions easier than unlikely ones, without altering the ideal path for any given word trajectory. This was intended to maintain a level of stability in the patterns required to generate letter sequences, such that the user could "bootstrap" from closed-loop to open-loop control. The adjustment was based upon a continuously-updating language model which inferred the next character given the previous text sequence. Hex-o-spell was adapted

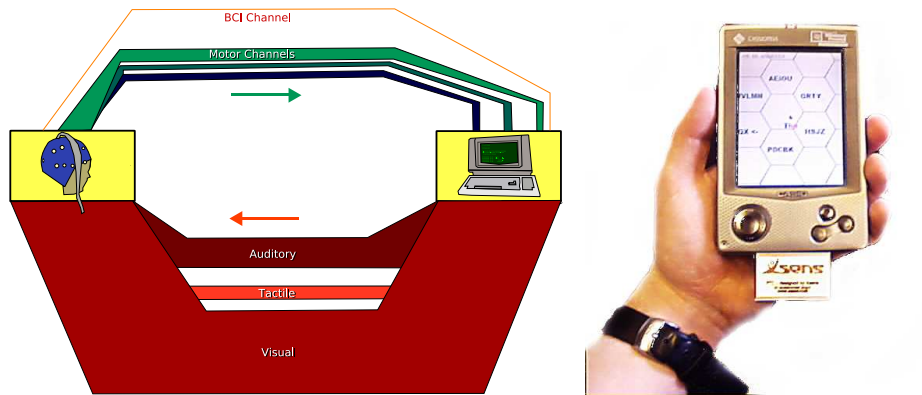


Fig. 1. (a) Left: Asymmetry of BCI communication. (b) Right: Text entry system *Hex* for mobile devices.

from this original system, replacing the two-dimensional tilt control with a rotation/forward switching input (see Section 2.1) and introducing layout rearrangement in place of the adaptive dynamics (see Section 2.2). In BCIs, where there is enormous asymmetry in the bandwidth of the channels in the control loops (see Fig. 1 (a)), high-quality language models are essential to extract every drop of salient information from the user, using the large display bandwidth to make the user aware of the effect of their actions in combination with the language model. Although these introduce continual changes which may be difficult to predict (from the point of view of the user), the consequent reliance on continuous feedback may be tolerable given the extremely limited bit rates. Users have to move so slowly that they have plenty of time to search the space for changes.

2 Methods

The decoding of mental states from brain activity as used in the Berlin Brain-Computer Interface system is described in another contribution in this volume, see [18], and in earlier publications ([15,19]). In short, the BBCI detects the user-specific spatio-spectral changes of the EEG during motor imagination of, e.g., the left or the right hand or the feet. Applications are controlled by a continuous control signal. Typically this is the graded classifier output which discriminates two motor imagery classes. It has been demonstrated that the machine learning approach which is realized in the BBCI allows to achieve high quality feedback already in the very first session without subject training ([15,19,20]). Bit rates (measured during one dimensional cursor control) range between 6 and 40 bits per minute. 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 to 450 ms in a 2 alternative forced choice task with finger movement responses to visual stimuli).

Note that there is a non-negligible percentage of the population for which BCI control does not work well enough to control applications. Since this phenomenon is re-

ported from all BCI laboratories it seems not to be a problem of data analysis but rather inherent in the neurophysiological properties of the scalp EEG in some subjects. An investigation of this issue will require a large experimental approach which is definitely one of the burning issues in BCI research.

2.1 Character Selection Procedure in Hex-o-Spell

The challenge in designing a mental text entry system is to map a small number of BCI control states (typically two) to the high number of symbols (26 letters plus punctuation marks) while accounting for the low signal to noise ratio in the control signal. The more fluid interaction in the BBCI system was made possible by introducing an approach which combined probabilistic data and dynamic systems theory based on our earlier work ([16]) on mobile interfaces.

Here we take the example that 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. 2. 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 center 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 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. 2. 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 5 symbols for choice in the second step, cf. rightmost screenshot of Fig. 2. Choosing the empty hexagon makes the application return to the first step without selection. This

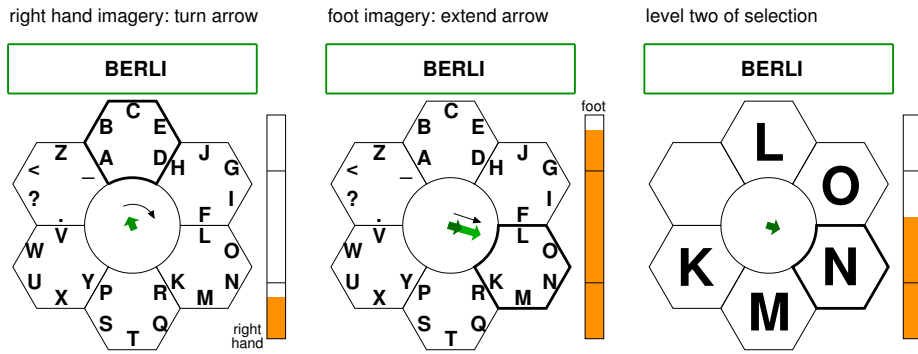


Fig. 2. The mental text entry system ‘Hex-o-Spell’. The two states classified by the BBCI system (bar on the right in each screenshot) control the turning and growing of the grey arrow respectively (see also text). Letters can thus be chosen in a two step procedure. If the classifier output is undecided (orange bar between the thresholds), the arrow maintains its direction and its length diminishes continuously to minimum.

transition allows a sort of limited undo. Misspelt characters can be erased by selecting the backspace symbol ‘<’.

There are several parameters that can be adapted to the specific capacities of the user, like the turning and the growing speed of the arrow.

2.2 The Language Model

A language model can ease the use of text entry systems effectively if it is incorporated in an intuitive way. The language model determines a probability distribution on all symbols, given the letters that have been written so far. Such probability distribution can be used to construct (and dynamically adapt) the decision tree, such that more probable symbols and be reached faster.

The implemented language model is a modified partial predictive-match (PPM) model ([21,22]), which comes close to the maximum possible compression for English ([23]). A tree of probabilities is stored, giving $P_{\text{PPM}}(X_N | X_{N-K}, \dots, X_{N-1})$ 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 $P_{\text{VPPM}}(X_N | X_1, \dots, X_{N-1})$ of the N -th letter in a word, given all 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 6th and all subsequent letters. The language models have been trained on a large corpus of German newspaper articles and some novels.

From an information-theoretic point of view the whole arrangement of symbols to the positions in the six hexagons should be controlled by the language model ([17] describes how this optimisation was carried out for the original Hex system). We decided not to do so, since such policy would reduce the potential for learning and require significant search times. S/he would need to search the whole screen before each selection to find the desired symbol. Instead we keep the grouping of the symbols to the six hexagons fixed (e.g. letter ‘A’ to ‘E’ are always in the topmost hexagon, see Fig. 2). Only the arrangement of the symbols within one hexagon is controlled by the language model. In this arrangement the probability of a letter according to the language model is matched with the rank of the position which reflect how easy that position can be reached. The symbol that is in ‘straight-forward’ direction can be reached most easily, since in this case the user just needs to go on with the ‘go straight’ command. For the position clockwise next to it, switching to the turn state is required for a 60° turn, and so on, cf. Fig. 3. On the other hand the language model dictates that after one symbol has been selected the arrow will point to that hexagon which contains the most probable next letter.

The right part of Fig. 3 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 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’. This has to following two consequences. After selection of ‘B’ (using the ‘go

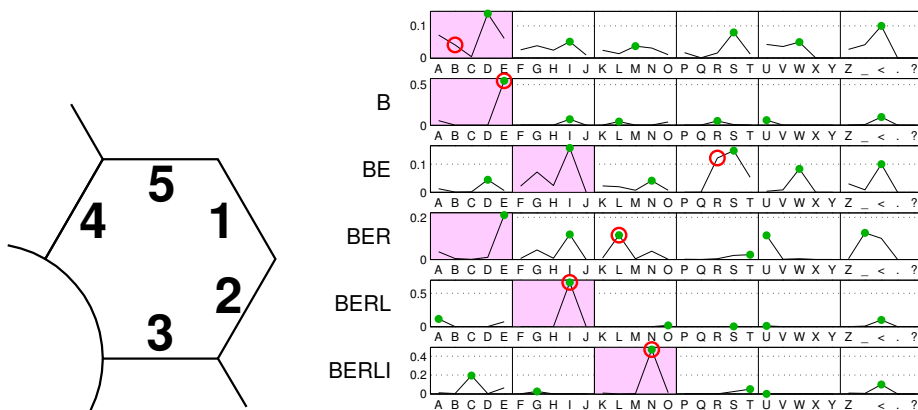


Fig. 3. Language model at work: The sketch on left shows the ranking of the positions within one hexagon. The position that can be reached in straight-forward direction from the center is the easiest to go to (no state switch needed). Then the ranking proceeds clockwise. The figure on the right illustrates 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. 2. 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.)

straight’ state in the end) the arrow is reset pointing to the hexagon containing the ‘E’, and the ‘E’ is placed in ‘straight forward’ direction within the hexagon (topmost position). This way the mere continuation of the previous mental state (corresponding to the ‘go straight’ command) 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.

2.3 Comments on the Design of Hex-o-Spell

Hex-O-Spell is unusual in that it uses that the user applies binary control to produce discrete output, but does so through a continuous control process. Control is effected by imagining motor movements; but these are based upon the feedback from the interface, which has a continuously changing state. This state is the result of the integration 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. 4 shows the structure of this control loop.

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 English to approximately 2 bits per character and choosing one from six transitions gives ~ 2.585 bits, this should be quite practical. [24] describes a functioning entry

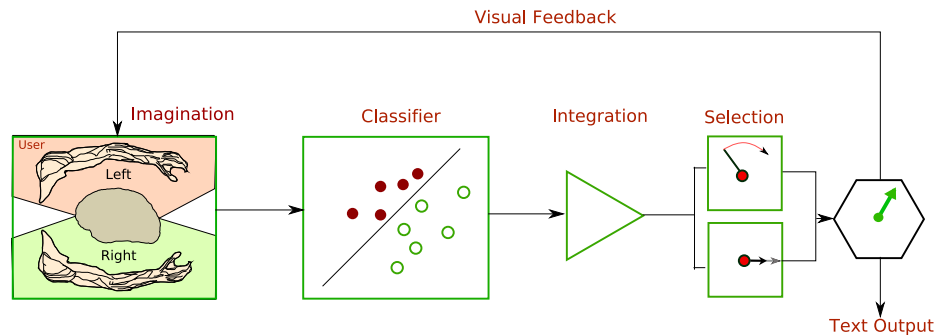


Fig. 4. The structure of the control loop in Hex-O-Spell, indicating the transformation of a discrete user intention into a continuous variable which is fed back to the user, while simultaneously generating discrete symbols.

system using only four transitions followed by a decoding step. Despite of its increment of information transfer rate, 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.

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. The language model, which adapts the layout, acts to minimize the time required for a selection, trading-off the minimization of the time required to rotate to the appropriate position for selection against the time required to 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 flying-like control task, the interaction is broken into smaller chunks which the user can proceed through at their own pace. This is one advantage over systems such as Dasher [25], which although extremely efficient control is possible with continuous, relatively noise-free input devices such as mice or eye trackers, but is less suited to the discrete, pulse-like control present in a BCI.

3 Results

On two days in the course of the CeBIT fair 2006 in Hannover, Germany, live demonstrations were given with two subjects simultaneously using the BCI system. These demonstrations turned out to be BCI robustness tests *par excellence*. All over the fair pavilion, noise sources of different kinds (electric, acoustic,...) were potentially jeopardizing the performance. A low air humidity made the EEG electrode gel dry out and

last, but not least, the subjects were under psychological pressure to perform well, for instance in front of several running 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 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 which is a demonstration of great stability. The typing speed was between 2.3 and 5 char/min for one subject and between 4.6 and 7.6 char/min 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 up to 560 per subject per day.

For a BCI driven text entry system not operating on evoked potentials this is a world class spelling speed, especially taking into account the environment and the fact that the subjects did not train the usage of the BCCI text entry interface: the subjects used Hex-O-Spell only twice before.

4 Discussion

Compared to spelling applications that have so far been coupled to BCIs ([10,26,8]) Hex-o-Spell is by far most sophisticated in terms of HCI principles. Conjoint with the powerful BCCI this mental text entry system allowed to achieve world class spelling performance. Hopefully this demonstration initiates the advancement of BCI applications from its rather simple state to more intelligent designed front-ends.

One of the aims of Hex-o-Spell is to make the best use of the language model to reduce the effort required to enter text, without inducing enormous cognitive load or extensive training time. There are four common approaches to introducing language models into text entry systems: post hoc interpretation (e.g. as used in T9); adaptive target resizing (as in Dasher [25]); dynamics adjustment (as in the original Hex); and layout re-ordering (used in Hex-o-Spell). Target resizing is simple to understand, but the visual display fluctuates rapidly and significant space is required to display the re-sized alternatives. The reinterpretation approach allows for more powerful language modeling (because subsequent letters can affect estimates of previous ones), but the lack of predictability of output can be confusing for users. Adaptive dynamics can be used to produce an efficient and visually stable display, but is less suitable for the one dimensional control inputs present in the BCI interface. The rearrangement strategy does require visual search at every new letter input, but the minimal reorganization algorithm used in Hex-o-Spell significantly reduces the impact of this. Compared to other potential entry styles, such as Dasher or grid selection mechanisms, Hex-o-Spell is also very visually compact; the hexagonal display can potentially be used as a small overlay on top of a text being edited, giving the user an overview of the context in which they are editing.

The prospective value of BCI research for rehabilitation is well known. In light of the work presented here we would advocate a further point. BCI provides stimulation to HCI researchers as an extreme example of the sort of interaction which is becoming more common: interaction with 'unconventional' computers in mobile phones, or

with devices embedded in the environment. These have a number of shared attributes: high-dimensional, noisy inputs, which describe intrinsically low-dimensional content; data with content at multiple time-scales; and a significant uncontrolled variability. The mismatch in the bandwidth between the display and control channels (as explained in the introduction) and the slow, frustrating error correction motivate a more ‘negotiated’ style of interaction, where commitments are withheld until appropriate levels of evidence have been accumulated (i.e. the entropy of the beliefs inferred from the behavior of the joint human-computer system should change smoothly, limited by the maximum input bandwidth). The dynamics of a cursor, given such noisy inputs, should be stabilized by controllers which infer potential actions, as well as the structure of the variability in the sensed data. Hex-o-Spell demonstrates the potential of such intelligent stabilising dynamics in a noisy, but richly-sensed medium. The results suggest that the approach is a fruitful one, and one which creates the potential for incorporating sophisticated models without *ad hoc* modifications.

References

1. Klaus-Robert Müller and Benjamin Blankertz, “Toward noninvasive brain-computer interfaces,” *IEEE Signal Processing Magazine*, vol. 23, no. 5, pp. 125–128, September 2006.
2. Benjamin Blankertz, Guido Dornhege, Matthias Krauledat, Michael Schröder, John Williamson, Roderick Murray-Smith, and Klaus-Robert Müller, “The Berlin Brain-Computer Interface presents the novel mental typewriter Hex-o-Spell,” in *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course 2006*. 2006, pp. 108–109, Verlag der Technischen Universität Graz.
3. L.R. Hochberg, M.D. Serruya, G.M. Friehs, J.A. Mukand, M. Saleh, A.H. Caplan, A. Branner, D. Chen, R.D. Penn, and J.P. Donoghue, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia,” *Nature*, vol. 442, no. 7099, pp. 164–171, July 2006.
4. J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O’Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. Nicolelis, “Learning to control a brain-machine interface for reaching and grasping by primates,” *Public Library of Science Biology*, vol. E42, 2003.
5. Andrea Kübler, Boris Kotchoubey, Jochen Kaiser, Jonathan Wolpaw, and Niels Birbaumer, “Brain-computer communication: Unlocking the locked in,” *Psychological Bulletin*, vol. 127, no. 3, pp. 358–375, 2001.
6. Jonathan R. Wolpaw, Niels Birbaumer, Dennis J. McFarland, Gert Pfurtscheller, and Theresa M. Vaughan, “Brain-computer interfaces for communication and control,” *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.
7. Guido Dornhege, Jose del R. Millán, Thilo Hinterberger, Dennis McFarland, and Klaus-Robert Müller, Eds., *Towards Brain-Computer Interfacing*, MIT Press, 2007, in press.
8. N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, “A spelling device for the paralysed,” *Nature*, vol. 398, pp. 297–298, 1999.
9. C. Neuper, G. Müller, A. Kübler, N. Birbaumer, and G. Pfurtscheller, “Clinical application of an eeg-based brain-computer interface: A case study in a patient with severe motor impairment,” *Clinical Neurophysiology*, vol. 114, no. 3, pp. 399–409, 2003.
10. Reinhold Scherer, Gernot R. Müller, Christa Neuper, Bernhard Graiman, and Gert Pfurtscheller, “An synchronously controlled EEG-based virtual keyboard: Improvement of the spelling rate,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 51, no. 6, pp. 979–984, 2004.

11. L.A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, pp. 510–523, 1988.
12. Matthias Kaper and Helge Ritter, "Generalizing to new subjects in brain-computer interfacing," in *Proceedings of the 26th Annual International Conference IEEE EMBS, San Francisco*, 2004, pp. 4363–4366.
13. D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayouth, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "A comparison of classification techniques for the P300 speller," *Journal of Neural Engineering*, vol. 3, no. 4, pp. 299–305, Dec 2006.
14. Thomas Elbert, Brigitte Rockstroh, Werner Lutzenberger, and Niels Birbaumer, "Biofeedback of slow cortical potentials. I," *Electroencephalography and Clinical Neurophysiology*, vol. 48, pp. 293–301, 1980.
15. Benjamin Blankertz, Guido Dornhege, Matthias Krauledat, Klaus-Robert Müller, Volker Kunzmann, Florian Losch, and Gabriel Curio, "The Berlin Brain-Computer Interface: EEG-based communication without subject training," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 147–152, 2006.
16. John Williamson and Roderick Murray-Smith, "Dynamics and probabilistic text entry," in *Proceedings of the Hamilton Summer School on Switching and Learning in Feedback systems*, R. Murray-Smith and R. Shorten, Eds., 2005, vol. 3355 of *Lecture Notes in Computing Science*, pp. 333–342.
17. John Williamson, *Continuous Uncertain Interaction*, Ph.D. thesis, Department of Computing Science, University of Glasgow, 2006.
18. Klaus-Robert Müller, Matthias Krauledat, Guido Dornhege, Gabriel Curio, and Benjamin Blankertz, "Machine learning and applications for brain-computer interfacing," in *Proceedings of HCI International 2007*, Beijing, P.R. China, 2007, submitted.
19. Benjamin Blankertz, Guido Dornhege, Steven Lemm, Matthias Krauledat, Gabriel Curio, and Klaus-Robert Müller, "The Berlin Brain-Computer Interface: Machine learning based detection of user specific brain states," *Journal of Universal Computer Science*, vol. 12, no. 6, pp. 581–607, 2006.
20. Benjamin Blankertz, Guido Dornhege, Matthias Krauledat, Volker Kunzmann, Florian Losch, Gabriel Curio, and Klaus-Robert Müller, "The berlin brain-computer interface: Machine-learning based detection of user specific brain states," in *Towards Brain-Computer Interfacing*, Guido Dornhege, Jose del R. Millán, Thilo Hinterberger, Dennis McFarland, and Klaus-Robert Müller, Eds. MIT press, 2007, in press.
21. T. Bell, J. Cleary, and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.
22. J. Cleary, W. Teahan, and I. Witten, "Unbounded length contexts for PPM," in *DCC-95*, 1995, pp. 52–61, IEEE Computer Society Press.
23. W. J. Teahan and John G. Cleary, "The entropy of English using PPM-based models," in *Data Compression Conference*, 1996, pp. 53–62.
24. M. D. Dunlop, "Watch-top text-entry: Can phone-style predictive text-entry work with only 5 buttons?," in *Mobile HCI 2004*, 2004, vol. 3160 of *Lecture Notes in Computer Science*, pp. 342–346.
25. D. J. Ward and D. J. C. MacKay, "Fast hands-free writing by gaze direction," *Nature*, vol. 418, no. 6900, pp. 838, 2002.
26. J. R. Wolpaw, D. J. McFarland, T. M. Vaughan, and G. Schalk, "The Wadsworth Center brain-computer interface (BCI) research and development program," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 207–207, 2003.