

DASHER—An Efficient Writing System for Brain–Computer Interfaces?

Sebastian A. Wills and David J. C. MacKay

Abstract—DASHER is a human–computer interface for entering text using continuous or discrete gestures. Through its use of an internal language model, DASHER efficiently converts bits received from the user into text, and has been shown to be a competitive alternative to existing text-entry methods in situations where an ordinary keyboard cannot be used. We propose that DASHER would be well-matched to the low bit-rate, noisy output obtained from brain–computer interfaces (BCIs), and discuss the issues surrounding the use of DASHER with BCI systems.

Index Terms—Assistive devices, augmentative communication, brain–computer interface (BCI), predictive language model, speller, text entry, user interfaces.

I. INTRODUCTION

DASHER is a user interface for entering text into a computer. DASHER was conceived as a means of communication in situations where use of a full-size keyboard is not possible. Such situations include the use of mobile computing devices (PDAs, mobile telephones) and communication by disabled users who are not able to use a keyboard.

DASHER's efficiency comes from its combination of a language model which predicts the probabilities of the characters that the user may write next and a user-interface which translates short, simple gestures into well-predicted text. DASHER is free, open-source software.¹

When DASHER is started, letters of the alphabet appear in boxes aligned to the right-hand edge of the screen. To write a phrase beginning with the word "hello," the user begins zooming in on the letter "h." As the box containing "h" fills more of the screen, the user is able to see sub-boxes containing possible subsequent letters (Fig. 1). Thus, the user navigates towards the "e" sub-box, then "l" within the "e," and so on. One way to conceptualize this process is by considering the library of *all possible books* arranged in alphabetical order along a single shelf (the right-hand side of the initial DASHER screen). To write, the user simply zooms in on the shelf at the position containing the book they wanted to write.

In DASHER, the size of each box within its parent box is determined by the corresponding letter's probability according to a language model. As a result, sequences of characters that are well predicted by the language model take less time to zoom into. Improbable sequences of characters are always possible to write, but take longer.

DASHER's language model is initialized using example text written in the user's language, and continuously adapts to the user's writing style.

The method by which users indicate which region of the DASHER "landscape" to zoom in on depends on the input device available, and is discussed below. In all cases, the essential principle is that the user navigates towards the location in the DASHER landscape containing the phrase they wish to write.

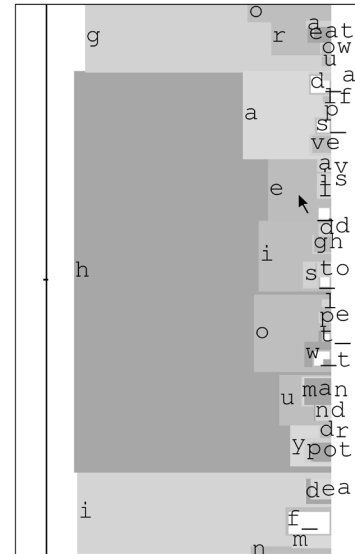


Fig. 1. Screenshot of DASHER when the user starts writing hello. Shelf of the alphabetical "library" is displayed vertically. Space character, shown by an underscore, is included in the alphabet after z. Here, the user has zoomed in on the portion of the shelf containing messages beginning with g, h, and i. Following the letter h, the language model makes the letters a, e, i, o, u, and y easier to write by giving them more space. Common words such as had and have are visible. Pointer's vertical coordinate controls the point that is zoomed in on, and its horizontal coordinate controls the rate of zooming; pointing to the left makes the view zoom out, allowing the correction of recent errors. From [1].

II. INPUT DEVICES CURRENTLY USED WITH DASHER

A. 2-D Continuous Input

DASHER was first developed to be driven by continuous two-dimensional (2-D) gestures, i.e., by directly controlling the position of a pointer on the screen. An origin near the center of the screen (and visible to the user as a crosshairs) is the null position: positioning the pointer there results in no motion of the DASHER landscape. If the user moves the pointer away from the origin, the interface zooms in towards the location pointed to by the vector from the origin to the pointer. The result is an intuitive navigation control in which the user simply steers DASHER towards the location containing the phrase they wish to write, and can control their speed by altering their distance from the origin. Whatever speed the user is comfortable with, DASHER will efficiently convert their vertical gestures into text.

2-D input devices used with DASHER include mouse, touch-screen, gazetracker, and head mouse (the latter tracks a reflective dot on the user's head, spectacles or cap). Under mouse control, novice users can reach writing speeds² of 25 words/min after 1 h of practice; expert users can write at 35 words/min [2]. Under eye control alone, users familiar with DASHER can write at 25 words/min [3], faster than any other gaze-writing system we are aware of. DASHER has many disabled users. One young man with cerebral palsy uses DASHER, driven with a head mouse, as his principal means of communication, and used it to write his undergraduate dissertation.

B. 1-D Continuous Input

For input devices offering a single continuous dimension of control, DASHER maps the one-dimensional (1-D) input onto a continuous curve within the normal 2-D control space. Midrange values of the input control the direction in which to zoom. Values towards the

²We follow the convention used in the human–computer interaction literature by which 1 word/min corresponds to 5 characters/min.

Manuscript received July 15, 2005; revised March 17, 2006; accepted March 20, 2006. The DASHER project is supported by the Gatsby Charitable Foundation.

The authors are with the Department of Physics, University of Cambridge, Cambridge CB3 0HE, U.K. (e-mail: saw27@mrao.cam.ac.uk; mackay@mrao.cam.ac.uk).

Digital Object Identifier 10.1109/TNSRE.2006.875573

¹Download DASHER from <http://www.inference.phy.cam.ac.uk/dasher>

extremes of the available range allow the user to zoom out and pause the interface.

One such 1-D control device is a “breath mouse” which we have developed for use with DASHER. The device measures the stretching and contraction of a partially-elasticated belt as the wearer breathes in and out, and converts this motion to a 1-D continuous signal. Using the breath mouse, an expert DASHER user can write at 16 words per minute. Novice users reached an average of 6 words per minute after 1 h of practice [4].

C. Discrete Inputs

Users who can activate buttons (virtual or physical) but cannot reliably provide a continuous output can use one of DASHER’s “button modes.” For example, the direct two-button mode maps one button to the action of zooming in on the top half of the visible DASHER landscape, and the second button to zooming in on the bottom half.

In button mode, DASHER converts bits from the user into written text at exactly the compression rate achieved by the language model. DASHER’s current language model PPMD5 compresses English text to around 2 bits per character [5]. Thus, on average, DASHER outputs one character for every two bits provided by the user’s button presses.

III. DASHER AS A BCI USER INTERFACE

A. Motivation

Current brain–computer interface (BCI) systems extract data from the user at a substantially lower information transfer rate than typical physical user interfaces. Consequently, it is especially important to make the best possible use of that information. DASHER offers an efficient method for converting the output of a BCI into text. DASHER can also use information about the reliability of the signals generated by the user.

DASHER’s language model can be initialized using text that is biased towards a limited set of phrases and words that the user is likely to wish to communicate. The user will be able to write these phrases, or variants of them, extremely quickly, while retaining the ability to write any other phrase should they wish to.

B. Continuous Control

Many BCI systems output a continuous 1- or 2-D signal which could be used to drive DASHER directly. DASHER is well-suited to a BCI signal which is likely to be under imperfect control of the user. In DASHER, users write by *navigating* to what they want to say, not by *selecting* letters or words. If the user accidentally steers in the wrong direction (either through an error in their intention or as a result of noise in the BCI system), they can correct their mistake by subsequent compensatory action. As with all navigation, all that matters is the final location arrived at.

C. Discrete Control

BCI systems that emit discrete events fall into two categories. The first category contains systems which internally convert a continuous variable into discrete outputs, for example, by measuring the average value of the variable in a trial period and classifying the result. Systems in which the user makes selections by driving a cursor to one of two or more on-screen targets fall into this category. We suggest that the best strategy for using DASHER with BCI systems in this category may be to use the continuous variable to drive DASHER directly, without an intermediate conversion into discrete options. Such systems would then come under the previous section.

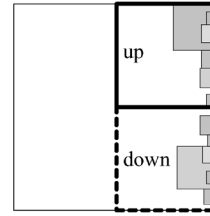


Fig. 2. Illustration of two SSVEP targets (heavy squares) overlaid on the DASHER landscape. Depending on which region the user attends to, DASHER zooms in on the top or bottom half of the screen.

The second category contains systems which are intrinsically discrete in nature. For example, both P300 and steady-state visually-evoked potential (SSVEP) interfaces determine which of several discrete visual targets the user is attending to. A natural way to use these techniques would be to paint P300 targets or SSVEP regions onto the DASHER landscape. For example, in the case of SSVEP, the right-hand half of the DASHER landscape could be covered by two or more regions flickering at different frequencies (Fig. 2). To zoom into one of these regions, the user attends to that region. The BCI system detects which region the user is attending to and causes DASHER to zoom in appropriately, and the cycle repeats. Likewise, P300 targets could be arranged down the right-hand side of the DASHER landscape, instead of in the commonly used speller grid [6].

We note that DASHER driven by an error-free discrete binary input will emit English text at an average rate of one character for every 2 bits communicated by the user (see Section II-C). This compares to the 5 bits needed for each character using a balanced binary decision tree to select letters [7]. (A Huffman-coded decision tree reduces this to an average of 4.2 bits per character ([8, p. 99].)

If the accuracy of the BCI system is high, then the optimal strategy for dealing with the rare errors that do occur may be to simply provide an additional target which instructs DASHER to undo the previous action (i.e., zoom out). Such a target needs to be present anyway, in case the user makes a mistake in selecting which region of the DASHER landscape contains the text they are trying to write. This is similar to the strategy of adding a “delete” node to a binary decision tree [9]. However, if the BCI misclassification rate is high, we suggest that the optimal strategy is to model the the BCI system as a noisy communication channel between the user and the computer, and to use information theory to inform the choice of an error-correcting code to use. For example, instead of accumulating evidence that the user is attending to a particular target over a single, long trial, it may be more efficient to run several shorter trials, each one individually less reliable. By varying the SSVEP frequencies on each target in each trial according to the coding scheme specified by the error-correcting code, the overall information transfer rate may be improved. This idea is explored further in [1].

D. Discussion

If a BCI signal can be used to select from a number of on-screen options, then various predictive/augmentative communications devices could in principle be driven by a BCI. DASHER offers a number of advantages over other assistive text-entry systems: it can use continuous-valued signals; it can express selection of a sequence of symbols as a continuous zooming process rather than a sequence of discrete events; it is designed using information theory so can take full advantage of a good adaptive language model; it works in all languages, and is free software.

One problem with driving DASHER by BCI may be that DASHER is a visually intensive task, requiring the visual processing of moving objects on the screen and imposing the cognitive load of searching

for the correct direction in which to zoom in. DASHER leverages this high-bandwidth *inward* communication channel to the user in order to improve the efficiency of the low-bandwidth *outward* channel. Whether these visual tasks will impede functioning of a BCI system remains to be discovered. This problem would be largely avoided in the discrete control case outlined above. In discrete mode, the DASHER interface moves only during brief zooming events. The system could alternate between periods in which the user studies DASHER in order to decide which section of the screen to zoom in on, and periods during which the BCI signal is measured in order to determine which target the user has chosen.

In contrast, a BCI user is less likely to become frustrated or inattentive when using DASHER than when using more repetitive paradigms such as the standard P300 speller. Trials with gazetrackers indicate that DASHER is considerably more fun, and less stressful, than on-screen keyboards.

IV. CONCLUSION

We wish to make the best possible use of the bits of information content that can be generated by severely disabled people. DASHER offers a paradigm for efficiently converting these bits to communication symbols. DASHER has proved its effectiveness for people able to use a gazetracker or make other motor actions. We believe that DASHER will be equally useful to users who retain functioning vision but are limited to communication through a BCI.

REFERENCES

- [1] D. J. C. MacKay, C. J. Ball, and M. Donegan, "Efficient communication with one or two buttons," in *Proceedings of Maximum Entropy and Bayesian Methods*, ser. Amer. Instit. Phys. Conf. Proc., R. Fischer, R. Preuss, and U. von Toussaint, Eds. Melville, NY: Amer. Inst. Phys., 2004, vol. 735, pp. 207–218.
- [2] D. J. Ward, A. F. Blackwell, and D. J. C. MacKay, "DASHER—A data entry interface using continuous gestures and language models," *Human-Computer Interaction*, vol. 17, no. 2–3, pp. 199–228, 2002.
- [3] D. J. Ward and D. J. C. MacKay, "Fast hands-free writing by gaze direction," *Nature*, vol. 418, no. 6900, p. 838, 2002.
- [4] T. H. Shorrock, D. J. C. MacKay, and C. J. Ball, "Efficient communication by breathing," in *Proceedings of the Sheffield Machine Learning Workshop*, ser. Lecture Notes in Artificial Intelligence, J. Winkler, N. D. Lawrence, and M. Niranjana, Eds. Berlin, Germany: Springer, 2005, vol. 3635, pp. 88–97.
- [5] D. J. Ward, "Adaptive computer interfaces" Ph.D. dissertation, University of Cambridge, Cambridge, U.K., 2001 [Online]. Available: <http://www.inference.phy.cam.ac.uk/djw30/papers/thesis.html>
- [6] L. A. Farwell and E. Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalograp. Clin. Neurophysiol.*, vol. 70, no. 6, pp. 510–523, Dec. 1988.
- [7] B. Obermaier, G. R. Miller, and G. Pfurtscheller, "'Virtual keyboard' controlled by spontaneous EEG activity," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 11, no. 4, pp. 422–426, Dec. 2003.
- [8] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
- [9] M. Tregoubov and N. Birbaumer, "On the building of binary spelling interfaces for augmentative communication," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 2, pp. 300–305, Feb. 2005.

ECoG Factors Underlying Multimodal Control of a Brain–Computer Interface

J. Adam Wilson, Elizabeth A. Felton, P. Charles Garell, Gerwin Schalk, and Justin C. Williams

Abstract—Most current brain–computer interface (BCI) systems for humans use electroencephalographic activity recorded from the scalp, and may be limited in many ways. Electroencephalography (ECoG) is believed to be a minimally-invasive alternative to electroencephalogram (EEG) for BCI systems, yielding superior signal characteristics that could allow rapid user training and faster communication rates. In addition, our preliminary results suggest that brain regions other than the sensorimotor cortex, such as auditory cortex, may be trained to control a BCI system using similar methods as those used to train motor regions of the brain. This could prove to be vital for users who have neurological disease, head trauma, or other conditions precluding the use of sensorimotor cortex for BCI control.

Index Terms—Brain–computer interface (BCI), electroencephalography (ECoG), sensorimotor cortex.

I. INTRODUCTION

Brain signals recorded from the electrocorticogram (ECoG) have many potential advantages for use with brain–computer interface (BCI) systems when compared to electroencephalogram (EEG). Our research is exploring the use of ECoG recorded from motor and nonmotor cortex to control a BCI. This paper presents preliminary evidence in support of this technique and describes further studies of ECoG-based BCI systems.

The potential advantages of using ECoG for BCI control are: 1) increased spatial resolution; 2) increased signal bandwidth; and 3) larger signal amplitude. Therefore, it may be possible to differentiate independent signals over a wide range of frequencies, on neighboring electrodes, using multiple strategies incorporating both motor and sensory imagery.

The control methodology used is based on the ability of subjects to voluntarily modulate one or more brain rhythms using imagery. Traditionally, motor imagery has been used because it was presumed to be the most accessible and reliable EEG signal. However, we propose that the advantages of ECoG will enable subjects to learn to use multiple modalities, including motor and sensory imagery, to control a BCI application. This would enable individuals with damage to the motor cortex due to stroke or other neurological disease to benefit from BCI systems.

We have the opportunity to evaluate this hypothesis because many of our subjects have electrodes placed over multiple nonmotor areas. Therefore, we are investigating the possibility of using non-motor imagery, focusing on auditory illusion combined with the more typical motor imagery task, while studying and utilizing unique ECoG principles.

Manuscript received July 19, 2005; revised March 25, 2006. This work was supported in part by the University of Wisconsin Graduate School and in part by the National Institutes of Health under Grant NIH K23 DC006415-01.

J. A. Wilson, E. A. Felton, and J. C. Williams are with the Department of Biomedical Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: jawilson@cae.wisc.edu; felton@cae.wisc.edu; jwilliams@engr.wisc.edu).

P. C. Garell is with the VA Hospital, Madison, WI 53706 USA (e-mail: garell@neurosurg.wisc.edu).

G. Schalk is with the Wadsworth Center, New York State Department of Health, Albany, NY 12201 USA, and also with the Rensselaer Polytechnic Institute, Troy, NY 12180 USA (email: schalk@wadsworth.org).

Digital Object Identifier 10.1109/TNSRE.2006.875570