Efficient communication by breathing

Tom H. Shorrock, David J.C. MacKay, and Chris J. Ball

Cavendish Laboratory, Cambridge, CB3 0HE, United Kingdom

Abstract. The arithmetic-coding-based communication system, Dasher, can be driven by a one-dimensional continuous signal. A belt-mounted breath-mouse, delivering a signal related to lung volume, enables a user to communicate by breath alone. With practice, an expert user can write English at 15 words per minute.

Dasher is a communication system based on a beautiful idea from information theory called arithmetic coding (Witten et al., 1987; MacKay, 2003, Chapter 6). Arithmetic coding is an optimal method for text-*compression* using a language model. By turning arithmetic coding on its head, we obtain an optimal method for text-*generation*.

We view a person's gestures as a source of information, and the sentences they wish to communicate as the sink of information. Good interface design maximizes the number of bits per second that are conveyed from the user into text. Poor interfaces waste the user's time either by failing to extract all the bits that the user could easily generate, or by diverting the user's bits into redundant activity.

The Dasher approach to interface design decouples the issues of efficient bit-generation and efficient language-generation. Unlike in most interfaces, a Dasher-user's gestures have no relationship to particular symbols in the language. Instead, they control *navigation* in a continuous space whose contents are laid out using a language model. For demonstrations, or to try Dasher for yourself – it's free – please visit `www.inference.phy.cam.ac.uk/dasher/`.

The objective of this paper is to offer a new method for helping a disabled person to communicate by breath alone. In contrast to widely used switch-scanning systems, our method makes use of fine breathing control. Of course, not everyone has fine breath control, but to those who have, we would like to offer the chance to make use of that information, rather than discard it.

1 How Dasher works

Imagine writing a piece of text by going into the library that contains *all possible books*, and finding the book that contains exactly that text. In this way, writing can be turned into a navigational task. What is written is determined by where the user goes. In Dasher's idealized library, the 'books' are arranged alphabetically on one enormous shelf. When the user points at a part of the shelf, the view zooms in continuously on that part of the shelf. To write a message that

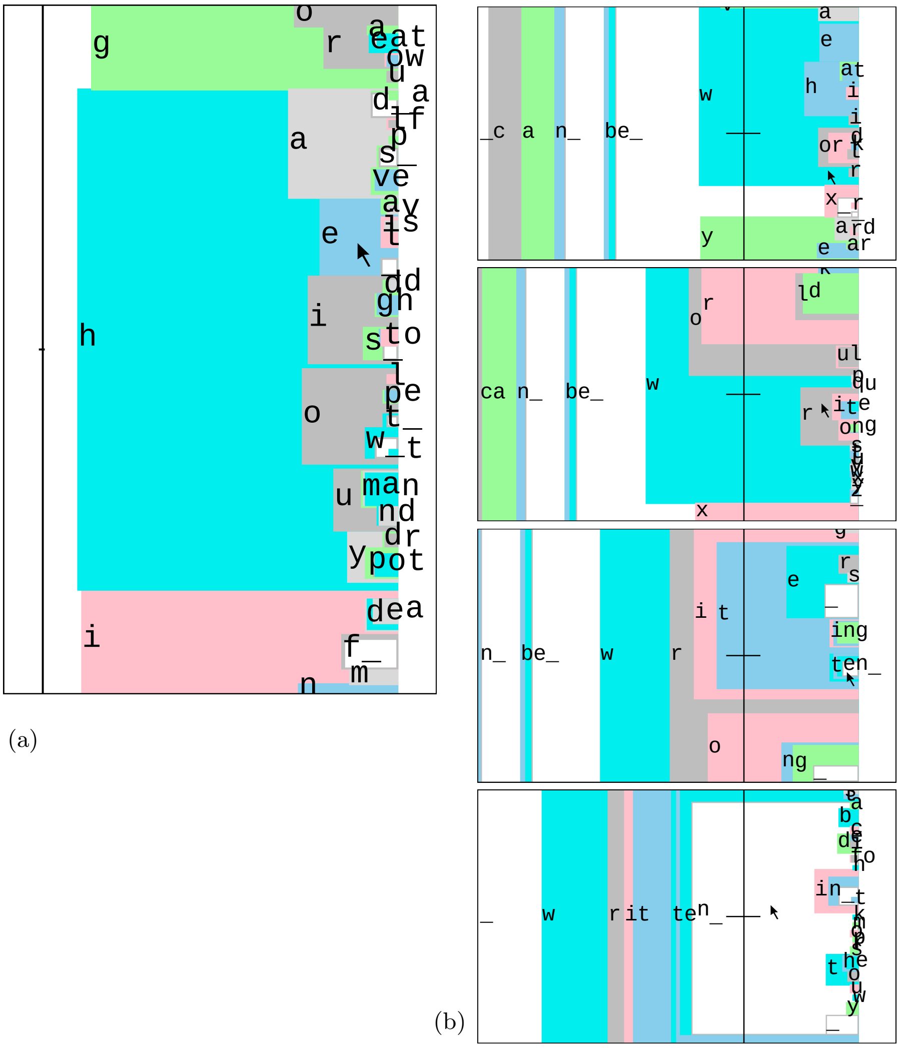


Fig. 1. A Screenshot of Dasher when the user starts writing `hello`. The shelf of the alphabetical 'library' is displayed vertically. The space character, '_', is included in the alphabet after `z`. In panel (a), the user has zoomed in on the portion of the shelf containing messages beginning with `g`, `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. The 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.
Panel (b) shows screenshots while the user writes '`any sentence can be written`'.

begins ‘`hello`’, one first steers towards the section of the shelf marked `h`, where all the books beginning with `h` are found. Within this section are sections for books beginning `ha`, `hb`, `hc`, etc.; one enters the `he` section, then the `hel` section within it, and so forth.

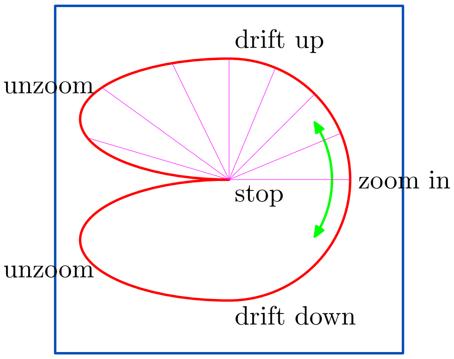


Fig. 2. Dasher’s one-dimensional mode. The curved line shows the sequence of two-dimensional control positions created throughout the range of one-dimensional control positions. The central point of the display corresponds to no motion; the two ends of the one-dimensional scale both map to this point. The centre of the one-dimensional scale is mapped to the three-o-clock position, zooming in at the maximum rate. The horizontal coordinate on the curve determines the rate of zooming in or out. The radial lines indicate the direction of motion produced for some positions along the upper half of the range.

To make the writing process efficient we use a language model, which predicts the probability of each letter in a given context, to allocate the shelf-space for each letter of the alphabet, as illustrated in figure 1. The shelf is recursively chopped up in such a way that the amount of shelf-space devoted to a string is proportional to its probability. The user’s gestures are turned into steering commands, controlling the portion of the display zoomed into. If the user can generate information at a rate of, say, 5 bits per second, then our aim is to feed these bits to Dasher in such a way that, each second, the display zooms in by a factor of $2^5 = 32$ on the region containing the text required by the user. When the language model’s predictions are accurate, many successive characters can be selected by a single gesture. The language model we use, PPMD5 (Cleary and Witten, 1984; Teahan, 1995), generates text at an exchange rate of about two bits per character. Thus the user will be able to write at 5/2 characters per second, or 30 words per minute. We could only beat this writing speed by enhancing the rate at which the user generates bits, or improving the predictions of the language model.

Dasher was first developed to be driven by continuous *two-dimensional* gestures, delivered via a mouse, touch screen, or gazetracker. Our experiments showed that, with Dasher, it is easy to spell correctly and hard to make spelling mistakes. Using an ordinary mouse, typical novice users reach a writing speed of 25 words per minute after 60 minutes of practice, and expert users can write at 35 words per minute (Ward et al, 2002). Results using Dasher with a gazetracker were record-breaking: after 60 minutes' practice, novice users can drive Dasher using a gazetracker at a speed of about 15 words per minute; expert users can write at 25 words per minute, and make almost no spelling mistakes (Ward and MacKay, 2002). We know of no faster method for communication by gaze.

In this paper, we discuss how Dasher can be driven by *one-dimensional* gestures.

2 Dasher's one-dimensional mode

In normal two-dimensional Dasher, the information content concerning the text desired by the user is conveyed entirely through the vertical dimension of the pointer. The horizontal dimension controls only the speed of text entry. Expert users of Dasher tend to write at a fairly constant zooming rate such as five bits per second. Thus the horizontal dimension is scarcely used: an expert uses it only if he makes a mistake or wishes to slow down, pause or unzoom.

In Dasher's one-dimensional mode, we select a simple one-dimensional curve from regular Dasher's two-dimensional navigation space; the single dimension conveyed by the user selects the steering direction from this curve (figure 2). The middle of the curve offers normal forward motion at a fixed zooming rate, with the one-dimensional coordinate determining the direction of forward motion. The extreme ends of the curve offer unzooming. As the control is moved from one end to the other, unzooming blends smoothly into drifting up without zooming, zooming up, zooming straight forward, zooming down, drifting down, and unzooming again. (The curve is composed of three half-ellipses.)

We can include *control nodes* in the Dasher alphabet so that the user can access special functions such as pausing and stopping by the same zooming process as is used for writing (much as an escape key can be used to access special modes in a keyboard-based editor). (Such control nodes were not used in the experiments described in the present paper.)

3 Experiments with a breath mouse

We obtained a continuous one-dimensional breathing signal using a breath mouse (figure 3).

3.1 Experiments on novices

Eight volunteers from the Cavendish Laboratory staff with very little or no experience with Dasher used breath-Dasher for a total of one hour. Of the volunteers,

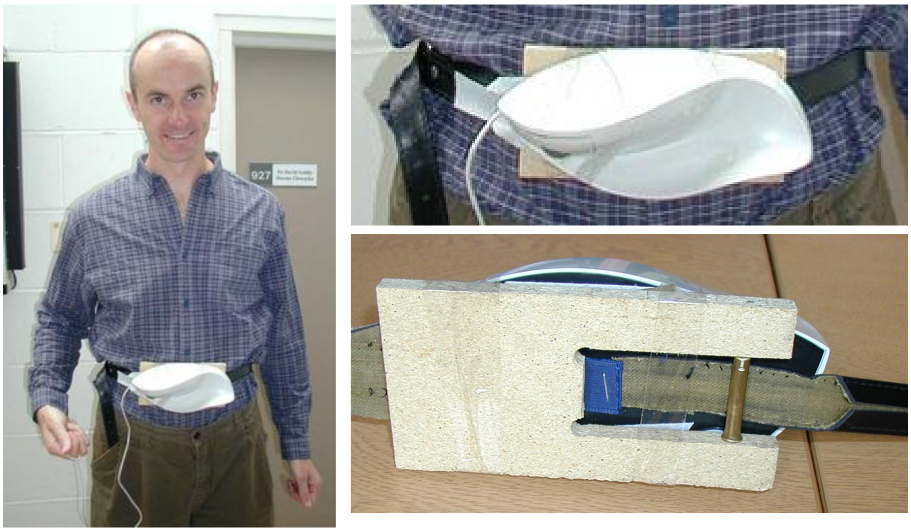


Fig. 3. Our first breath mouse, made from an optical mouse, a belt, and a piece of elastic. The mouse is fixed to a piece of wood, to which a belt is also attached. Two inches of the belt are replaced by elastic, so that changes in the waist circumference produce motion of the belt underneath the eye of the mouse. This sensor measures breathing if the user breathes using their diaphragm (rather than their rib cage). We oriented the mouse so that breathing in moves the on-screen mouse up and rotates the pointer anti-clockwise along the curve; and breathing out moves the on-screen mouse down and rotates the pointer clockwise. The sensor also responds to clenching of the stomach muscles, but we encourage the user to navigate by breathing normally.

four were women. Six had English as their first language; one, German; and one, Italian.

Our protocol was similar to that of Ward and MacKay (2002). We gave users dictation from Jane Austen's *Emma* in five minute periods. Dasher's language model was trained on *Emma*, excluding the dictated passages. We used a 54-letter alphabet (the twenty six letters in both upper and lower case, the space character and the full stop). Dasher was started and stopped manually at the beginning and end of each dictation period.

Each subject's twelve five-minute dictation trials were spaced out over several days. Two trials could be taken one after one another in a single session, with a few minutes' break between. The volunteers were allowed up to two sessions each day, with a maximum of three days between two consecutive trials. In one case, three sessions were conducted on a single day, with at least three hours separating successive sessions.

Before dictation all volunteers were allowed to read a paper copy of the text that they were expected to write, to try to reduce the frequency of writing-errors *not* associated with using Dasher.

After each dictation trial, the subject was offered the chance to adjust the overall speed of the interface by 5 or 10%.

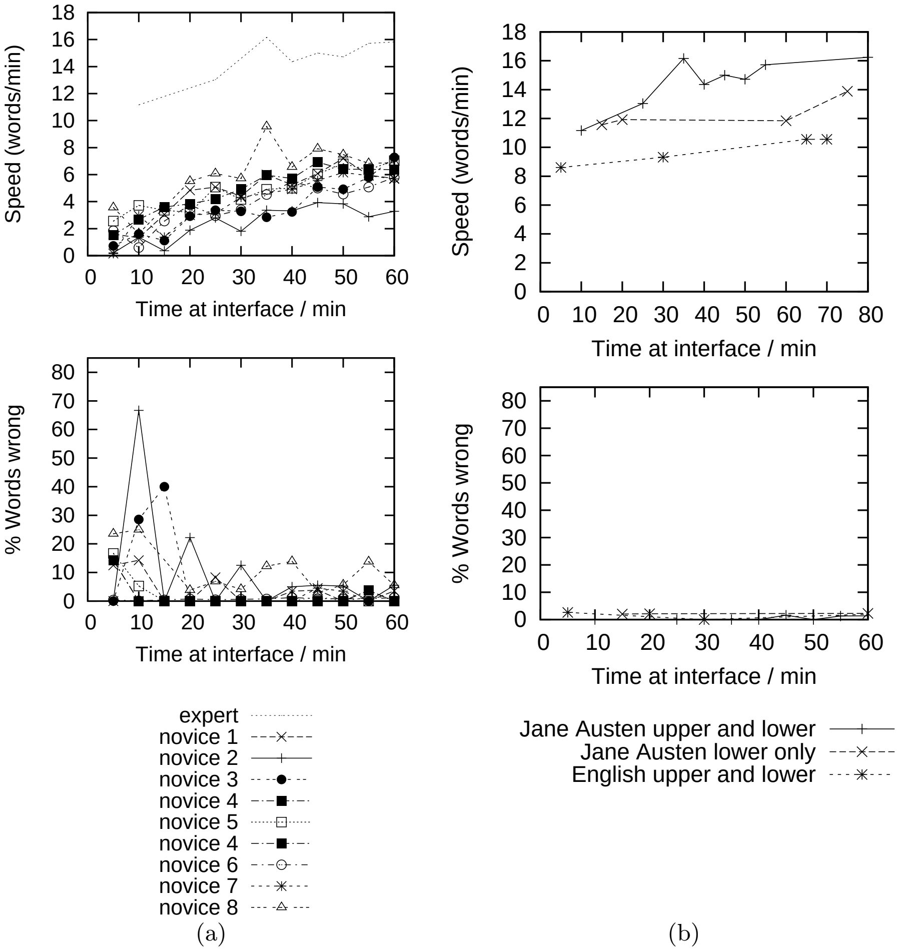
The writing speeds and error rates for all 8 novices and one expert are shown in figure 4a. Figure 5 shows the speed settings chosen by the users.

Observations Two volunteers (novices 2 and novice 3) had difficulty controlling the breath mouse. We believe they sometimes clenched their stomach muscles instead of breathing naturally.

Most novices had difficulty finding the full stops. Users had relatively little practice in using them since in early trials a single sentence was often not completed. This inexperience was compounded by the large probability of a space character following the full stop, causing the users to not notice the full stop. In early experiments the location of the full stop often had to be pointed out. Although this problem reduced with experience, of all the letters this was the most persistently troublesome. Users generally dealt with capital letters well.

Some users had difficulties at low speeds because while the low speed was necessary for them to find their place, once the correct direction had been determined they found the wait for the interface to zoom uncomfortable. Some users made use of the feature that the interface could be stopped by breathing right in or out, to give themselves time to find their place.

When the speed control was set at a low speed, zooming out at extremes of breath intake was found uncomfortably slow. With experience, this problem diminished, firstly because users noticed their mistakes earlier and so did not need to unzoom so much, and secondly because their speed was increased so they did not have to hold their breath for so long.



(a) (b)

Fig. 4. (a) Breath-Dasher results for 8 Dasher novices and 1 Dasher expert. Upper graph shows writing speed in words per minute. Lower graph shows the percentage of words containing errors. (b) Expert user: results for different training texts and alphabets.

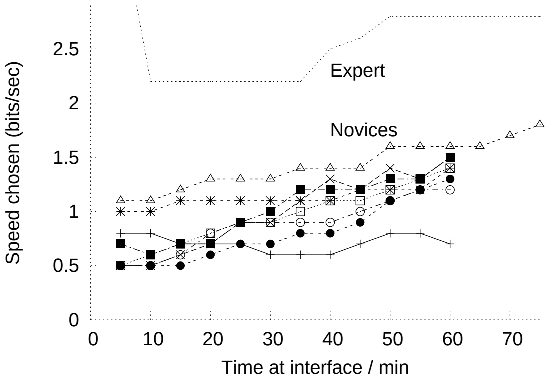


Fig. 5. Dasher speed settings chosen by each user at the start of each trial. Dasher's maximum speed is specified in bits per second.

3.2 Expert trials

An expert who was very familiar with Dasher (with perhaps 50 hours of use) and had considerable experience of the breath mouse (about two hours of practice before the experiment started) was also tested. We measured his performance using three different combinations of alphabet and training text, so as to quantify the effects of (1) including upper and lower case characters; (2) choosing a training text that is well matched to the dictation text.

Figure 4b shows the results.

Alphabet choice The top, solid line in figure 4b shows the results where the training text and alphabet were identical to those used by the novices. The second line (with crosses) shows results where the alphabet was lower-case only; the training text was the same *Emma* corpus. It is striking that *increasing* the number of characters from single-case to mixed-case, which doubles the number of letters available, actually *increases* the rate of writing.

The explanation for this result is that the mixed-case language is easier for our language model to predict. Even though the number of possible characters is twice as great, the entropy of the mixed-case language is slightly smaller. The cost of selecting occasional upper-case characters is offset by the increased predictive power of mixed-case contexts.

Training text The expert also took dictation of *Emma* using a Dasher system that had been trained on generic English text (the default 300 kilobyte training file of assorted English sentences from the Dasher website).

The lowest line in figure 4b shows that the writing speed drops by about 33% when a generic training text is used. Users can therefore expect a 50% increase in speed if they pre-train Dasher with texts similar to what they intend to write.

3.3 Comparison with sip-and-puff

Beginner users of Dasher wrote at 6.0 ± 1.3 words per minute after an hour's training, with on average 2.0% of words misspelled. An expert user can write at over 16 words per minute.

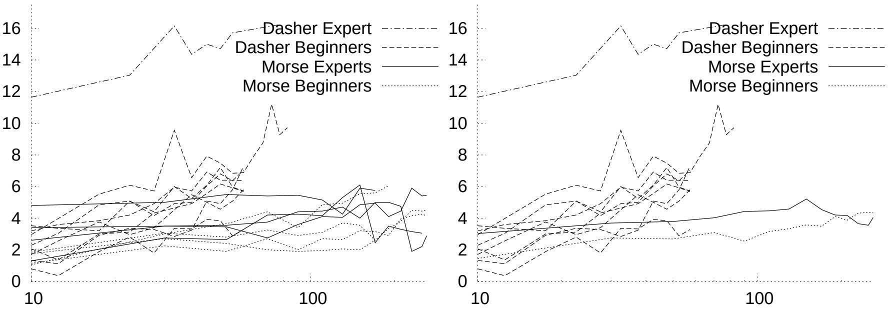


Fig. 6. Breath-Dasher writing speeds compared with writing speeds achieved by 8 sip-and-puff Morse code users. Horizontal axis shows time using the interface in minutes, *on a logarithmic scale*. Vertical axis is writing speed in words per minute. In the righthand panel, the results for the Morse users have been averaged. Morse data kindly provided by Denis Anson (Anson et al, 2003).

For comparison, one method for writing by breath is 'sip-and-puff', with sips and puffs being mapped to the dots and dashes of Morse code. One experienced sip-and-puff user reports that he can write at 17 words per minute when using a combination of Morse code and word-completion software.[1] Data on learning curves for this method were kindly provided by Denis Anson. The study by Anson et al (2003) involved 8 subjects, four of whom had no prior experience with Morse, and four of whom were radio hams with Morse experience. All subjects wrote for more than 180 minutes in 20-minute trials. The sip-and-puff with Morse writing method required no visual feedback, but did use auditory feedback: users could hear the dots and dashes they entered. The learning curves for sip-and-puff Morse are compared with those for breath-Dasher in figure 6. The plateau writing speeds reached by Morse code novices were 4.9, 2.2, 4.1, and 5.7 words per minute, with error rates of 5%, 4%, 0%, and 4%, respectively. The plateau values of Morse code experts were only a little better: 4.9, 5.2, 5.3, and 6.4 words per minute, with error rates of 3%, 4%, 5%, and 2% respectively.

Another widely-used method for communication by sip-and-puff is a scanning system that offers the users sequences of discrete menus to select from. Vanderheiden (1985) reported that users of scanning systems wrote at six or less words per minute; we know experienced users who can write at 12 words per minute by scanning, but have not been able to find full learning curves for this method.

[1] `http://www.makoa.org/jlubin/ahfeat4.htm`

We conclude that Dasher has a better learning curve than sip-and-puff with Morse. Dasher is a promising writing method for a sip-and-puff user who could convey a continuous signal with their breath. An alternative approach that would use standard sip-and-puff hardware would be to use one of the two-button modes of 'button-Dasher' (MacKay et al, 2004).

3.4 Development ideas

In the light of users' complaints that they occasionally ran short of breath when using breath Dasher, we propose to include the option for breath-Dasher to add a 0.1 Hz periodic signal to the one-dimensional coordinate. To steer Dasher as before, the user will have to breathe in and out to cancel the effect of this added signal.

We hope the one-dimensional mode of Dasher will also be useful for hand-held computers with tilt sensors.

Dasher is free software, distributed under the GNU General Public License, and available from `www.inference.phy.cam.ac.uk/dasher/`.

Acknowledgments

We thank Caroline Gray, David Colven, Alan Blackwell, and Matthew Garrett for helpful discussions.

We gratefully acknowledge the support of the Gatsby Charitable Foundation, Martin King, and the Nine Tuna Foundation.

References

1. Witten, I. H., Neal, R. M., and Cleary, J. G., Arithmetic coding for data compression, *Communications of the ACM*, **30**, 520–540 (1987).
2. MacKay, D. J. C., *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003, available from `www.inference.phy.cam.ac.uk/mackay/itila/`.
3. Cleary, J. G., and Witten, I. H., Data compression using adaptive coding and partial string matching, *IEEE Trans. on Communications*, **32**, 396–402 (1984).
4. Teahan, W. J., Probability estimation for PPM (1995), Probability estimation for PPM. In *Proc. of the N.Z. Comp. Sci. Research Students' Conf.,* available from `citeseer.nj.nec.com/teahan95probability.html` (1995).
5. Ward, D. J., Blackwell, A. F., and MacKay, D. J. C., Dasher – A data entry interface using continuous gestures and language models, *Human-Computer Interaction*, **17**, 199–228 (2002).
6. Ward, D. J., and MacKay, D. J. C., Fast hands-free writing by gaze direction, *Nature*, **418**, 838 (2002).
7. Anson, D. K., Glodek, M., Peiffer, R. M., Rubino, C. G., and Schwartz, P. T., Long-term speed and accuracy of Morse code vs. head-pointer interface for text generation (2003), presented at RESNA 2004 Annual Conference, Orlando, FL. Available from: `atri.misericordia.edu/Papers/MorseVrsOnScreen.php`.

8. Vanderheiden, P. J., “Writing aids,” in *Electronic aids for rehabilitation*, edited by J. Webster, A. Cook, W. Tompkins, and G. Vanderheiden, Chapman and Hall, London, 1985, pp. 262–282.
9. MacKay, D. J. C., Ball, C. J., and Donegan, M., “Efficient communication with one or two buttons,” in *Proceedings of* Maximum Entropy and Bayesian Methods, edited by R. Fischer, R. Preuss, and U. von Toussaint, American Institute of Physics, Melville, New York, 2004, vol. 735 of *AIP Conference Proceedings*, pp. 207–218.