Dasher's One-button Dynamic Mode – Theory and Preliminary Results

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 single switch. In MacKay et al. (2004), we proposed two versions of one-button Dasher, 'static' and 'dynamic', aimed at a theoretical model of a user who can click with timing precision g, and who requires a recovery time D between clicks. While developing and testing those two versions we invented a third way of using one button, which we present here. This new dynamic mode, is much simpler than the dynamic mode presented in MacKay et al. (2004), and has a direct theoretical connection to our model user. This paper explains the theory, and gives preliminary experimental results for a single user on a prototype implemented in Dasher 3 in July 2007. This method of driving Dasher has been included in Dasher 4 (released 2006) under the name 'one-button dynamic mode'.

In MacKay et al. (2004), we presented a theoretical model of a single-switch user. We also presented two ad-hoc ideas for connecting Dasher to one switch, and plotted graphs indicating how close these ideas would get to the theoretical capacity of the idealized user. One idea involved scanning through a static continuum of navigation options. The other, which we dubbed metronome mode, allowed the user to alter the direction Dasher was moving in, but this mode had many ad hoc features. We implemented and tested both these ideas, and eventually realised that a more elegant solution was possible, which we now present.

This solution was found by thinking carefully about what information theory has to say about the capacity of a single switch.

1 How to use one button in Dasher?

As in MacKay et al. (2004), we assume that the user controls only the times of presses, not the times of releases. We model the user with two parameters: a timing accuracy g, and a recovery time D. The user clicks within a time $\pm g/2$ around each intended click time, and then requires a delay of duration D before she can click again.

We assume that the user *cannot* distinguish between short and long presses, except perhaps 'on special occasions' (for example, to indicate the desire to stop using Dasher).

As in MacKay et al. (2004), we chop time into boxes of duration g. We can then think of the user's actions as consisting as a sequence of choices between **two** actions, '0' (doing nothing, which takes time g), and '1' (pressing the switch, which takes time D). (See figure 1.) As soon as an action has been completed, the user is free to make another choice.

Given the two action durations g and D, it is a simple exercise in information theory to find the optimal probabilities with which the two actions should be used. (MacKay 2003, p. 125)

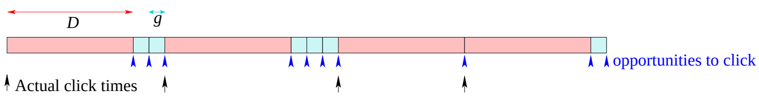

Fig. 1. Model of precise single-switch-pressing as a sequence of selections of brief '0' events and lengthy '1' events.

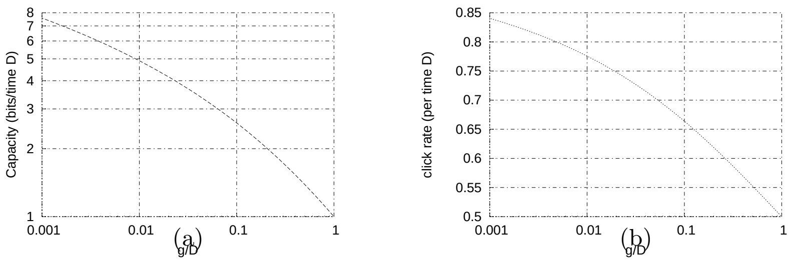

Fig. 2. (a) Capacity of the theoretical model as a function of D/g (in bits per duration D).
(b) Clicking rate, assuming optimal use, as a function of D/g (in clicks per duration D).

The Capacity is C, where

$$p_1 = 2^{-CD},$$

$$p_0 = 2^{-Cg}$$

and

$$p_1 + p_0 = 1.$$

Figure 2 shows the capacity and the optimal clicking rate as a function of D/g.

It's interesting to ask what the information **rates** are 'during a `1`', and 'during a `0`'. The information content conveyed by choosing 1 `1` is $\log_2 \frac{1}{p_1}$, and it occupies a duration D. So the information rate 'during a `1`' is $\frac{\log_2 \frac{1}{p_1}}{D}$ and the information rate 'during a `0`' is $\frac{\log_2 \frac{1}{p_0}}{g}$; both are equal to C bits per unit time.

This constancy of information rate is probably obvious in retrospect, but it has a profound implication in terms of Dasher dynamics: when Dasher is being driven optimally by two actions labelled `0` and `1`, having the probabilities and durations above, *Dasher should zoom at a steady rate.*

This observation motivates a very simple idea: Dasher should zoom in steadily, with the interval displayed shrinking alternately first at one end and then at the other, with the alternation of which end is being shrunk being controlled by the timing of the user's clicks.

2 Free composition results

We implemented two prototypes based on this idea. The simplest 'steady' prototype zoomed at a steady rate (for example 1.7 bits per second), exactly as suggested above. Button presses caused the location being zoomed in on to alternate between the top and the bottom of the Dasher screen. We also implemented a 'pulsing' prototype, in which the zooming rate was initially quite fast after a press, until a zoom by a large factor had taken place (a factor of 5 or so in about half a second); then the zooming rate became slower (about 0.5 bits per second, say).

One expert Dasher user (DJCM) tested these two modes by writing freely for sessions lasting a few minutes each for a total of two hours. The user occasionally switched between the steady and pulsing prototypes, and was free to adjust the zoom-rate parameter at the end of each composition.

In addition to the single switch used to achieve navigation, two other switches were provided: one to stop and start Dasher (used only at the beginning and end of a composition); and an emergency 'cheat' switch, whose effect was to temporarily cause Dasher to unzoom. This cheat switch permitted error correction if the user clicked too late. The aim was to use the cheat switch as little as possible. [Alternative methods for error correction in one-button dynamic mode have been included in Dasher 4.]

Figure 3 shows the results. The top graph shows the writing speed in words per minute; the second graph shows the clicking rate in clicks per minute; the third graph shows the number of characters generated per click – note for comparison that a regular 50-key keyboard and a disambiguating keyboard with 12 keys ('T9') both generate roughly 1 character per click, so the results shown here are remarkably efficient in terms of the gestures required per character; the bottom two graphs show the error rate (fraction of words mis-written) and the cheating rate (the number of presses of the 'cheat' button, per click).

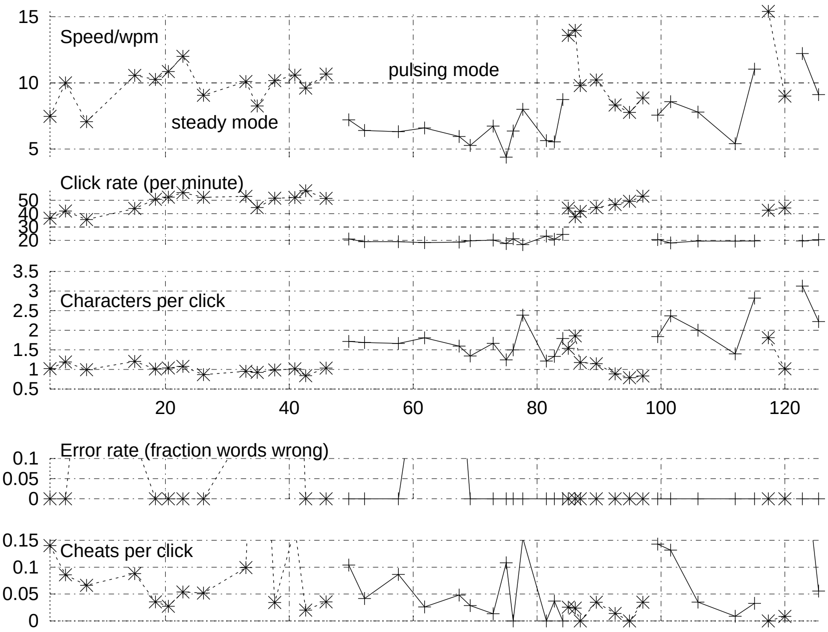

Fig. 3. Free composition results for one user. The user was expert in using other versions of Dasher. Horizontal axis: number of minutes using one-button dynamic mode.

The 'steady' mode required less cheating, and achieved a slightly higher writing speed (in words per minute). 'Pulsing' mode delivered a slightly higher number of characters per click.

3 Morse comparison

Using Dasher's one-button dynamic mode, the user achieved roughly **10 words per minute** with a very low gesture rate: roughly **0.4 gestures per character** or **20 gestures per minute**. How does this compare with Morse code, which is also a one-switch communication method using timings?

If English is written in Morse, the gesture rate is roughly 2 gestures per character (counting each dot and dash as a single gesture). Thus to write a 10 words per minute in Morse, a user **must** make 100 gestures per minute. The gesture rate of one-button Dasher is thus five times smaller than Morse. [Note also that Morse requires the ability to distinguish short and long presses, which one-button Dasher does not; and Morse does not ordinarily distinguish upper and lower case, wherease Dasher works in any alphabet.]

4 Connections

The question of how to communicate efficiently through a single switch is identical to the question of how to communicate efficiently over a neuronal link MacKay and McCulloch (1952).

The idea that spiking neurons mainly communicate information to each other through their *rates* of spiking rather than through the timings of action potentials seems an awfully wasteful use of resources, akin to forcing a disabled person to communicate a choice of A or B or C or . . . Z by clicking a switch at a particular rate.

Acknowledgments

We thank Mick Donegan, Caroline Gray, David Colven, Alan Blackwell, Radford Neal, 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. MacKay, D. J. C., Ball, C. J., and Donegan, M., "Efficient communication with one or two buttons," in *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.
2. MacKay, D. J. C., *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003, URL www.cambridge.org/0521642981, available from `www.inference.phy.cam.ac.uk/mackay/itila/`.
3. MacKay, D. M., and McCulloch, W. S., The limiting information capacity of a neuronal link, *Bull. Math. Biophys.*, **14**, 127–135 (1952).