# Efficient Communication Through The Timings of One Or Two Buttons


Phil Cowans  David MacKay
James Mead

Cavendish Laboratory,
19 J J Thomson Avenue
University of Cambridge,
United Kingdom,
CB3 0HE
pjc51@cam.ac.uk, mackay@mrao.cam.ac.uk, jmgm3@cam.ac.uk


September 22, 2006


**Abstract**

Dasher is a communication system in which writing is a navigational process. Here we evaluate versions in which navigation is controlled using time critical presses of one or two buttons. Using the two button mode, after 1 hour of practice, novice users can write at up to 14 words per minute with virtually no spelling mistakes. Experts, with over 5 hours of experience, can write as fast as 25 words per minute. Preliminary results are presented for the one button mode, novice users achieving a maximum writing speed of 17 words per minute. In both cases it is concluded that Dasher is a very gesture efficient communication system, ideal for users for whom every gesture is an effort.


## 1 An Introduction To Dasher

Dasher is a gesture efficient keyboard alternative based on Arithmetic Coding (for an account see [6] or [1]). 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, the sink of information. Good interface design maximizes the number of bits conveyed per second from the user into text. Poor interfaces waste the user's time either by failing to extract all the bits 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. To try Dasher out yourself (it's free) please visit *www.inference.phy.cam.ac.uk/dasher/*. In this paper we explore a new mode of Dasher that allows navigation through the use of two buttons and present preliminary results of another mode that allows navigation through the use of just one button.

The first section of the paper (sections 2-5) describe analysis performed of two button mode. Section 6 describes the one button mode.

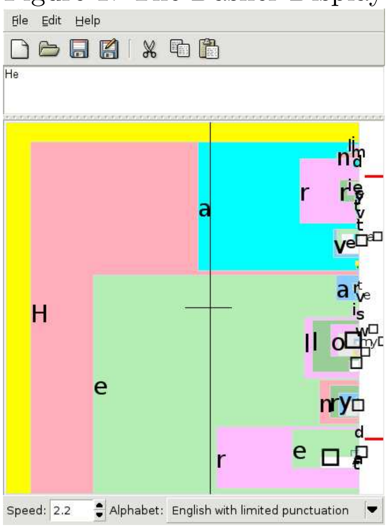
## 2 How Two Button Dasher Works

Imagine writing a piece of text by going into a 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 depends on where the user goes. In Dasher's idealized library, the 'books' are arranged alphabetically on one shelf.

In Dasher's two-button dynamic mode, navigation is achieved as follows. There are two 'fiducial' markers (horizontal red lines – they can be seen in figure 1) to the right of this display. The view zooms continually onto the central part of the display until the user presses a button, at which point the view shifts so as to centre the point of the display adjacent to the selected fiducial. The two buttons control which fiducial is selected; one is associated with each fiducial. To write a message, one first waits for the view to zoom until the desired location is adjacent to a fiducial, at which point the relevant button is pressed.

The writing process is made efficient by the use of a language model, which predicts the probability of the use of each letter within a given context and allocates shelf space accordingly. The language model used by Dasher is PPMD5. A picture of the Dasher interface while writing the word 'hello' is shown below in figure 1,

Figure 1: The Dasher Display



Both modes of button Dasher are aimed at people for whom every click is an effort - they are able to make precise clicks, but only at a low frequency. Precise clicks are 'rewarded', as they convey a lot of information. In this paper, an underlying theme is the ability of button Dasher to generate a large amount of information per gesture.

## 3 Theoretical Analysis of Two Button Mode

$\tau$ is the time taken to enter one bit of information. The speed of Dasher, measured in bits per second (bps) is $\frac{1}{\tau}$. The efolding time, $\tau_e$, is defined as $\frac{\tau}{\ln 2}$. Distances are measured in terms of screen coordinates; the centre point of the screen is at a vertical height 0, the top of the display at a vertical height 1, and the bottom at a vertical height -1.

To write a word, the user aims to click when the fiducial is adjacent to the desired point on the screen. However, there will be an error associated with this clicking action - the user clicks when the fiducial is a certain distance away from the desired location (where the distance is measured in terms of the display coordinates). If we denote the error immediately after the click as $\epsilon_0$, then after a time, t, spent zooming , the vertical distance between the centre of the display and the destination, will be

$$\epsilon(t) = \epsilon_0 e^{\frac{t}{\tau_e}} \tag{1}$$

At time T the fiducial is aligned with the desired location. The fiducial position, measured in screen coordinates is $\phi$.

$$\phi = \epsilon_0 e^{\frac{T}{\tau_e}} \tag{2}$$

The user does not click exactly at the optimum time, T, but instead at a slightly different time, $T_{\text{click}}$. The difference, $t_{error} = T_{\text{click}} - T$ is modelled as a bi-exponential distribution with mean $\mu$ and characteristic width $\sigma$. Denote, D, as the minimum time between clicks the user can sustain, i.e. the maximum clicking rate is $\frac{1}{D}$. Failure (i.e. having to reverse or being unable to write accurately) will result if the clicking rate required by Dasher, $\Gamma = \frac{1}{T}$, exceeds $\frac{1}{D}$. Using (2) to get an expression for $\frac{1}{T}$, and requiring this to be greater than $\frac{1}{D}$, the following inequality is obtained, which expresses the condition that must be satisfied for a user to fail,

$$\tau_e \ln \left| \frac{\phi}{\epsilon_0} \right| < D \tag{3}$$

A further 'speed' can be defined as $\frac{\partial \epsilon}{\partial t} = \frac{\epsilon(t)}{\tau_e}$. This determines how fast the distance between the desired location and the screen centre grows with respect to the on-screen vertical coordinates. Taking $\epsilon(t) = \phi$, the distance of the fiducial from the desired location (the error) at the time of clicking $T_{click}$, is given by the timing error multiplied by the aforementioned speed,

$$\epsilon_0 = \frac{t_{error}\phi}{\tau_e} \tag{4}$$

Substituting for $\epsilon_0$ in (3) gives the condition for failure,

$$t_{error} > \tau_e e^{-D/\tau_e} \tag{5}$$

It is now possible to derive an equation, which can be solved for $\tau_e$, given knowledge of parameters $\mu$, $\sigma$ and D. The speed is set such that on any given click, the probability of failure is less than $\delta$. So,

$$p(|t_{\text{error}}| > \tau_e e^{-D/\tau_e}) < \delta \tag{6}$$

where,

$$p(t_{\text{error}}) = \frac{1}{2\sigma} e^{(-|t_{error}-\mu|/\sigma)} \tag{7}$$

Integrating the probability distribution,

$$\begin{aligned} p(|t_{\text{error}}| > \tau_e e^{-D/\tau_e}) &= \int_{\tau_e e^{-D/\tau_e}}^{\infty} \frac{1}{2\sigma} e^{-|t_{\text{error}}-\mu|/\sigma}\, dt_{\text{error}} \\ &+ \int_{-\infty}^{-\tau_e e^{-D/\tau_e}} \frac{1}{2\sigma} e^{-|t_{\text{error}}-\mu|/\sigma}\, dt_{\text{error}} \end{aligned} \tag{8}$$

As $D \approx \tau_e$ and $\tau_e >> \mu$ (as $\mu$ is of the order of milliseconds),$\tau_e e^{-D/\tau_e} > \mu$, so the modulus signs in (8) can be ignored. On integration, setting the result equal to $\delta$, the result is, after some rearrangement,

$$\tau_e e^{-D/\tau_e} = \sigma \ln\left(\frac{1}{\delta}\cosh\left(\frac{\mu}{\sigma}\right)\right) \tag{9}$$

This equation can be solved for $\tau_e$ and thus the zooming speed of Dasher.

It is possible to use the above analysis to calculate an upper bound to the average clicking rate, denoted $\langle\Gamma\rangle = \left\langle\frac{1}{T}\right\rangle$, at a given speed. Using (2),

$$\langle\Gamma\rangle = \frac{1}{\tau_e \left\langle \ln\left(\frac{\phi}{\epsilon_0}\right)\right\rangle} \tag{10}$$

Using Jensen's Inequality, $\langle \ln\left(\frac{\phi}{\epsilon_0}\right)\rangle \geq \ln\left(\frac{\phi}{\langle\epsilon_0\rangle}\right)$. From (4), $\langle\epsilon_0\rangle = \frac{\phi\langle t_{\text{error}}\rangle}{\tau_e}$. As $\langle t_{\text{error}}\rangle = \mu$ the bounding inequality is,

$$\langle\Gamma\rangle \leq \frac{1}{\tau_e \ln\left(\frac{\tau_e}{\mu}\right)} \tag{11}$$

Thus accurate clickers (those with low $\mu$) can convey a greater amount of information per click, and so it takes a smaller number of gestures to transmit a given amount of information relative to users who click less accurately. So the two button mode is aimed at people who can make a limited frequency of gestures, but are able to make them accurately. The upper bound of (11) is plotted for a variety of $\mu$ in figure 2,

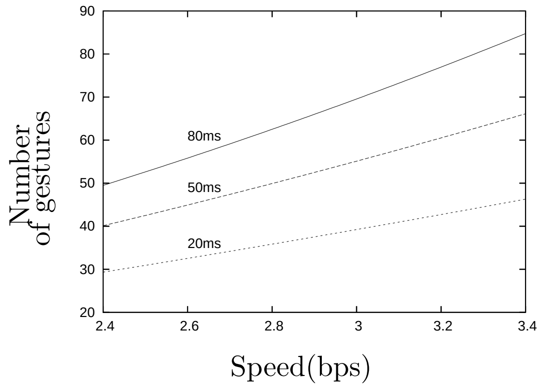



Figure 2: Predictions of (11) for different values of $\mu$

# 4 User Trials on Novice Subjects

The first set of user trials aimed to test how people improved in using two button mode over a period of 1 hour.

## 4.1 Experimental Procedure

1. Subjects

   Two male and one female subject were used. None had experience with two button Dasher. Subject 2 had limited experience with Mouse-Dasher. All had vision corrected to normal and were right-handed.

2. Task

   The task was to enter text dictated from Jane Austen's *Emma*. Dasher's alphabet consisted of lower case and capital letters, the space character and a full stop. Subjects were asked to capitalise words correctly, e.g. Mr Knightley.

3. Apparatus

   (a) **Platform**: Pentium IV 2.8GHz running Linux 2.8.3

   (b) **Dasher Display**: 700x491 pixels, font size 12

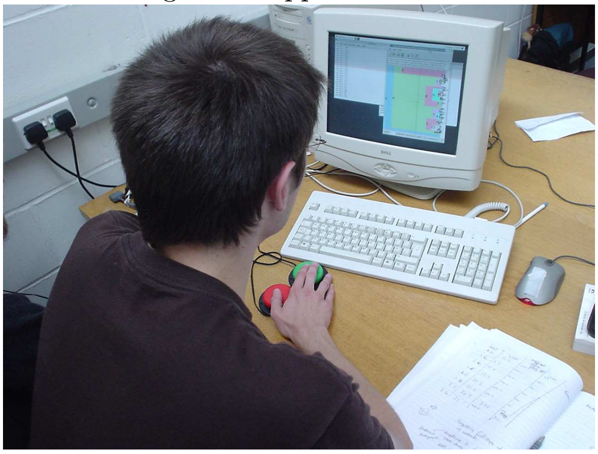
   (c) **Input Device** Two buttons, using a Don Johnston switch interface. The trialists could zoom out ('reverse') by holding either one of these buttons down for half a second.

   (d) **Dictation** The passages were recorded as a series of audio files and cycled through by a demonstrator

   (e) **Language Model** PPMD5 trained on a corpus from Emma, excluding dictation passages.

   The apparatus is pictured in figure 3,

Figure 3: Apparatus used



4. Procedure

   The protocol is similar to that in [5]. The subjects used Dasher in 6 sessions, with two 5 minute periods of writing within each session. Within these 5 minute periods the trialists were read a dictation from Jane Austen's *Emma*, and entered the text they heard using the Dasher interface. The dictation passages were stored as audio files on a computer and cycled through by the dictator as required. Before the first session, trial participants read an information sheet explaining the Dasher concept, and were allowed to experiment with mouse-driven Dasher for 5 minutes, and two button Dasher for a further 5 minutes to familiarise themselves with the controls. The initial speed of Dasher was set to 1.0bps. Before each dictation, the trialists were also allowed to read a copy of the dictation passage, to minimise errors arising from Austen's unusual writing style and spelling. At the end of each of these 5 minute periods, the user was given the option of increasing/decreasing speed by 0.1, 0.2 or 0.3bps. Between each 5 minute writing spell the users had a break of 5 minutes. No more than two sessions took place on a single day and the maximum spacing between any two sessions was two days.

### 4.2 Results

Figure 4 shows the results for the novice subjects. The number of words written is defined to be the number of characters divided by five. The most common form of error was the omission of full stops, which users found difficult to locate, although with practice all users learnt to write very accurately, with all error fractions below 5% after 1 hour.

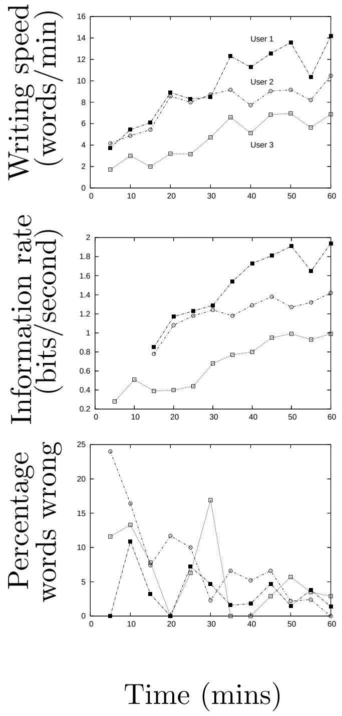



Figure 4: Data from novice subjects

# 5 Expert Users With and Without Automatic Speed Control

Expert users, both with over 5 hours of experience of two button mode, were tested. An automatic speed control (ASC) was developed. As Dasher runs, it stores the times between clicks ($\Delta t's$). The ASC slows the speed whenever Dasher records a $\Delta t$ below some adjustable factor multiplied by the median $\Delta t$.

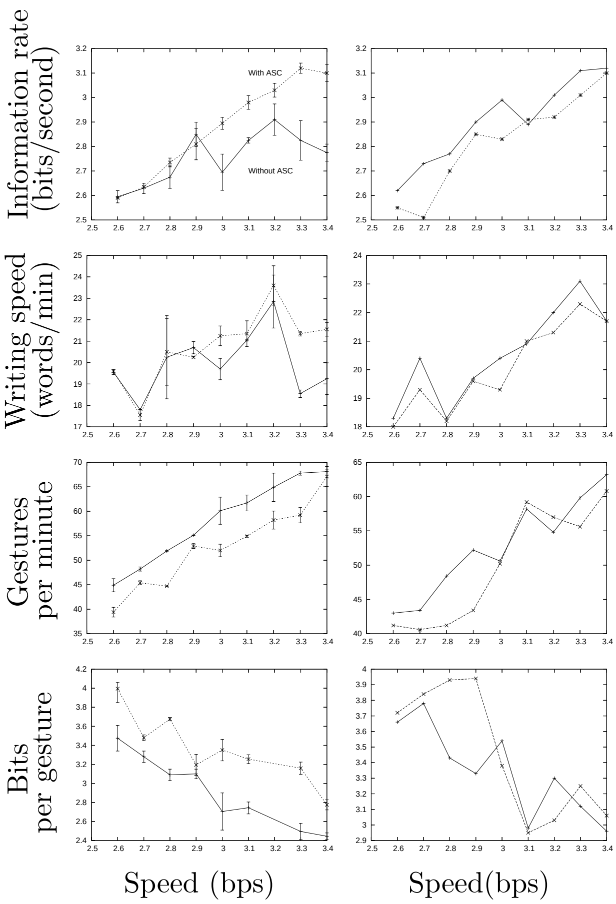
The ASC was developed so as to help a user who is struggling, whilst not 'punishing' a user who is doing well. It was hoped that the ASC would both increase writing speed and reduce the number of gestures made per minute. Rapid clicking is a sign of distress and inefficiency (failing to communicate the maximum number of bits with each gesture). By slowing the speed, the user has time to make the next click accurately and recover. Without the slow down, the rapid clicking produces a 'cascade' of inaccurate clicks which will lead to failure.

## 5.1 Procedure

An expert underwent four 45 minute sessions; two with the ASC and two without. Another expert performed two sessions (one with the ASC and one without). A session consisted of 9 five minute dictations using the same apparatus as before. Speed was incremented by 0.1bps from 2.6 bps to 3.4 bps. The ASC slowed the speed by 90% whenever a $\Delta t$ below 0.5 of the median, or a $\Delta t$ below 0.3s was detected, whichever is greater. The latter condition is imposed because 0.3s was the minimum time between clicks the buttons allowed. Dasher would then accelerate back to its original speed over 1 second. Dasher measured the $\Delta t's$ as it ran, and formed the distribution from which the median was drawn from the first 100 clicks.

### 5.2 Results

Writing data is plotted in figure 5. For expert 1, the averages from the two relevant sessions are plotted, with the errors taken as half the range divided by $\sqrt{2}$. The error percentage is not plotted, as for both experts this was negiligible.




(a) Expert 1 (b) Expert 2

Figure 5: Expert data

Timing data is plotted in figures 6, 7 and 8. The fraction of reverses which were preceded by a $\Delta t$ of less than half the median (without the ASC) is plotted in figure 6 . $\Delta t$ distributions are plotted in figure 7, without the ASC, which has little effect on the overall shape of the distribution. The $\Delta t$ events are placed in bins of equal width, and the graphs show the number of events in each bin. The average number of reverses used at each speed is plotted in figure 8.

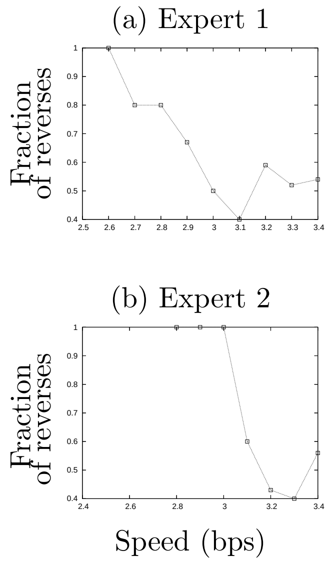



Figure 6: Fraction of reverses without the ASC preceded by a $\Delta t$ less than half of the median.

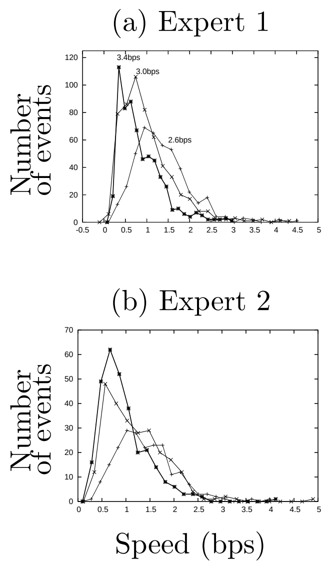



Figure 7: $\Delta t$ distributions at three different speeds

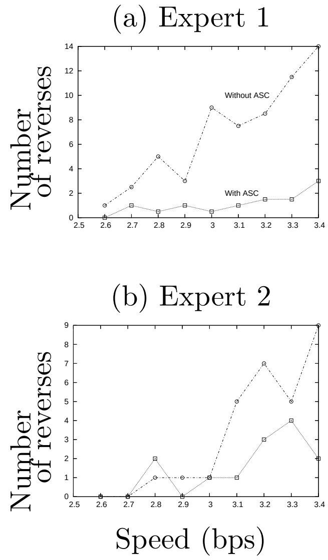



Figure 8: Number of reverses used

### 5.3 Analysis

The useful region of these graphs lies before the ‘optimum’ speed (3.2bps for expert 1, 3.3bps for expert 2), as even writing at the so-called ‘optimum’ can feel uncomfortably fast, and it is likely that most users will choose to operate at speeds lower than this.

The graphs of information rate and words written per minute in figure 5 show peaks. At low speeds, the user is very comfortable, makes no mistakes, and writing speed is limited by the number of bits being fed to the screen. Past the optimum, the user makes many mistakes, and writing speed is limited by the accuracy of button pressing; a small inaccuracy (clicking when the fiducial is not at the desired location) very quickly translates into a large error on screen, which may necessitate the use of the reverse. The information rate error bars are small at low speeds as the reverse was practically never used, and hence the information rate was almost a maximum.

The data in figure 6(a) confirms the hypothesis that a reverse is preceded by a ‘small’ $\Delta t$ for expert 1. For expert 2, figure 6(b) indicates that this hypothesis is supported until a speed of 3.2bps, where it breaks down. The hypothesis is strongly supported for both experts below 3.0bps. The ASC has almost eradicated the use of the reverse (figure 8). This is an important feature of the ASC if two button mode is to be used by real disabled users, at whom it is aimed - for many, the long hold could be a difficult gesture to make, and so any system that cuts down the number of intensive long-hold gestures will be highly desirable. There will be other occasions where a small $\Delta t$ does not precede a reverse - a situation in which, although the user clicks rapidly, she is not in difficulty and the speed is slowed down unecessarily. However, reversing is such a time-consuming manoeveur, that provided the user triggers the reverse a certain number of times, the eradication of the reverse far outweighs any penalty from the speed control being activated unecessarily.

From figure 5, the ASC improves the information rate and bits conveyed per gesture significantly for expert 1. For expert 2 the improvement was less marked, but in the region of interest below the optimum the bits conveyed per gesture improved slightly on using the ASC. Based on the results in figure 5, no firm conclusions can be drawn about whether the number of words that can be written per minute changes using the ASC for either expert 1 or expert 2.

It is clear from the results in figure 5 that the ASC had a more pronounced effect on the performance of expert 1 than of expert 2. The explanation is simple. Firstly, we note that expert 2 used the reverse fewer times than expert 1. The primary gain of the ASC is in cutting out reverses, so if a user never has to make a reverse, the ASC will actually be detrimental to his performance (through slowing the speed unecessarily). The reason the ASC helped expert 1 so much, was because expert 1 used the reverse a lot, as he was a less accurate clicker than expert 2. Secondly, the ASC relies on the fact that reverses are preceded by a 'small' $\Delta t$ - for expert 2, this hypothesis was only upheld (figure 6) at lower speeds, and in this speed region, on observation of figure 5, the ASC worked well in reducing the number of gestures made per minute, and increasing the bits conveyed per gesture relative to the performance without the ASC. Thus we have two factors that must be present for the ASC to improve performance. Firstly, the user must be using the reverse and secondly, the reverses must be preceded by a small $\Delta t$ where 'small' is defined as before.

Performance aside, both users agreed that writing with the ASC felt more comfortable than writing without. In future versions a possible modification that could be made is that the ASC is only triggered when Dasher detects the fraction of reverses preceded by a 'small' $\Delta t$ above 0.6 – this way we can be sure that the user is both using the reverse, and that reverses are preceded by rapid clicking, which are our two key requirements for the ASC to work well.

In these trials, 'small' was always qualified as being less than half the median $\Delta t$. In future experiments, a variety of different factors should be tested to observe which is optimal. It is highly likely that this will vary between users, and Dasher allows the user to set the factor manually, so the system can be customized to the user's desires.

Button-Dasher can be compared with Morse. Taking the full stop as a 6 gesture character, using character probabilities from *Emma* and the number of gestures required to write each character in Morse, the mean number of gestures required per character is 2.1925 (without capitalisation). In the table below, the maximum writing speed of each expert is taken, with and without the ASC, and the number of gestures required to write at this speed using Morse. It is clear that Button-Dasher is very gesture-efficient compared to Morse.

| | Optimum speed (bps) | Average Gestures per min | |
|---|---|---|---|
| | | **Dasher** | **Morse** |
| **Expert 1 without ASC** | 3.2 | 64.9 | 250.5 |
| **Expert 2 without ASC** | 3.3 | 59.8 | 253.2 |
| **Expert 1 with ASC** | 3.2 | 58.2 | 258.7 |
| **Expert 2 with ASC** | 3.3 | 55.6 | 244.5 |

The predictions of equation (11) can be compared with the performance of expert 1. As (11) was derived under the assumption that speed remained constant within a session, it will only be compared with the data collected without the ASC. A maximum likelihood method was used to determine the parameter $\mu$. Expert 1 had $\mu = 55ms$. Since no error bars could be drawn for expert 2, the maximum likelihood method could not be used. The curve was fit via least squares regression and gave $\mu = 45ms$.

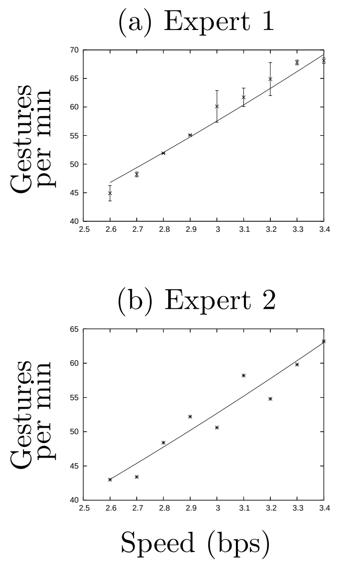



Figure 9: Gestures per minute: theoretical predictions and experimental observations

The model fits the data well and produces 'sensible' predictions for clicking accuracy, but has shortcomings. The assumption is that the user is clicking at his maximum average rate, so the inequality of (11) can be treated as an equality for the purposes of calculating $\mu$. Thus, what has really been calculated is the maximum average value of $\mu$ for the expert subjects. Additionally, it has been assumed that $\mu$ remains constant over the speed range.

# 6 One Button Mode

In [2], a theoretical model of a single-switch user is presented. As in [2], it's assumed that the user controls only the times of presses, not the times of releases and that the user cannot distinguish between short and long presses.

## 6.1 Theoretical Analysis

The user is modelled with two parameters: a timing accuracy $g$, and a recovery time $S$. The user clicks within a time $\pm g/2$ around each intended click time, and then requires a delay of duration $S$ before she can click again.

As in [2], time is divided into boxes of duration $g$. The user's actions are a sequence of choices between two actions, '0' (doing nothing, which takes time $g$), and '1' (pressing the switch, which takes time $S$). As soon as an action has been completed, the user is free to make another choice.

The Capacity is $C$, where

$$p_1 = 2^{-CS} \tag{12}$$

$$p_0 = 2^{-Cg} \tag{13}$$

and

$$p_1 + p_0 = 1. \tag{14}$$

The information rates during a '1' and '0' can be calculated. The information content conveyed by choosing 1 is $\log_2 \frac{1}{p_1}$, and it occupies a duration $S$. So the information rate is $\frac{\log_2 \frac{1}{p_1}}{S}$. Performing a similar calculation for a '0', the information rate is $\frac{\log_2 \frac{1}{p_0}}{g}$; both are equal to $C$ bits per unit time. This indicates that 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. Using the same screen coordinates as before, the Dasher screen is split into two sections of length $p_0$ and $p_1$. An event '1' zooms Dasher into the section of length $p_0$ and an event '0' zooms Dasher into the section of length $p_1$. After every button press the positions of the two sections are reversed.

## 6.2 Preliminary Experimental Results

One button mode was tested on two subjects who were experts with two button and normal Dasher but who had never used the one button mode before. The trials consisted of 12 dictations of 5 minutes each. The apparatus was left unchanged from the previous sections, except there was only one button to press. After each dictation the subject had the option of increasing/decreasing speed by 0.1, 0.2, 0.3bps.

The results for the two subjects are in figure 10. The $\Delta t$ distributions for the subjects are in figure 11. Figure 10 does not include data on the percentage of words with errors and this was often 0% and always below 5% - this is because both users were already very familiar with the Dasher concept, and so many of the most common mistakes in the previous novice trials (such as missing full stops) were not made by them.

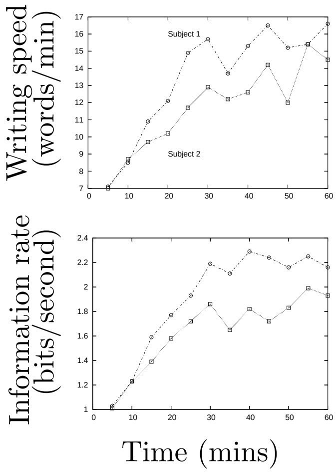



Figure 10: Data from novice subjects

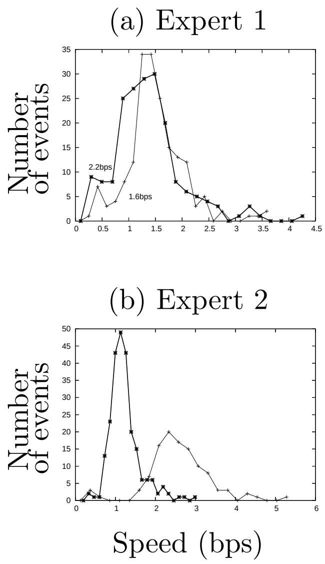



Figure 11: $\Delta t$ distributions for two speeds.medium line 1.6bps, light line 1.0bps

The number of gestures made per minute at a given speed are shown figure 12. In cases where multiple trials were performed at the same speed, an average of the gestures made in each trial is taken.

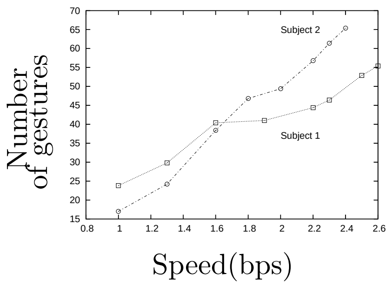



Figure 12: Number of gestures made per minute at different speeds

### 6.3 Analysis

The maximum writing speed was 16.6wpm using 61.4gpm. The mode also remains faithful to the defining feature of the button Dasher modes - the ability to communicate information in a gesture-efficient manner, particularly at lower speeds. However, the gesture-efficiency is lower than for two button mode - in the two button mode, using fewer than 60gpm it is possible to write at almost 25wpm. However, the one button subjects were not experts, and with practice the number of gestures required to write a given quantity of information should fall. In future trials, when expert subjects are available, we hope to do a direct comparison between the one and two button modes.

## 7 Conclusions and Further Work

1. Theory suggests that the gestures per minute required in two button mode rely on the user's ability to click accurately. Accurate clickers will be 'rewarded' with a higher information content per gesture. Dasher is gesture efficient, and designed for those for whom every click is an effort.

2. Three novices, who had used two button mode for 1 hour were able to write at 6.9, 14.2 and 10.5 words per minute respectively. They all wrote with less than 5% of words containg errors.

3. Two experts, with over 5 hours experience of two button mode, were able to write at 23.6 and 23.1 words per minute using 58.2 and 59.8 gestures per minute respectively. A speed control was developed which slowed the speed when the user clicked more rapidly than normal. It made the writing experience more comfortable and allowed one of the experts to write faster. For both experts it improved the number of bits that could be conveyed per gesture.

4. Preliminary results for a one button mode were presented. Novice users could write at a maximum of 16.6 words per minute.

5. The two button mode is now ready to test on real, disabled users. Throughout these trials, able bodied subjects used Dasher, but their behaviour may be different from a disabled subject who is physically restricted to a certain gesture frequency.

6. The one button mode has not been as extensively tested. Tests on expert subjects are required, and after that, tests on disabled users.

## References


[1] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. Available from `http://www.inference.phy.cam.ac.uk/mackay/itila/`.

[2] David J. C. MacKay, Chris J. Ball, and Mick Donegan. Efficient communication with one or two buttons. In Rainer Fischer, Roland Preuss, and Udo von Toussaint, editors, *Proceedings of* Maximum Entropy and Bayesian Methods, volume 735 of *AIP Conference Proceedings*, pages 207–218, Melville, New York, 2004. American Institute of Physics.

[3] Thomas H. Shorrock, David J. C. MacKay, and Chris J. Ball. Efficient communication by breathing. In M. Niranjan J. Winkler and N. Lawrence, editors, *Machine Learning Workshop*, volume 3635 of *LNAI*, pages 88–97, Berlin, 2005. Springer. (in press).

[4] 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*, 17(2-3):199–228, 2002.

[5] D. J. Ward and D. J. C. MacKay. Fast hands-free writing by gaze direction. *Nature*, 418(6900):838, 2002.

[6] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.