

---

# Fast and Flexible Selection with a Single Switch

---

**Tamara Broderick**  
Department of Statistics  
University of California, Berkeley  
Berkeley, CA 94704  
tab@stat.berkeley.edu

**David J.C. MacKay**  
Cavendish Laboratory  
University of Cambridge  
Cambridge, UK CB2 1TP  
mackay@mrao.cam.ac.uk

## 1 Introduction

In single-switch communication, user input consists of repeated clicks, distinguished only by timing information; these clicks might be generated by pressing a button or blinking. For instance, the range of movement of individuals with severe motor impairments may be limited to a single muscle. Alternatively, a crowded or jostled mobile technology user may be able to click precisely while other actions are difficult or sloppy. A single switch may also be useful when information conveyed, such as a PIN, is sensitive and hand location on a normal keyboard might betray this content. Our method, “Nomon” (e.g. Figure 1), expands the application scope of existing methods and facilitates faster writing than the most common single-switch writing interface.

Existing single-switch communication methods include scanning [1, 2] and One-Button Dasher [3, 4]. (Morse Code, in contrast, requires either click duration information or multiple switches.) These methods require options to be arranged in a particular configuration. By contrast, traditional operating systems, web browsers, and free-form applications such as drawing place options at arbitrary points on the screen. We seek a single-switch selection method that is not limited to certain forms of option placement. We want our method to work for any number of options; to be able to effectively reorder the set of selections without imposing additional cognitive load; and to allow the user to attend only to the desired target. Our method, Nomon, accomplishes these objectives. It can further automatically adapt to individuals’ clicking abilities and incorporate prior beliefs about option selection frequency.

To test our method, we developed a writing application, the Nomon Keyboard (Figure 1), and compared its performance with a popular commercial scanning interface, The Grid 2 [5] (Figure 2). We examined study participants’ writing speeds, error rates, and number of clicks made per character as well as the subjective ratings of their experiences. We found that novice users wrote 35% faster with the Nomon interface than with the scanning interface. An experienced user (author TB, with > 10 hours practice) wrote at speeds of 9.3 words per minute with Nomon, using 1.2 clicks per character and making no errors in the final text.

Our work has recently been published [6]. The full technical report describing Nomon is available online at [http://www.inference.phy.cam.ac.uk/nomon/files/nomon\\_tech\\_report.pdf](http://www.inference.phy.cam.ac.uk/nomon/files/nomon_tech_report.pdf). The Nomon Keyboard, as well as a drawing application (Nomon Draw) and instructions for the use of both applications, is available for download at <http://www.inference.phy.cam.ac.uk/nomon/> under the GNU General Public License 3.0.

## 2 A New method

The trademark of a Nomon application is a set of small clocks, one clock associated with each selectable option. Each clock appears alongside its corresponding option on the screen. For instance, in a writing application (the Nomon Keyboard), a clock appears next to each character, word completion, or text editing function (Figure 1). Each Nomon clock features a moving hand and a fixed line at noon. All moving hands rotate at the same, fixed speed but, at any time, are located at a

variety of angles relative to noon. The user tries to click precisely when the moving hand on her desired clock is at noon. She repeats this action until the clock is selected. Selection is signalled by the desired clock being highlighted with a darker color and the entire application flashing a lighter color; there may also be audio feedback. Between clicks (if more than one click is required to select a clock), the clock angular offsets are adjusted by a heuristic to maximize the expected information content of the user’s next click.

|                          |                     |                          |                       |                       |
|--------------------------|---------------------|--------------------------|-----------------------|-----------------------|
| a and_<br>a_<br>as       | b be_<br>by_<br>but | c can_<br>could_         | d do_<br>did_<br>down | e                     |
| f for_<br>from_<br>first | g get_              | h he_<br>have_<br>had    | i in_<br>it_<br>is    | j just_               |
| k know_                  | l like_             | m more_<br>my_<br>me     | n not_<br>no_<br>now  | o of_<br>on_<br>or    |
| p people_                | q                   | r                        | s she_<br>so_<br>said | t the_<br>to_<br>that |
| u up_                    | v very_             | w was_<br>with_<br>which | x                     | y you_<br>your_       |
| z                        | _                   | .                        | Delete                | Undo                  |

Figure 1: The Nomon Keyboard, a writing application (screenshot). Words that are prefixed by the concatenation of the current context and the letter X appear next to the letter X. Underscore represents a space. Options for period, a character-deletion function, and an undo function are also available.

Row-column scanning can be viewed as a special case of the Nomon selection method where clocks are synchronized and the number of clicks to selection is fixed. In fully general Nomon, though, we can effectively completely reorder the set of selections after each click without demanding any extra cognitive load from the user. Moreover, the independent movement of the clock hands in Nomon frees the user to attend only to the desired target, in contrast to the need, in scanning, for the user to attend both to the desired target and the moving highlight. And in Nomon once the target is located visually, the user is free (without suffering a performance penalty) to focus on selecting a single, fixed clock (rather than track the moving scan highlight). Since the clock periods are usually much shorter than a full scanning rotation, there is little penalty for missing a potential click time.

In Nomon, unlike scanning, we assume that the user will not always click perfectly at the desired time. Nomon can learn a user’s probability of clicking at different (typically small) offsets relative to noon. This learning is accomplished via an approximate Parzen window estimator, with contributions from more recent clicks weighted more strongly to allow adaptation to a user whose skill changes over time. We can also specify a prior probability distribution over clocks according to a predictive model of user choices. For instance, in the writing application tested below, our language model assigned prior probabilities to letters and word completions based on the British National Corpus word-frequency list [7]. These prior probabilities could be adaptive and context dependent.

During a particular selection process, the posterior probability of any clock given the clicks thus far can be calculated from Bayes’ theorem. When the probability of a single clock is sufficiently high, we declare it the winner. The probability threshold for winning is an adjustable parameter of the model; it can vary according to context or across clocks. A higher threshold can ensure greater safety for critical actions. Our selection method implies that the time taken to make a selection in Nomon scales logarithmically with the number of clocks if the prior over clocks is uniform.

### 3 Writing application experiment

We developed the Nomon Keyboard (Figure 1) to compare writing with Nomon to writing with a popular commercial scanning interface, The Grid 2 [5] (Figure 2). To that end, sixteen study participants with no previous experience of either interface wrote with both; we also collected data from an experienced user (TB, > 10 hours experience in each). In each of two sessions, a participant used one of the interfaces to write short phrases appearing on screen. A session was divided into

four blocks of about 14 minutes each. During the first three blocks, each participant was allowed to adjust the rotation-period or scanning-delay parameter, respectively, at the end of each written phrase. No changes were allowed during the final block. For each interface, cash prizes of £5 were won by the faster half of participants in the final block beyond the base compensation of £10/session.

|  |   |   |   |   |    |
|--|---|---|---|---|----|
|  | a | b | c | d | e  |
|  | f | g | h | i | j  |
|  | k | l | m | n | o  |
|  | p | q | r | s | t  |
|  | u | v | w | x | y  |
|  | z | _ | . | ← | ■← |

Figure 2: The scanning grid from The Grid 2 used in this comparison study (screenshot). The six long rectangles on the left hold word completions. The remaining options are fixed and include letters, an underscore for space, a period, a character-deletion function, and a word-deletion function.

**Text-entry rate** We calculated text-entry rate in words per minute (left panel of Figure 3), where a word is defined as five consecutive characters in the output text. In the final block, participants wrote faster with Nomon than with The Grid 2 ( $F_{1,15} = 135, p = 6.8 \cdot 10^{-9}$ ). The average writing speed with Nomon was 35% faster; participants wrote at 4.3 words per minute on average with The Grid 2 and 5.8 words per minute with the Nomon Keyboard. While the alphabetic layout was easy for novices to use, a computer simulating writing from a conversational corpus with no errors has been shown to achieve a 19% faster writing speed with a frequency-ordered layout than with an alphabetic layout [2]. Even if we artificially inflate the novice writing speeds using The Grid 2 by 19%, novices remain faster at writing with Nomon ( $F_{1,15} = 19.14, p = 5.4 \cdot 10^{-4}$ ).

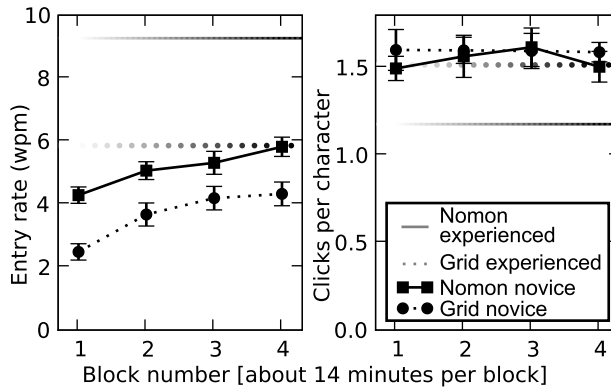


Figure 3: Mean entry rate (*left*) and click load (*right*) across interface blocks. Mean entry rate is measured in words per minute, and click load is measured in clicks per output character. In both panels, error bars represent 95% confidence intervals for the novice user means, and the average experienced user (TB) performance is illustrated by horizontal lines for comparison.

**Error rate** The error rate during a block is the character-level Levenshtein distance [8] (a.k.a. edit distance) between target phrases and the text written by the participant divided by the number of characters in the target phrases. The average novice character-level error rate (over all blocks) for the Nomon Keyboard was 0.43%, and the average novice error rate for The Grid 2 was 0.34%. There was no significant difference in novice error rate between the two interfaces ( $F_{1,15} = 0.71, p = 0.41$ ). The experienced user made no errors while using Nomon and made one error while using The Grid 2, for a mean scanning block error rate of 0.06%.

**Click load** The click load (right panel of Figure 3; a.k.a. “keystrokes per character” [9] or “gestures per character” [4]) is the number of clicks per output-text character (cpc). The average novice rate

(over all blocks) was 1.58 cpc for the Nomon Keyboard and 1.55 cpc for The Grid 2. There was no significant difference in novice click load between the two interfaces ( $F_{1,15} = 0.49, p = 0.49$ ). The experienced user required, on average, 1.51 cpc for The Grid 2 and 1.18 cpc for the Nomon Keyboard. For comparison, error-free writing on a normal keyboard requires 1 click per character.

**Subjective ratings** At the end of each session, we gave novice participants a questionnaire that asked them to quantify how much they agreed with written statements, largely the same as in [10]. Participants agreed more strongly with the following statements for the Nomon interface: “I liked writing using X;” “It was easy to select word completions;” and “It was easy to correct errors.” Unique to an “Open Comments” section was the sentiment that Nomon looks unusual at first but is worth getting to know. One participant remarked, “Surprisingly, I found this more user-friendly.” Another noted, “The writing system looks intimidating when it first comes up on screen but is actually very easy to use.”

## 4 Discussion

Nomon benefits in this comparison from its nice scaling properties and clock-position flexibility. While a Nomon writing application allows a straightforward comparison of Nomon with existing single-switch communication methods, the Nomon selection method is not limited to writing. For example, Nomon can be used for internet browsing by placing a Nomon clock next to each link. Or Nomon can be used for drawing by placing a dense grid of, say, hundreds of clocks on a canvas. (The Nomon Draw application works in this way.) A user can draw a line by selecting its endpoints directly from the canvas.

## Acknowledgments

This research was made possible by unrestricted donations from The Nine Tuna Foundation and Nokia. Sensory Software generously donated a 60-day free trial of the scanning software. TB’s research was supported by a Marshall Scholarship. We are grateful to Per Ola Kristensson, Geoffrey Hinton, Keith Vertanen, Philipp Hennig, Carl Scheffler, and Philip Sterne for helpful discussions.

## References

- [1] R.I. Damper. Text composition by the physically disabled: A rate prediction model for scanning input. *Applied Ergonomics*, 15:289–296, 1984.
- [2] H. S. Venkatagiri. Efficient keyboard layouts for sequential access in augmentative and alternative communication. *Augmentative and Alternative Communication*, 15(2):126–134, 1999.
- [3] 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, *Maximum Entropy and Bayesian Methods*, volume 735 of *AIP Conference Proceedings*, pages 207–218, Melville, NY, USA, 2004. American Institute of Physics.
- [4] David J. Ward, Alan F. Blackwell, and David J. C. MacKay. Dasher—a data entry interface using continuous gestures and language models. In *UIST ’00: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pages 129–137, New York, NY, USA, 2000. ACM.
- [5] Sensory Software International Ltd. *The Grid 2: Version 2.4*. Sensory Software International Ltd., December 2008.
- [6] Tamara Broderick and David J. C. MacKay. Fast and flexible selection with a single switch. *PLoS ONE*, 4:e7481, October 2009.
- [7] Adam Kilgarriff. BNC database and word frequency lists, 1998. Available at <http://www.kilgarriff.co.uk/bnc-readme.html>. Accessed on 18 March, 2009.
- [8] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173, 1974.
- [9] I. Scott MacKenzie. KSPC (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of the Fourth International Symposium on Human Computer Interaction with Mobile Devices*, pages 195–210, Heidelberg, Germany, 2002. Springer-Verlag.
- [10] Per Ola Kristensson and Leif C. Denby. Text entry performance of state of the art unconstrained handwriting recognition: a longitudinal user study. In *CHI ’09: Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 567–570, New York, NY, USA, 2009. ACM.