Stepped Quartering compared with Alternative Code. Short version.

Joris Verrips

Stepped Quartering with a single switch is an input technique intended for non-speaking people who depend on switches to communicate with alphabetical systems. It employs pauses of variable length and combines well with abbreviation expansion and with word prediction. To test an able-bodied test subject practised with it and repeatedly copied English poetry with and without word prediction. The same test was done with a single switch and Alternative Code. This variant of Morse Code requires from users the ability to effect short clicks and *long clicks, has pauses of minimal length, and appears faster yet.*

Introduction

People who access their computer with a single switch need efficient techniques to do so. This paper explains and compares two such techniques, both of them relatively new, and also documents how one of them can be used to have conversations when combined with synthetic speech, character macros and word prediction.

Stepped Quartering

Figures 1 to 5 illustrate Stepped Quartering[1].

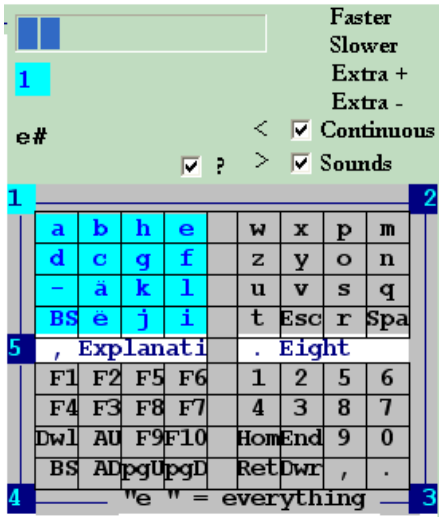

Figure 1. A scanmatrix with the alphabet and thirty-eight other characters, among them Spa for Space.

The character 'e' was recently selected and is marked on the panel above the scanmatrix as 'e#'. It is also shown on the last line of the scanmatrix as part of an abbreviation expansion, “e ” = everything. The first quarter is coloured to *suggest* it, when a click is given the first quarter is *selected.* This situation is shown in Figure 2.

1 For a low quality video visit www.depratendecomputer.nl/somevideos.htm.

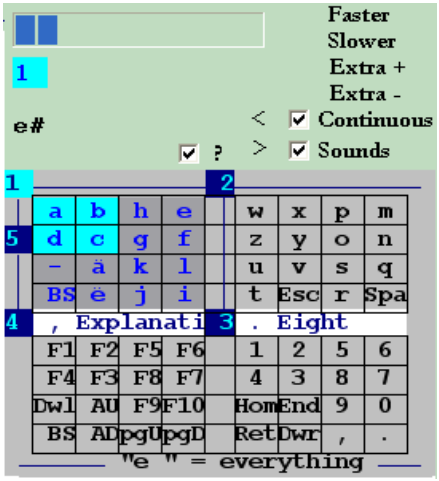

Figure 2. Stepped Quartering. The first quarter is selected and is therefore marked.

In Figure 2, the first quarter *of* the first quarter is coloured, to suggest it. When another click is given, we see Figure 3.

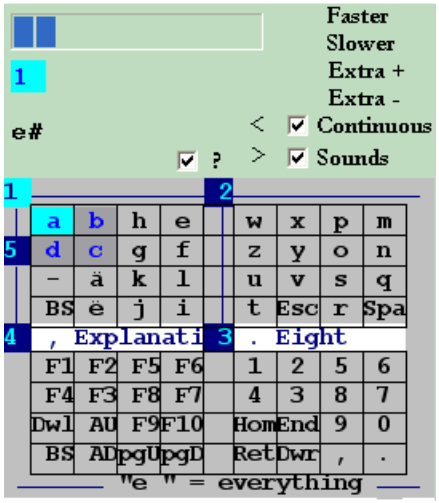

Figure 3. Stepped Quartering, the first quarter of the first quarter is now marked.

In Figure 3 the first quarter of the first quarter of the first quarter is coloured and contains just one character. When a click is given this character 'a' is selected and word prediction will be based on 'ea', as is shown in Figure 4. When *no* click is given, 'b' will be suggested. When we wait a bit, and select the fifth option indicated as 5, Figure 1 will reappear *without* selection of a character. In Figure 3 the fifth option does not indicate a quarter but corrects.

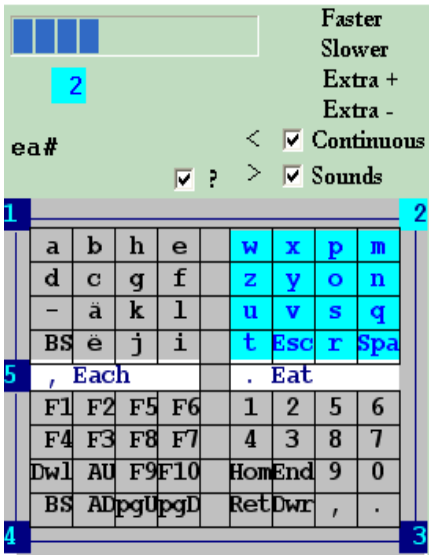

Figure 4. Word prediction is now based on 'ea'. The software stepped 1-2 and a click will now select the second quarter.

Figure 5 shows a list of predictions, a two-word character macro and correct, placed between 1 and 2.

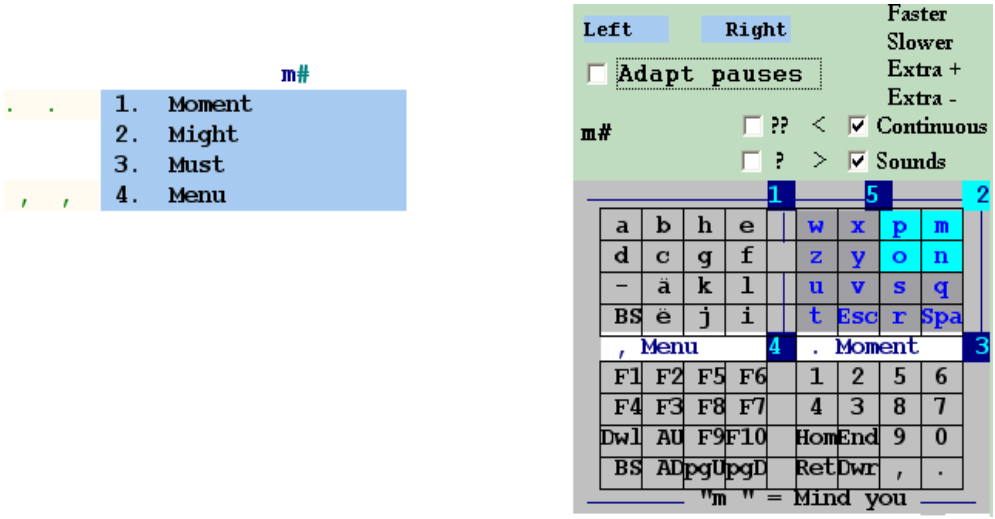

Figure 5. If - means a step, with 2-3 click click Spa will be selected, this means Space and will expand into the macro “m ” = “mind you ”.

Alternative Code

If shorter and longer clicks can be entered volitionally and reliably Alternative Code probably is the fastest system with a single switch. Figure 6 shows that frequent characters have short codes and that most consecutive characters have consecutive codes. A (preceded) Long Hold like ..._ may access word prediction and both character macros and word macros are easily accessed due to the short code for Space. Also, once learned, no visual attention is needed, and when combined with special input devices[2] each and every other program is accessible, though perhaps not at the same speed as is possible with eye-gaze or with a head-mouse[3].

2 Such as the Tandem from www.tandem.com or the DarciToo from www.westest.com.
3 With a head-mouse a small mirror is attached to the forehead. It is a cheap alternative to input by eye-gaze.

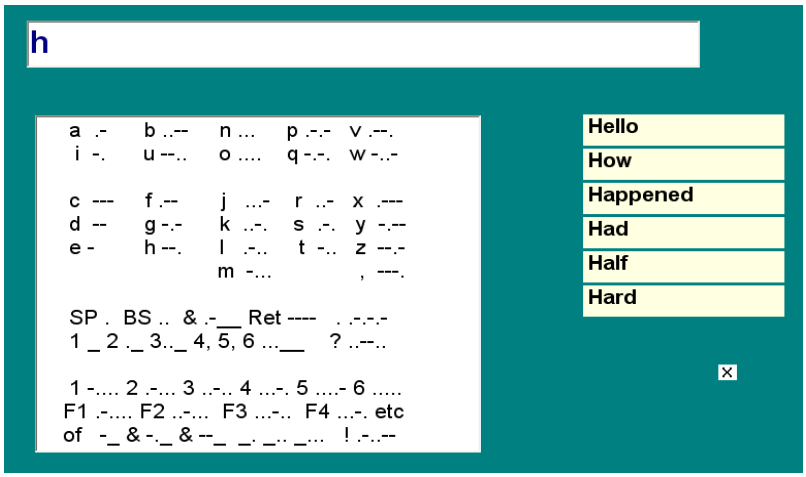

Figure 6. Screen image of Alternative Code with word prediction.

Hypotheses

Three hypotheses were formulated:

Protracted training combined with gradual shortening of the pauses will lead to faster input with Stepped Quartering.

Word prediction might speed up a copy task with Stepped Quartering, and more so with a repeated copy task.

An input technique that requires fewer clicks, fewer pauses and less visual attention, as is true for Alternative Code, must be faster.

Experiments

Before testing began the test subject signed an agreement that his input would be registered and used for research purposes. He was allowed to play with Figure 5, to make him feel at ease. His pause time was set first at 500 milliseconds and after half an hour of practice at 400 milliseconds per step (-) with 400 milliseconds extra pause (_). He had to correct all errors and was asked to copy some English phrases[4] to practice, this took about seventy minutes hands on, partly to learn to handle the single switch. Next, three experiments were done. First the test subject copied a well known and short poem[5] twice with (A and B) and once without (C) word prediction and from paper. See Figure 7 and Table 1 for results with a single test subject.

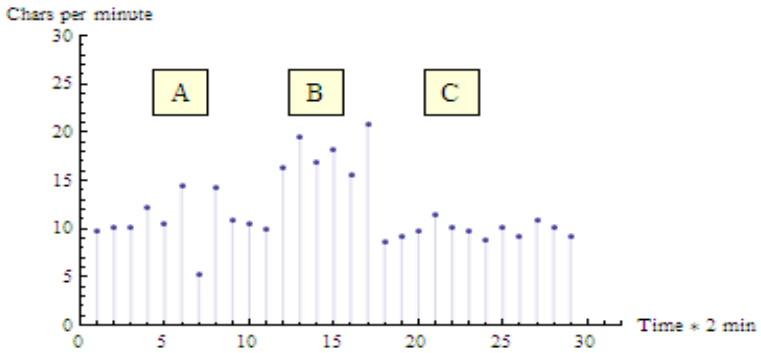

Figure 7. Input rates of the test subject when repeatedly copying English poetry, twice with (A and B) then once without (C) word prediction, pause time 400 milliseconds.

4 Phrases were taken from MacKenzie, 2009.

5 'Days' by Philip Larkin, 1964.

A Average input rate ± SD	B Average input rate ± SD	C Average input rate ± SD
10.7 ± 2.5	17.9 ± 2.0	9.8 ± 0.8
A Steps per character ± SD	B Steps per character ± SD	C Steps per character ± SD
2.9 ± 0.9	1.7 ± 0.2	3.2 ± 0.37

Table 1. Average input rate and Standard Deviation as well as steps per character, all of them with a stepping time (or pause time) of 400 milliseconds and an extra pause _ of 400 milliseconds that was counted as a step.

Picture 1. FV engaged in a copy task.

Next, the test subject copied texts from MacKenzie 2009 with gradual and automatical adaptation of the pauses to 250 milliseconds and word prediction, experiment D. This went smoothly and took about forty-five minutes at what time little further speed enhancement was perceptible. Input rates are in Figure 8.

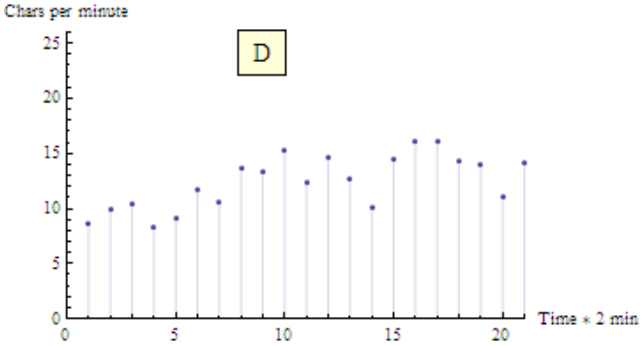

Figure 8. Results from copy task with gradual adaptation of pauses and with word prediction.

The first three experiments were then repeated, with a pause time of 250 milliseconds and extra pauses of 250 milliseconds. We call these E, F and G and expect higher rates, due to shorter stepping time (or pause time) and more practice. See Figure 9 and Table 2.

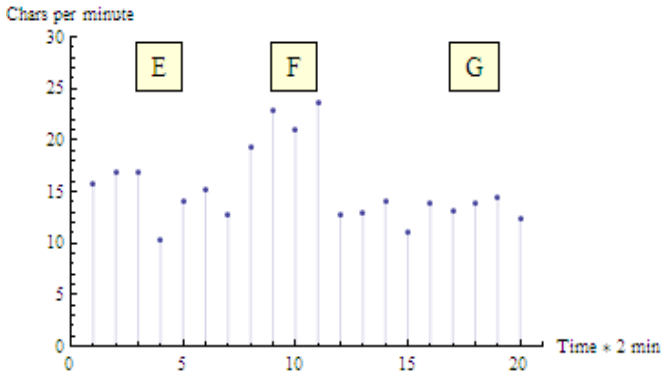

Figure 9. Copy task of English poetry after more training, twice with (E and F) and once without (G) word prediction.

D Prose, copy task with word prediction and Stepping time gradually diminishing from 400 ms to 250 ms	E Poetry, 1st copy task, with word prediction, Stepping time 250 ms	F Poetry, 2nd copy task, with word prediction, Stepping time 250 ms	G Poetry, 3rd copy task, without word prediction, Stepping time 250 ms
12.42 ± 2.4, ??	14.56 ± 2.4, 0	21.7± 1.9, 0	13.2 ± 1.1, 4

Table 2. Average input rate ± SD, BackSpaces given in D to G with Stepped Quartering.

The user was interviewed after each test and consistently indicated that he found Stepped Quartering fun to do. To compare with Alternative Code, phases just like D to G and named H, I, J, K were done with word prediction except in K and without abbreviation expansion. Pause time was gradually diminished from 400 to 250 milliseconds in phase H and sustained dashes could be followed by clicks to move the pointer downwards. After 50 * 2 minutes the input rate still appeared to increase but phase H was ended nevertheless and followed by phases I to K. See Figure 10 and Table 3.

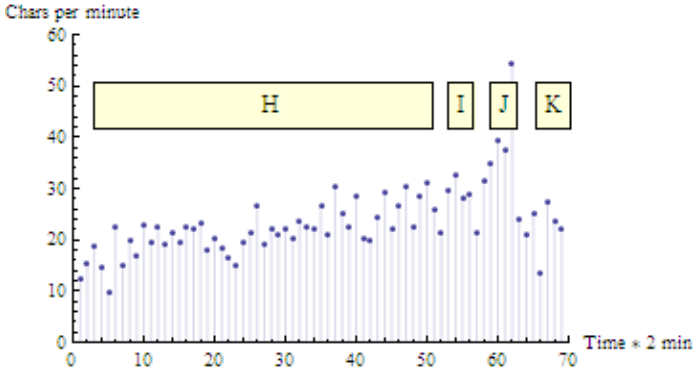

Figure 10. Phases H to K with Alternative Code and word prediction.

H Prose, copy task with pause time diminishing from 400 to 250 milliseconds and with word prediction	I Poetry, 1st copy task, with word prediction	J Poetry, 2nd copy task, with word prediction	K Poetry, 3rd copy task, without word prediction
23.4 ± 6.8	28.9 ± 4.9	43.8 ± 9.1	22.4 ± 4.4

Table 3. Average input rate ± SD with Alternative Code, phases H to K, and pause time of 250 milliseconds. No BackSpaces are documented.

The last 27 datapoints of phase H have as average 24.9 cpm with SD 3.9, almost fast enough for a simple conversation. Due to a technical error with many BackSpaces as a

result, data in phase K are at least 15% lower than should be expected, therefore no comparison between H, I or J and K is possible based on these data. The test subject intensely used word prediction from the start. When interviewed he said that Alternative Code is less fun, more efficient, more error prone and faster than Stepped Quartering.

After the experiments were finished we also tried to have conversations with two speaking computers, to ensure a symmetrical rate problem. We both used Alternative Code, character macros, word prediction and a single switch. We spent about one hour hands-on, did some exercises and had open conversations. Logfiles and simple forms were used to document our experiences. The input rate of FV was around 20 cpm. The input rate of the author, who trained a lot, was above 40 cpm. To communicate was possible, although it was not *easy*.

Interpretation

Differences of input rate between A and B as well as between E and F and I and J were expected and document a property of the word predictor that may be useful during conversation. With this able-bodied test subject, Stepped Quartering was easy to learn and appeared to become faster with practice and with diminishing pause time. Alternative Code was faster, but requires either two switches or the user being able to effect a short click and a long click and, for rapid access to word prediction, to give a Long Hold as well. No statistical tests were done except for a Sign-test on the last twenty datapoints of D and H, with $p<0.01$.

Conclusion

Stepped Quartering is an input technique with a single switch that combines well with word prediction and with abbreviation expansion. It was compared with Alternative Code that appears faster yet but that, if used with a single switch, requires the ability to give short clicks *and* long clicks.

Bias Some researcher bias is to be expected because the author investigated his own software.

Download A complete version of this text including references, footnotes, details and Appendices is on www.depratendecomputer.nl/steppedquarteringandalternativecode.pdf.

Illustrations Figures were either made with Wolfram's Mathematica or are screen images of relevant software. Picture 1 was taken by the author.

Thanks to Frank Voorhuis for his many efforts and for his many suggestions.

Video see www.depratendecomputer.nl/videos/steppedqandac.wmv and www.depratendecomputer.nl/somevideos.htm

All comments are welcome. Mail: j.verrips@planet.nl.