Sustained dashes

Due to diverse neuro-muscular diseases, some people have no intelligible speech but can read and spell. Such handicaps vary from dysarthria after speaking for a while, alleviated by writing messages, to communicating 'yes' and 'no' by eye-blinks or grunting with support from specialized care givers. If speaking, writing, typing, mousing and pointing are impossible, switches may slowly select character after character, either by a form of scanning or by encoding. Sustained dashes, where a switch is pressed for a longer time, allow active scanning with a single switch and allow combining Morse Code with word prediction. This technique was discovered by the author and was described, analyzed and tested.

Scanning

The input technique used most often with switches is row column-scanning, shown in Figure 1 with the alphabet. A (semi-)professional interpreter records eye-blinks, a moving finger and the like. Naming first colors then characters allows text construction at about *ten* characters per minute. This is frustratingly slow compared to normal speech of about *six hundred* characters per minute. Other techniques use a perspex frame and eye-gaze, Morse Code, switches coupled to software and computer based eye-gaze systems. As a rule, only some techniques are possible with the individual patient due to diverse physical problems such as poor eyesight, nystagmus or athetosis.

A	B	C	D	End of word	
E	F	G	H	End of sentence	
I	J	K	L	M	N
O	P	Q	R	S	T
U	V	W	X	Y	Z

Figure 1. 'Red Yellow click E F G click' will select 'G'.

In active scanning the user clicks to move the focus, usually a row or a column at a time, and waits for the machine to accept an item. It normally requires two switches, let us say one called 'dit' for horizontal and one called 'dash' for vertical movement, and it is also known as 'inverse scanning'. In passive scanning the machine moves the focus and the user accepts (accords), first a row then an item. As clicks take less time than pauses, active scanning must be the faster method, independent of the type of switch used. More variants of scanning exist, like 'stepped scanning' that combines active and passive scanning and grouped or blocked scanning with a shrinking focus. All forms of scanning require much (visual) attention and are frustratingly slow. This makes having a conversation with scanning almost impossible.

In the scanning matrix, a row or a column can be reserved for word prediction and for various other functions. Frequent occurring items can be placed in the upper left corner, to reduce the total number of clicks and pauses. This comes at a price, because users must remember or search for the exact location of items, and most forms of scanning are slow *and* error prone. Errors can be quite annoying, and occur more frequently with (infrequent) items far removed from the origin. Figure 2 shows a scanning matrix designed to require little attention with an empty column where the focus may rest, a small number of items that was ordered alphabetically, and word prediction accessed by the numbers on the first row. After seven dits or seven dashes the focus (the blueish spot in Figure 2) reappears in the upper left corner. Therefore, both row and column errors are easily corrected. Topologically speaking the matrix behaves like a torus, and it can even move instead of the focus, possibly advantageous for people with lateral gaze paralysis. With active scanning it appears relatively fast, error tolerant *and* easy to learn.

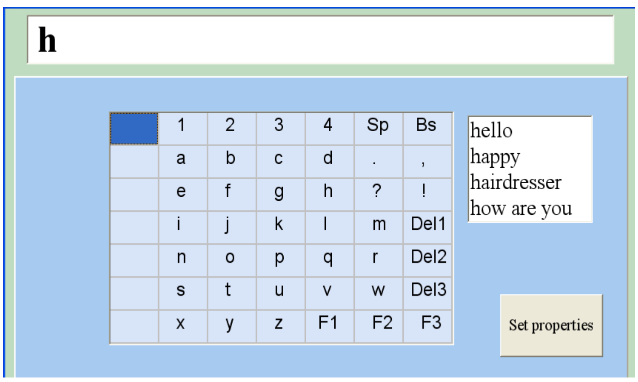

Figure 2. Scanning matrix with empty first column, where the focus may pause. 'h' was selected, perhaps by two dashes followed by four dits and a pause.

In Figure 2, due to word prediction, dit pause = 'hello ' and four dits pause = 'how are you '. With a single switch a short press or '.' is called dit and a long press or '-' is called dah or dash. If dit = move horizontally, and (sustained) dash = (repeatedly) move vertically, we can have two dimensional active scanning with a single switch, possibly accompanied by auditory cues. With two switches, sustained dashes may reverse horizontal scanning direction[1]. They might also implement a jump option as from the first to the fifth row[2] or might change the content of the matrix. For athetoid users, and due to the empty column, the pause can be replaced by a 'prior' switch, to the effect that in Figure 2 'a' would be accepted by 'dash dit dash' and '2', meaning 'happy ', by 'dit dit dash'. This role might also be assigned to a sustained dash.

As to the scanning matrix itself, all kinds of alternatives are possible including dynamical frequency optimized variants and families of linked grids with stored text that users may adapt themselves. Figure 3 shows a simple matrix without word prediction.

	Sp	Bs	a	b	c	d
	e	f	g	h	.	,
	i	j	k	l	m	Del1
	n	o	p	q	r	Del2
	s	t	u	v	w	Del3
	x	y	z	F1	F2	F3
	1	2	3	4	?	!

Figure 3. Without word prediction, fewer clicks are needed per character.

Morse Code

As regards the number of clicks and pauses per character, related to input speed after training, no conceivable scanning method beats Morse Code. Well known for its use in telegraphy and in radio telegraphy, and defined in 1838 by Vail, it can be entered with either one, two or three[3] switches. If a trained interpreter is available it can also be used without any further technology, using eye-blinks or a moving foot and the like. Originally there were no codes for SP

1 And would make Backspace and Space easy to reach, Sp and Bs in Figure 2.

2 However, this is not necessary. If the 'jump' option is set, 'a' has to be selected by vertical horizontal: horizontal then vertical will select 'n', twice horizontal then vertical 'o', 't' and so forth.

3 The third switch replaces the pauses that every code ends with.

(space), BS (backspace) and numerals. When at a later date they were added only codes of length four and higher were available so '..--' was assigned to SP. We may change that, see Table 1.

a .-	f ..-.	k -.-	p .--.	u ..-
b -...	g --.	l .-..	q --.-	v ...-
c -.-.	h	m ..--	r .-.	w .--
d -..	i --	n -.	s ...	x -..-
e .	j .---	o ---	t -	y -.--
BS ----	SP ..	, ---.	1 .----	z --..

Table 1. Morse Code, with 'm' changed from '--' to '..--', SP (space) changed from '..--' to '..' and 'i' from '..' to '--'. In the distant past, a long pause was counted as a space, therefore no code was needed for SP.

In Table 1, fifteen different characters have a code of length four. If '.-.-' is bound to Return all sixteen codes of length four are used.

<table>
<tr><td colspan="8">E .</td><td colspan="8">T -</td></tr>
<tr><td colspan="4">SP ..</td><td colspan="4">A .-</td><td colspan="4">N -.</td><td colspan="4">I --</td></tr>
<tr><td colspan="2">S ...</td><td colspan="2">U ..-</td><td colspan="2">R .-.</td><td colspan="2">W .--</td><td colspan="2">D -..</td><td colspan="2">K -.-</td><td colspan="2">G --.</td><td colspan="2">O ---</td></tr>
<tr><td>H
....</td><td>V
...-</td><td>F
..-.</td><td>M
..--</td><td>L
.-..</td><td>Ret
.-.-</td><td>P
.--.</td><td>J
.---</td><td>B
-...</td><td>X
-..-</td><td>C
-.-.</td><td>Y
-.--</td><td>Z
--..</td><td>Q
--.-</td><td>,
---.</td><td>Bs
----</td></tr>
</table>

Table 2. Morse Code, tree-like display of thirty codes of length one to four.

Other keys such as numerals require at least five clicks and traditionally 1=.---- 2= ..--- 3=...-- 4=....- 5=..... 6=-.... 7=--... 8=---.. 9=----. and 0=----- (five dashes). Many other codes may be defined and used for all other keys, often with the help of mnemonics such as PU or .--...- for Page Up (p=.--. and u=..-). Morse Code is compact, error prone, takes time to learn well and can replace both keyboard and mouse. To see how compact it is compared to RC-scanning consider that codes for V(...-), F(..-.), L (.-..) and B(-...) all would select G in Figure 3. Although it is fast, it requires significant attention[4] and is not appreciated by everybody.

Sustained dashes with Morse Code

4 Users might object that it can be performed with eyes closed and becomes fully subconscious on the long run. Beginning users will close their eyes to concentrate on error-free input.

Sustained dashes, or 'Long Hold', allow to do more with switches than either press them or not. They may combine Morse Code with word prediction in situations where ample feedback can be given. When generating a dash, the user just holds down the switch. After a specified interval, the system will treat the dash as a 'sustained dash' and communicates this to the user acoustically, visually or otherwise. When a single switch is used, first the machine will read dit, or '.', then dash, or '-', and then, when the switch is not released, a sustained dash of variable length. The meaning of this sustained dash depends on how long it is held and on its context. It may be used to access word prediction, to select function keys, commands or other listed items. But sustained dashes may also access different 'modes', such as a mouse mode to allow control of the mouse with short codes or with sustained dashes. And they might mean a host of other things in mobile phones and the like. See Figure 4 and Table 3.

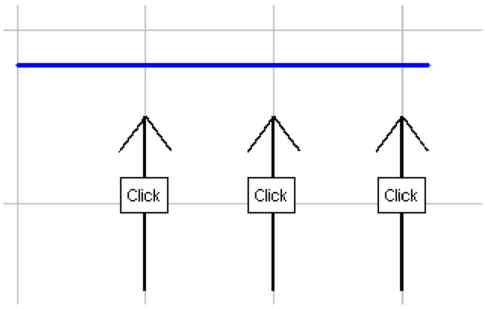

Figure 4. Sustained dash with three audible clicks, possibly combined with visual or tactile cues. Time runs from left to right, where the dash is released then interpreted.

.	a short click means '.' or dit.
—	a somewhat longer click, with *one* audible signal, means '-' or dash.
——	a longer click, with *two* audible signals, means '1'.This is the shortest *sustained* dash.
———	a long click, with *three* audible signals, means '2'.
————	a very long click, with *four* audible signals means '3'.
.———	a short click followed by a long click may mean a series of several backspaces, again with appropriate feedback.
..———	two short clicks followed by a sustained dash may define a function key or may mean different numbers, such as '6' downto '1'.

Table 3. Interpretation of sustained dashes with a single switch. With two switches, a sustained dash of length one needs only one audible signal, and so forth.

Sustained dashes need *not* necessarily be followed by a pause and may be preceded by every other code to change their meaning[5]. BackSpace, to correct, can be selected by a 'preceded' sustained dash, a special code, a sustained dash outside word prediction. Also, with two switches, if first right is chosen and then *simultaneously* left is clicked. This deletes the current code and on repetition BackSpace is interpreted. Other applications of sustained dashes must be possible, such as two directional scanning (as if one might use two mouse wheels simultaneously). This is not covered here.

If integrated in a speaking communication aid, codes may be displayed on screen to facilitate learning, as illustrated in Figure 5. On the right side of the screen, a list with word predictions is displayed and the third word 'Apples ' is marked due to an (active) sustained dash. When the switch is released the dash is no longer active and is interpreted. The word 'Apples ' will be spoken and will be added to the edit line, as shown in Figure 6. If the switch is not released, 'Apprehensive ' will be suggested, possibly with an audible click.

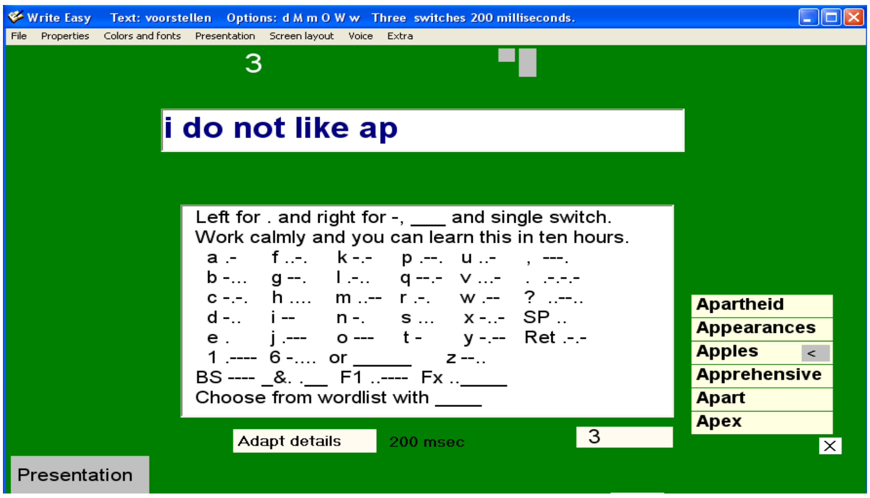

Figure 5. After the characters 'ap' are clicked in the right switch is held for over a second. After 'Apartheid ' and 'Appearances ', 'Apples ' is now suggested.

5 Something to consider carefully because it makes a compact code even more compact.

Figure 6. Effect of 'release switch' in Figure 5.

Details

Three switches allow discrete dashes. One may define 1=. 2=- 3=end of code so '123' will mean 'a'. If we give '1233' the second 3 may move the highlight in the selection list. Now 1 may accept the highlighted item without a timed response. So if 'ap' is chosen by '12312213' (or .- .--.) then '33331' selects 'Apprehensive ' in Figure 5 with thirteen clicks and no pauses. If coordination in time is possible, 3 may just move the highlight so '12<pause>12213333<pause>' may select 'Apprehensive ' with ten clicks and two pauses.

With both Figure 2 and Figure 5 the pause time is important for input rate after sufficient learning. This is the time taken by the machine to wait on the user after the last click, and before either a code is interpreted or the focus 'reads' a character in the scanning matrix. If one clicks '…' in Figure 5 the machine will wait a moment before '...'=s is interpreted, to ensure one does not enter another click with the intent of either '….'=h or '...-'=v. This is related to the so-called speed-accuracy tradeoff that complicates to compare different techniques reliably. It is also related to subtle matters like how much faster active scanning must be compared to passive scanning[6] and to what it is that users do. More than click and wait as is shown by the following exercise. If a click takes two hundred milliseconds, a pause takes three hundred milliseconds and one enters text with an average of four clicks per character at thirty characters per minute, that is an average of two seconds per character, we kind of understand eleven hundred milliseconds per character, only 55% of the time spent.

Text prediction comes in many flavors and much research has been done on the subject including how to make a list of predictions syntactically correct and how to make them 'bright'. Generally speaking it is close to impossible to make it work for the average user in the average conversation in the sense that it rarely speeds up. Though less flexible than word prediction, abbreviation expansion (or macros) has the advantage that it can be learned by heart and does not require to scan a display to decide which -if any- alternative to select. In both Figure 2 and Figure 5 recency information as well as abbreviation expansion is integrated with word prediction. When the next word has started, an abbreviation is presented on top of the list and recently used words just thereafter.

Many factors must influence learning, including instruction, if users like to look up things

6 In my personal experience the difference is impressive but the literature on the subject is not conclusive.

on the display or rather learn them first, what additional (cognitive) problems they may have, details of the switches, if the pause time is gradually diminished and other details of the tasks and of the contexts of use. With all else equal, one might expect that RC-scanning is learned faster than Morse Code but must have a lower input rate after training. Sustained dashes combined with abbreviation expansion might be used in repetitive conversations but not or rarely in ordinary copy tasks.

Test to determine learning curves

Figure 2 and Figure 5 were tested by a healthy test subject with previously established rapid learning when he encoded Morse Code on paper for about one hour, eight months earlier. Pause times were 340 milliseconds in both systems and word prediction was active from the start though it was only used inadvertently. Sustained dashes were demonstrated though no use was expected. The subject played with both systems for about ten minutes to get used to the switches. The same texts were entered twice[7], alternating from Morse Code to RC-scanning after about one hour. As learning Morse Code takes many hours, one might expect to find very different 'learning curves'. See Figure 7 and Figure 8 for the rather unexpected results with two switches based on logfiles made by the software.

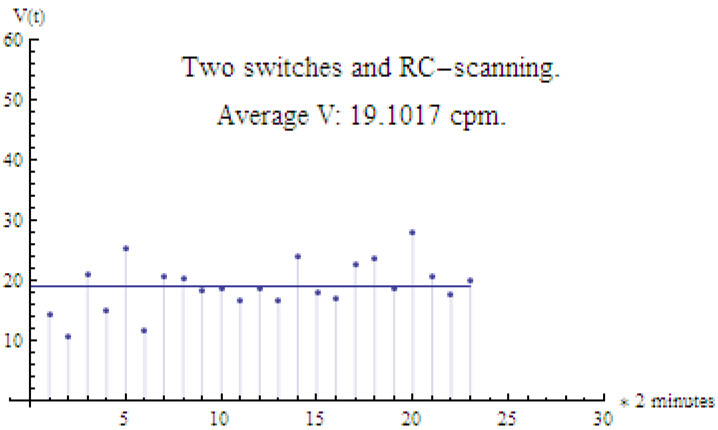

Figure 7. Input rate with Figure 2 in a copy task of less than one hour. SD was 4.45.

7 From the set used by several other researchers interested in creating text with switches or with eye-gaze. It can be downloaded from Scott McKenzie's homepage http://www.yorku.ca/mack/PhraseSets.zip and consists of English texts taken from popular culture. These lines were previously read over slowly to facilitate spelling.

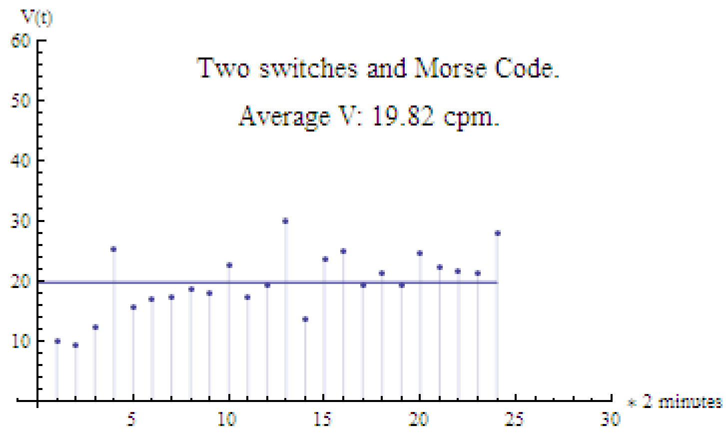

Figure 8. Input rate with Figure 5 in a copy task of less than one hour. SD was 5.17

Test to observe sustained dashes in conversations

Another speaking friend learned Morse Code and exercised with diverse copy tasks and open conversations for a total time of at least twenty six hours and in the course of one year. Compared to our normal conversations, what is said seemed less reliable, often a bit awkward, and extremely slow at about six words per minute. Data were logged by the software, see Figure 9 and Figure 10.

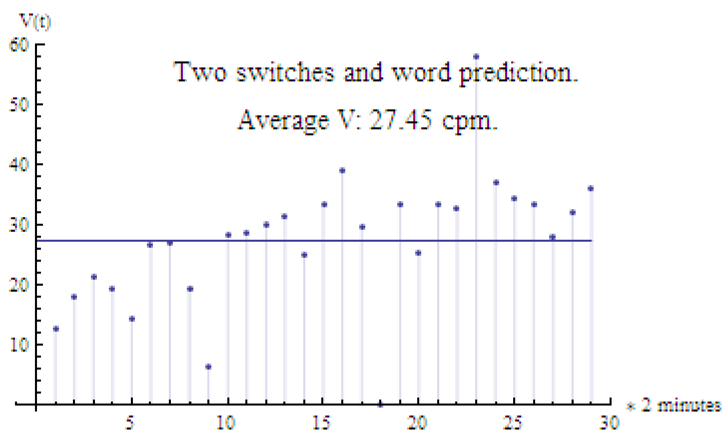

Figure 9. Several conversations by the test subject after about twenty six hours of practice, with Figure 5, two switches, pauses of 300 milliseconds and sustained dashes.

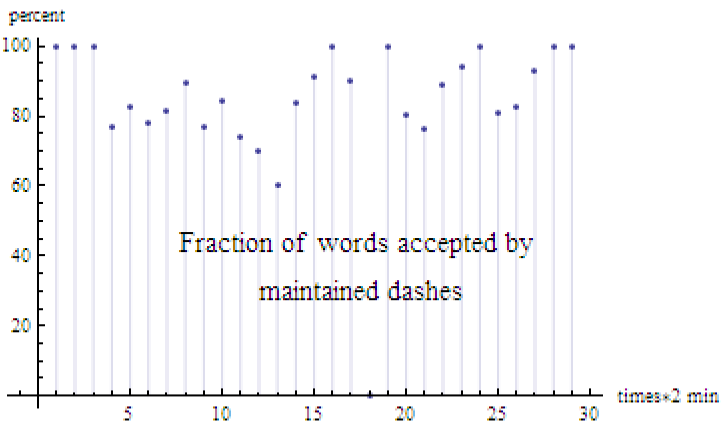

Figure 10. Same conversations, percentage of word selection ended by sustained dashes.

Results of user tests

Both RC-scanning with an empty column and Morse Code appeared usable right away, in part due to the fact that Morse Code was projected on screen. Figure 10 shows use of sustained dashes in simple conversations after acceptable learning effort. This suggests that sustained dashes speed up *some* conversations. If but a handful of hours are available to assess, advice and service a single client and family, RC-scanning will often be preferred. Slow or not, it requires minimal learning. Morse Code is faster on the long run. Sustained dashes might well be used with it, especially in conversations with a repetitive character[8].

Conclusion

Sustained dashes allow rapid access to word prediction if combined with Morse Code and can also be applied in RC-scanning with a single switch. Sustained dashes appear usable, efficient and clinically relevant.

8 Due to the recency information the same words will be present high in the prediction lists.

Author: Joris Verrips, MD Paramaribostraat 138" 1058 VP Amsterdam

Last changes: 1-12-2010

Same text plus appendices with tests, models, background and references:

www.depratendecomputer.nl/sustained dashes with appendices.pdf

Site: www.depratendecomputer.nl

Software: www.depratendecomputer.nl/writeeasysetup.exe[9]

www.depratendecomputer.nl/englishscanprojectsetup.exe

Videos: http://www.depratendecomputer.nl/somevideos.htm

About thirty videos of diverse input techniques.

http://www.depratendecomputer.nl/results.htm

Four videos that show the main points. Competing techniques, such as eye-gaze, are *not* displayed.

Email: j.verrips@planet.nl

Thanks to Marjan Sipma Flokstra for her contribution to redesign,

to Heidi Koester and David Colven for helpful criticism,

to John Paulin Hansen and others for moral support,

to Sem van der Pol, Jack van Dillewijn and Maarten Wang for cooperation with the experiments.

Illustrations

Figure 1 was inspired by a report on diverse scanning techniques by Colven and Simon.

Figure 2, Figure 3, Figure 5 and Figure 6 are screenshots made by the author.

Figure 4 was created with Graphic Calculus, software for mathematics education by Piet van Blokland, VuSoft.

Figures 7 to 10 were made with Wolfram's Mathematica.

9 This does not include synthetic speech, available at excellent quality and at a reasonable price from firms such as CereProc. Dutch and Spanish versions are also available.