WORKING NOTE AC91

Alan Creak
23 September 1994

MULTIPLE KEYING FOR FASTER COMMUNICATION

It is suggested that communication speeds for people using a conventional keyboard slowly may be substantially increased if multiple simultaneous key depressions can be used.

In information theory, the capacity of a channel can be computed as the product of the number of bits communicated by a message and the number of messages transmitted each second. The number of bits associated with a message is related to the number of distinct messages from which the selection is made, with a wider choice logarithmically increasing the number of bits.

The basic information theoretic treatment deals only with the physical process of transmission, and says nothing about the meanings of the messages transmitted; it is therefore not an accurate guide to the effectiveness of a channel for communication as it is understood in common usage. Nevertheless, in qualitative terms the ideas of information theory are illuminating, and it is generally true that – other things being equal – communication is assisted by broad channels using many symbols and by fast channels in which symbols are closely spaced in time.

The picture is usually complicated to some extent because most communication, and all communication in which people take part, involves several different channels used sequentially. Leaving aside what happens in our brains, human communication begins with signals sent from the brain to whatever organ of the body is used for communication (vocal tract, hands, etc.). At that point the message is reformulated as sound waves, (speech, perhaps becoming electrical signals in a telephone system), or as light (manual signing), or as key depressions (for a typist), or whatever. Perhaps after more or less complicated mechanical transmission, which may involve many transformations from one representation to another, the signal appears once more in sensible form, encoded as a light, sound, or tactile symbol. It is received by the recipient's sensory organs, and transmitted to the brain. Even in the simplest communication, therefore, several communications channels are used, and the overall effectiveness of the process is determined by the component, or the transformation, which has the smallest capacity. If this lies within one of the human participants in the communication, and is of significantly smaller capacity than is usual, we speak of a communications disability.

We have remarked elsewhere[1] on the factors which govern the communication rates in practice, and on the importance of matching the properties of channels at the points of transformation. In this discussion, we concentrate on one of these points of transformation which is critical for many people, and, by extending the matching between the communicating sides, discover an approach which could in favourable cases lead to a significant increase in communication rate.

USING A CONVENTIONAL KEYBOARD.

Many people with physical disabilities can use a conventional character keyboard as found on a typewriter or computer. Some may use an organ of the body (hand, thumb, nose, etc.), others use an artificial aid such as mouthstick or head pointer. Typically their motor control is sufficiently accurate to locate the required key reliably, but the action is often comparatively slow.

The communication rate is reasonably measured by the number of characters communicated per second. It would be convenient for our discussion if this quantity could easily be translated into key depressions per second, but in practice, this simple view is obscured by various devices commonly used to get more work out of the simple keyboard – so, especially on computer keyboards, we must also cope with escape, control, shift, option, command, and alternate keys, all of which can be used, alone or in combination, to switch between different keyboard modes, and all of which can lead to difficulties for people with limited physical abilities. As the subject of our discussion is independent of all these, we shall simplify matters by ignoring them for the time being, and directing our attention to the simple basic keyboard unadorned by complication.

With this simplification, we can think of a keyboard as composed of keys for a number of characters, each key uniquely associated with a single character. We assume that all the characters we require are present. The rate at which we can communicate using this keyboard is governed by the rate at which we can press the keys. The limiting factor is at the human side of the interface : the electronics will be able to keep up. The capacity of the channel depends on the nature of the muscular control of the keyboard user's arm and hand. Several characteristics of the control are important, but we shall simplify the discussion by concentrating on overall speed and precision. These two factors determine the rate of communication, though the relationship may not be simple. The precision determines the minimum separation of the keys, which may affect the number of keys which can be used, while the speed, given the separation and perhaps other geographical details, determines how quickly keys can be reached and pressed. How can we increase the rate ?

There are just two possibilities[2] : we can increase the rate of selection of keys, or we can increase the amount of information associated with each key depression. The rate of selection is limited by the human side of the interface, and there is little enough our machinery can do about that. We therefore consider the relationship between information and key depressions.

The information theory approach draws attention to the connection between the information content of a symbol and the number of symbols available, so one possibility is to increase the number of keys. To illustrate the point, we could (according to information theory) approximately double the amount of information per key depression by augmenting the basic keyboard with a separate key for every possible character pair. In practice, things are not quite so simple, for two reasons.

The first reason is that, because of the human side of the interface, there are strong constraints on the extensions which we can make – in brief, we can only add keys that make sense to people, and which they can cope with without undue difficulty. A few additional keys could be both productive and manageable. For example, we could add keys for common words (and, the), or for common combinations of two or more characters (th, wh, ing). The proposed extension goes far beyond that : it is a much larger keyboard which would impose significantly greater physical demands on its users – and it would be quite wasteful, because most of the keys would almost never be used. (How many words contain "zf" ?) The constraints here are cognitive in nature, reaching right back to the origin of the communications stream; we need to identify things which in some sense we can remember easily so that our brains can generate the stream of instructions sufficiently quickly to keep the rest of the system going as fast as possible. Notice that simply remembering isn't enough : we also want things which we can naturally fit into the mental processes involved in converting thoughts into hand movements and think about easily. The sort of extension we suggested is cognitively harder to use than a simple keyboard, because we have to make decisions about when to use separate characters and when to use the special keys, and therefore needs a little more thinking time to use effectively. A very similar phenomenon is observed with word prediction systems[3] : they can be very helpful for people whose physical movements are slow, because they have more time between actions to plan their next moves, but are a hindrance to those who can type quickly.

The second argument against increasing the number of keys has nothing to do with information theory. It is that special keyboards are expensive. If you buy a computer, you get a standard keyboard, and you don't want to have to buy another. A software solution is therefore potentially cheaper, and – at least on modern microprocessor systems – likely to be easier to install. What sort of software can we write which will increase the information per key depression while using a standard keyboard ? This depends on the information available to the computer when the keyboard is used, which in turn depends on the signals in the communications channel between keyboard and computer.

HOW THE KEYBOARD WORKS.

Now we have moved to the machine's side of the interface; raw speed is no longer a difficulty, and we can be confident of handling any information which arrives in the computer. We must also be very careful that we receive all the available information, and preserve it, for any loss of information is potentially a reduction in communication speed. It may be, of course, that some incoming information is not used for communication, and can therefore be discarded without danger, but we must certainly make quite sure that information is not useful before losing it. We must therefore look closely at the input stream to assess the importance of the different types of information which it carries. How are the data coming from the keyboard encoded ?

The most important signal in common usage is the key depression signal. It is a momentary signal which says something like "The Q key has just been depressed". At first sight, this is all we want. It is sufficient to tell us the order in which the keys have been depressed, and – if we want it – the time at which each key was depressed. If this is all we have, then any action taken by the computer must depend on the identity of the key, and perhaps the time of receipt of the signal. This is sufficient for normal typing – indeed, it is more than sufficient, because we do not usually need to know the time of the key depressions, as the sequence alone is sufficient.

In fact, though, we have lost some information, for the key-down signal alone is not enough to determine the current state of the keyboard; we cannot tell how long a key remains depressed. This information, or something equivalent, may be available. As well as the key-down signals, keyboards commonly emit signals when keys are released; with this additional information, the computer can keep track of how long keys are depressed. Many interfaces make use of this information to repeat the character, but it can also be used for other purposes – for example, to give uppercase characters without the burden of using the shift key. This is undoubtedly a useful extension to the normal keyboard repertoire, but it is gained at the expense of slowing down the keyboard action. To derive two distinct signals from a single character key by this method, the computer must be able to distinguish unambiguously between two different periods of key depression, and for someone with limited dexterity who may have difficulty in controlling the speed of manual operations, thus may require rather long key depressions to ensure reliable identification.

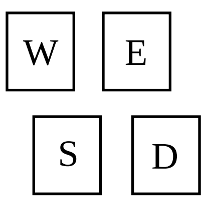
Given both the key-up and key-down signals, there is an alternative. Instead of timing the depression of a single key, we can look for simultaneous depressions of different keys. This is how (some of) the various modifier keys (which we have excluded from our discussion) work, and is recognised as a difficult operation for people with some physical disabilities. The difficulty, though, comes from the requirement for two distinct simultaneous actions – you must hold down the shift key while pressing the G key.

Our suggestion is that, if two keys can be simultaneously depressed with the *same* action, we can exploit this as a means of widening the communication channel. We are led to this notion by considering the principle of matching mentioned earlier. Dual, and occasional triple, key depressions in conventional equipment are matched to the abilities of people with good motor control in two hands, but do not match the abilities of many less dexterous people. To provide multiple keying for these people, we seek multiple-key signals which can be produced using a conventional keyboard using only one hand.

SIMULTANEOUS KEY DEPRESSIONS.

We suggest that people with quite accurate but slow physical control could significantly increase their communication speed if they were able to use combinations of adjacent keys depressed simultaneously as – in effect – additional keys. While experimental measurements would be needed to determine just what sorts of action can usefully be exploited, simple preliminary tests suggest that certain simultaneous key depressions can be achieved by hand or with rigid pointers without much more difficulty than selecting single keys. It seems possible that with practice, and perhaps using shaped or flexible pointers, reasonable communication could be managed.

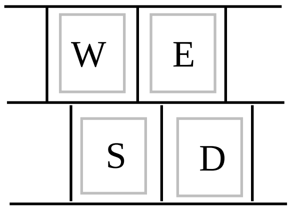
Certain constraints are imposed by the conventional keyboard layout. For reasons which have more to do with the need to pack the machinery of manual typewriters between the keys than with the constraints of computer keyboards, the keys are not arranged as symmetrically as one might expect. For example, the diagram shows a part of the conventional keyboard.



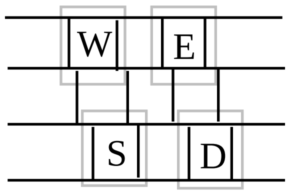
Notice that the W key is much closer to the S than is the E, so it is much easier to depress W and S simultaneously than E and S. S and D can easily be managed; W, S, and E are also possible, though a little more difficult, as are S, E, and D, and all four keys can be managed with care.

If all these combinations (S, (SW), (SD), (SWE), (SED), (SWED)) should prove feasible, the number of "keys" on the keyboard is in effect approximately sextupled. (The factor is rather less than six, because not all combinations are available for keys at the top or right of the keyboard.) Can we use this additional capacity effectively ? As we have already remarked, it will only be useful if we can make it both accessible and comprehensible.

It is interesting to digress briefly on just what is going on when we do this. A keyboard is in essence a quantiser – it converts the continuous movement of a finger or other object in the two-dimensional space above it into a set of discrete signals, one for each key, thereby dividing the space into distinguishable cells.



Our proposal can be viewed in two ways. The first is to use the same raw input signals to produce a more extensive quantisation. This view is appropriate if the keying device is rigid and constrained always to the same attitude, and makes clear the requirement for much more precise position control than is needed for conventional keyboard use. This can be seen as a finer spatial division thus :



With a more flexible (and controllable) keying device, an alternative view is possible – and, we believe, preferable. Now we can see the different selections of key combinations as a way of tapping a new coordinate of the keying device : the *attitude*. Each attitude selects a certain type of key combination, and is effective in a number of positions with essentially the same spacing as the keyboard itself. We gain the additional information by exploiting a communications channel (the attitude) which we had previously ignored. In practice, it seems likely that the effective behaviour will lie somewhere between these two models, but both views will contribute to the performance to some extent.

We have already covered one aspect of the question of accessibility in finding which key combinations can easily be depressed simultaneously. We have to qualify our remarks by adding that the simultaneous depressions take a little care to manage, and because of this the actions are slightly slower than those for depressing single keys. This may be an obstacle to people who can use a keyboard fairly quickly, but we would expect a break-even speed below which the simultaneous depression technique would be advantageous.

ENCODING.

To ensure that the system developed is comprehensible, we must pay careful attention to the encoding technique – that is, how the meanings are associated with the key combinations. The object is twofold : the associations must be easy to remember, and they must be significant in increasing the communication efficiency. Notice that this statement of the encoding problem, while similar to that of encoding new keys, is not the same. With the new keys, there was no existing context to guide associations; using multiple keys, we begin with the letters themselves, and can reasonably try to construct codes which begin from this mnemonic base.

One way to make it easy to remember conventional interpretations is to reduce the conventions to a few simple rules. This is the principle (though not always obviously the practice) behind much of graphical user interface design. It is unfortunate that English spelling is too various to permit many such generalisations. There are a few obvious possibilities. For example, one could regularly denote a combination of a base character (taken as the lower left character in the diagram above) with the above right key as a common character pair with the base character as the first. (SW) could then be interpreted as SH, with corresponding encoding for TH, WH, CH, PH, NG, and perhaps one or two more. After that, though, the convention breaks down, as there are few other very common and easily recognisable character pairs. It is always possible to invent a list of character pairs, some of which might be useful, but it is hard to escape the feeling that, say, the (XS) combination could be more profitably used.

An alternative is to follow the Minspeak pattern[4], and rely on association of ideas to suggest the interpretations to be used. It is certainly sensible to provide for people to define their own encodings, but the conventional keyboard is rather less evocative than the collection of Minspeak keys, and association may be harder.

On the other hand, this picture may be painted too dark. Many people (ham radio enthusiasts, Unix freaks, IBM personnel) can remember large numbers of quite meaningless letter combinations and use them effectively. This may not be quite as good as it sounds, though, for in such cases, memory is assisted by widespread use of the terms in a community in which they are current; unless similar widespread conventions can be adopted, that factor will be of no help in establishing an encoding convention for the keyboard. Perhaps this is a factor which requires experiment for proper evaluation.

REFERENCES.

1 : G.A. Creak, R. Sheehan : *The representation of information in rehabilitation computing*, Auckland Computer Science Report #54, Auckland University Computer Science Department, July 1991.

2 : G.A. Creak : *Essay on a multichannel computer interface for people with physical disabilities*, Auckland Computer Science Report #93, Auckland University Computer Science Department, June 1994, page 17.

3 : M. Soede, R.A. Foulds : "Dilemma of prediction in communication aids and mental load", in *RESNA 86 Employing Technology* (Association for the Advancement of Rehabilitation Technology, 1986), page 357.

4 : B. Baker : "Minspeak", *Byte* **7**, 186 (September, 1982).