

AN ASSIGNMENT OF KEY-CODES FOR A JAPANESE CHARACTER KEYBOARD

Yuzuru Hiraga, Yoshihiko Ono, Yamada-Hisao

Dept. of Information Science
Faculty of Science, University of Tokyo
7-3-1 Hongo, Bunkyo-ku
Tokyo 113, Japan

Summary

An input method for ordinary Japanese text is proposed. A regular keyboard has been selected as the input device, as a means to effect touch typing. The primary objective of the design of the code system is to make the keying fast, least fatiguing and error-free. Modeling for performance simulation was carried out, which showed the system to be better in its efficiency in comparison with the Standard English keyboard.

1. Introduction

Since its first appearance some 100 years ago, the English keyboard typewriter has come to be one of the most indispensable tools of the Western society today. This is not only for the great role it plays in business offices, but also for its widespread acceptance for everyday use by the majority of the people. Indeed the typewriter caused a social revolution. The most remarkable of its influence is that it opened up an entirely new career (especially for females), -- the typist. And now, the invention and rapid development of the computer has opened a yet new important role for the typewriter keyboard as a remarkably simple, though effective interface between machines and human beings.

While all this was going on in the Western hemisphere, no device in Japan (and China, and other non-alphabetical countries) had attained the high potential of the typewriter keyboard throughout this period. The introduction of computers has not altered this situation, since the alphabetical keyboard was directly imported as the computer-input device.

Why so? The answer is quite simple - it is because of the huge character set used for the Japanese writing system. This affects all aspects of Japanese text processing, such as how to create and store character fonts, how to output them to the end devices, and most serious of them all, how to input the text. But eventually the rapid development of computers and peripheral devices is becoming to enable us to do what was impossible. Complicated characters may be printed by means of a fine dot matrix printer. Memory units to store these character fonts and processors to handle them are becoming reasonable in size and economy. Still, the problem at the inputting end remains.

Our intention in this note is to suggest an input method that would assume the position in Japanese text processing what the keyboard does in Western societies. That is, we are laying emphasis on the efficiency of input, presupposing professional uses. We do not assume the use by untrained users. We further note that the current "QWERTY" English keyboard is by no means optimally designed, and we must not follow the path that the Western society has followed and, as a consequence, suffers today from being stuck with the poorly constituted device.

2. A Brief Survey

The Japanese language uses an extensive and complicated character system. There are two types of phonetic symbols (*kana's*), *hiragana* and *katakana*. They are in one to one relation with each other, and about 150 in all. Arabic numerals and alphabets are also commonly used. An indefinite set of punctuation marks is quite similar to that of English. Finally, there is a huge set of ideograms called *kanji*. There are about 50,000 of them in all, though an average person would use some 800 in daily life. About 2000 are taught in the elementary education to the ninth grade.

With this complicated system, the usage is quite flexible, that is, it is legal to write texts in quite an arbitrary mixture of these character types. So, it is possible to write down a Japanese sentence with only the phonetic symbols, *kana*, but that would deter the readability of the present day text. One reason that *kana*-written texts are hard to read is because the Japanese language contains numerous sets of homonyms, each of which is written the same in *kana* form. So for ordinary text handling, we must take into account the *kanji* characters. That immediately means that we must deal with a character set of a few thousand.

Several types of input devices have been proposed and realized in the past. They may be roughly categorized as in figure (2-1).

(1) Direct Methods

"Direct" means that all information necessary to select a correct input character is provided by the typist. (The term "typist" is used in a general sense as the person who handles the input process.) This type includes:

- 1) One-to-one keying that uses one separate physical key for each character. Devices for this may be made without the help of electronics, and are relatively easy and inexpensive to build. Several types of these are already on the commercial line.
- 2) Multi-shift keying which uses 100 to 300 main keys, each key standing for some number of characters from which a selection is made by means of shift keys. Thus, a character is printed by pressing two keys at a time.

This principle may be extended to:

- 3) Chord keying, which uses a combination of keys hit simultaneously to represent a character, or even a word or a phrase. This type of keying is used mainly in stenographs, and requires a high level of trained skill.
- 4) Multi-stroke keying, on the other hand, is operated by hitting only one key at a time. A character is expressed as a sequence of key strokes. This may be implemented on a keyboard of a small size of 30 to 40 keys, and is amenable to touch-typing. This type would require electronic devices for implementation.

(2) Interactive Methods

These are methods to support multi-stroke methods. Since the character codes for multi-stroke methods are (or seem to be) hard on memory, the operator first types simple

Table (2-1): Input Methods for Japanese Text. (Adapted from Yamada)

<u>Direct Methods</u>	
One-to-one	Wabun typewriter (planar, drum) Tablet
Multi-shift	One-page Multiple-page (Hitachi)
Chord (Stenotype style)	
Multi-stroke	Uniform length code With shift (Kantec, Taikei) Without shift (Rainputto, Yamura, Superwriter) Variable length code Kanji, coded Kanji, synthesis from parts.
<u>Interactive Methods</u> (Multi-stroke)	
	By sound By shape Hybrid (Hitachi) Kana-kanji conversion (Toshiba) Others

Handwriting Recognition
Voice Recognition

Table (2-2): Two Types of Operations. (Adapted from Yamada)

methods features	Sight method (Hunt & Peck)	Touch method (Blind)
Eyesight	Strained by moving from manuscript to keyboard to platen.	Mostly stationary at the manuscript.
Decisions	Definitely conscious, especially at character selection at the keyboard	Subconscious reflex of hands and fingers
Rhythm	Lacking	Much
Speed	Low	High
Skill acquisition	Production work can commence without much training.	Needs systematic training; heavily depends on training method.
Fatigue	High	Low
Mental stress	High, tend to cause restlessness.	Low
Operator's morale & pride	Low	High

associatively recallable codes, and the machine answers back with possible selections from which the operator picks the proper choice. Among this type is the kana-kanji conversion method, in which the text is first input by its phonetic representation, and the machine converts it into proper ideograms.

Comparing these methods from the viewpoint of human factors, we see two extremes on the line. One is the Sight (or Hunt & Peck) method, which requires constant watching of the input device, either because there are too many keys, or the keys are not position oriented, like a dial lock. The other is the Touch (or Blind) method, where the operator requires practically no viewing of the input device. This is the way of professional (English) typists. For touch typing, it is important to keep the keyboard small enough so that all keys are within the reach of fingers. We note that touch typing is done in the subconscious level of mind, that is, the motion of the hands is a conditioned reflex, and not a result of some consciously made decision. The operation in interactive methods lies somewhere between the two extremes. They require constant intervention by the machine, and the subconscious flow of mind is kept interrupted. The merits and demerits of the two extremes are listed in table (2-2).

It is natural to conclude that when both methods are available, touch methods are far more effective than sight methods for a trained typist. One question is whether an enormous character set would ever be made touch-typable in the first place. We will discuss this problem later in section 8.

3. Designing Principles

The crucial point in the design of our code system is its "efficiency." By this we mean a code system that embodies the following features:

- (1) A high level of input speed is attainable. A good English typist easily types 80 to 100 words per minute. A method that cannot outpace handwriting is not acceptable. Hunt & peck devices for Japanese tend to be even slower than handwriting, so they would serve as a device of necessary evil only for final printing.
- (2) The typist suffers from less fatigue. Employing a trained operator as a user means that long hours of continuous work are expected from her. Thus attention must be given to the work load on the typist, not to impair the health of the typist while maintaining the high rate of input.
- (3) A high rate of accuracy may be maintained. That is, the codes should not be susceptible to erroneous finger motions.

These are of course not independent factors, but are rather closely correlated to each other. Of them, speed is in a way the most decisive, and also the most appealing, factor. It is also the easiest to examine experimentally. The other two involve complex human factors study, to measure and to analyze.

From the discussions in the preceding section, we see that the best method to accomplish the above objectives is one that allows touch typing. Our objective would be met by a multi-stroke code system on a small keyboard. This immediately implies that the system is not usable by an untrained user, for she must look up the code of each character to input.

There are several multi-stroke systems currently implemented and put to use, most of which are provided with some kind of mnemonic features for character codes. They may be in terms of the pronunciation of kanji in kana or alphabet, or built around the constructs, or visual forms, of kanji. Assume a situation that a practical number of codes are learned by heart, that is, they are not memorized by way of the conscious mind, but are attained as subconscious reflexive motions. What counts then is that the input operation is highly efficient. The mnemonic codes are not likely to work in favor of efficiency, in fact, often against it. So, we basically dispense with all intentions for mnemonic associativity, and pursue efficiency in the main. We will even make no distinctions among different character types, namely kanji, hiragana, katakana symbols, and the punctuation marks. This will make it still harder for the untrained user. Consequently, our code system is primarily for the use of trained personnel, who are likely to be professional typists.

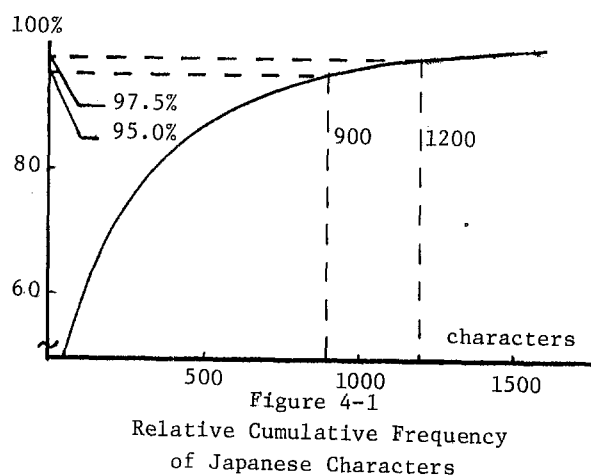
Our measures for the efficiency of keyboards has been obtained from the analysis of the current English keyboard.¹ We find from them that keeping a steady rhythm is the best strategy for speed. Alternate hand stroking is the best for this purpose. This suggests that a code should consist of an even number of strokes -- where a text character is entered by an alternate hand sequence such as R(right), L(left), R, L..., then the next again by R, L, R, L..., and so on. Of course, the total number of strokes per code must be held as small as possible.

To meet these requirements, the code system for the entire character set can not be of uniform length, but be a 2-levelled one. We assign 2-stroke codes to the set of *basic* characters, and longer codes (presumably 4 or 6 strokes) to the rest, called *outside* characters. This partitioning has certain other merits. We may require only the codes for the basic characters to be learned completely (by hands). The longer codes for outside characters may have mnemonic features. (This would not harm the whole typing process if codes are constructed with care. Outside characters cover only a small fraction of the total text.) The size of the basic character set will be about 900 on a 30-key keyboard, and 1200 on a 40-key keyboard. The latter does not use all the possible 2-stroke combinations because there are key pairs which are not suitable for good touch typing.

4. The Determination of the Character Set

The selection of the basic character set is based upon the frequencies of the usage of characters in a sample text, taken from newspapers.⁸ However, alphabets are excluded from the list since it would be better to treat them in a different keyboard mode. All Arabic numbers are among the 50 most frequently used characters, and they are included in the basic character set. As for punctuation marks, only the standard ones are included. Most of the kana characters are among the top 900, and it is of no problem to include them in the set. The rest of the characters are kanji. Considering a statistical fact that about 500 kanji and 150 kana characters are used by an average person for his daily use, although the set may change gradually, our selection of the basic character set seems reasonable.

By examining the cumulative frequency graph of Japanese characters given in figure (4-1), we see that 95% of all the usage is covered by the top 900 characters. However, if the distribution is looked at in *jukugo* units, that is, a character combination that forms a concept in Japanese (as a "word" does in English), then 87% of the whole text is covered by the top



900, so we might extend the size of the set to perhaps 1200 or more, by using not 30 keys but 40 on the keyboard. Even in this case, it is not necessary to use all the available key pairs for the reason of maintaining the quality for good touch typing performance, and the top row will be used only in alternate hand stroking. This extension is planned as a future work.

5. Coding of the Basic Character Set

The coding of the basic characters is based strictly on the efficiency of finger movements. Our method maps characters, arranged in the order of the frequency of usage, to the key pairs arranged in a suitable ordering as defined below. Second order adjustments will be made afterwards.

The ordering of key pairs is obtained by assigning certain weights to certain characteristics of hand motions and using their linear sum for each key pair. The characteristics that are thought to have a greater importance will be given a larger weight. In this way, the ordering of 900 key pairs on a 30-key keyboard was made.

Though the key pairs have an ordering based on their inherent features, this ordering is not immediately usable to fix the assignment of characters directly. This is because that the typing process is not a collection of isolated key pairs, but their continuous sequence. For example, if key pair "d-k" is with a high score, then its reverse, "k-d" would also be with a high score, but frequent appearances of these two key pairs would result in the frequent tapping motion of key pairs "k-k" and "d-d" in the interval of consecutive "k-d"s and "d-k"s, or vice versa, which are known to be less preferred. This would also be adverse to alternate hand stroking as well.

Through such considerations, the desirable keyboard characteristics may be itemized as follows:

- (1) The whole typing procedure is to keep as much keying rhythm as possible. Fluent rhythm, as well as high average of typing speed, is best realized by alternate stroking by both hands. Thus, it would be our principal objective to let the code system be such that it would allow alternate hand stroking as much as possible.
- (2) Hands should not be moving up and down incessantly on key rows, but stay in the same row as much. Thus, strokes on the home row should be used as much as possible and excursions to other rows should be held minimum.

Comparing between the upper and the bottom row, all evidences point out that hands are more fluent on the upper row, so the ranking of rows should be in the preference order of the home, the upper, and the bottom.

- (3) Fingers should be loaded in proportion to their dexterity. In typing motions, fingers are divided into the stronger ones (index and middle fingers) and the weaker ones (ring and little fingers). Index and middle fingers are not so much different in their capacity and functions. However, we must keep in mind that each index finger must cover two inner columns. The difference between ring and little fingers is also not so obvious. Although a ring finger is superior in its stroking force in typing motions, a little finger may have the advantage of the twisting motion of the wrist. (Though in reality, this motion might lead to more typing errors.) In the present study, little fingers will be given more emphasis than the ring. Numbering the key columns 1 through 5 from the outer one inward, their ranking in the order of manipulative superiority will be 3, 4, 1, 5, 2.
- (4) The number of awkward keying sequences must be decreased as much as possible. Almost all awkward key pairs are of one-handed stroking, again attesting to the importance of alternate hand stroking. The major awkward key pair sequences, in the order of their disadvantages are:
- 1) Hurdling: the stroking from the upper to the bottom row or vice versa, jumping over the home row.
 - 2) Reaching: the stroking of different keys with the same finger.
 - 3) Tapping: the stroking of the same key.
 - 4) Rocking: stroking with adjacent fingers, especially from an inner to an outer one.

There are other minor considerations that should be made, one of which is the load balancing between the two hands. We have loaded the right slightly heavier, but we do not consider this factor that important, and the roles of the right and the left hand may be reversed to obtain a code set with the mirror image assignment to hands.

The actual weighing process starts by accommodating for condition (1) above. The key pairs are divided into 4 blocks, namely RL, RR, LL, and LR blocks, where symbols L and R stand for the hands that stroke the keys of the pair. The blocks are given preference in the order given above, and key pairs in each of the blocks are then ordered by taking further conditions into account.

The above ordering of blocks comes from the distribution of the frequencies of the usage of Japanese characters. Since character pairs have a frequency distribution proportional to each of the individual character frequencies, characters belonging to blocks of lower rank would appear most of the time alone in a sequence of characters belonging to the top-ranked block, forming a singular point. If this character belongs to block RR, then the sequence would look like:

R L R L R R R L

↑

but if LR, then:

R L R L L R R L

↑ ↑

They would have the same effect on the average LR-sequence length (described later), but the latter causes two singular points in the basic "R-L" sequence. Also, in the R-L-R-L environment, it is possible that they might be typed in the reverse order. Thus, from the alternate stroking point of view, it is preferred to use block RR, not LR, next to RL. (If a character pair is made of characters with nearly the same frequencies, then the above statement is not true.)

Within individual blocks, conditions (2), (3), ..., are evaluated and weighed accordingly, and the whole ordering is decided. Awkward sequences are deliberately given negative weights in order to bring down their ranking, thus decreasing their occurrences when the codes are used.

The above procedure takes into account only key pairs, or, from the viewpoint of the source text, the distribution of individual characters. For the further improvement of keying motions, considerations on the distribution of character sequences should be made. But tests on our code system showed that these secondary changes would not seriously affect the overall rating of the system, so this line of modifications has not been fully carried out in the present status. One exception is the introduction of entry codes into the outside character set. Codes "j-f" and "f-j" are used for this purpose. The entire code table for the basic characters is given in the appendix. This table is yet subject to further changes.

6. Coding of the Outside Characters

As for the outside characters, we do not insist on having the codes made free of mnemonics as for the basic characters. One reason for this is that the length of the codes are too long for easy learning, and another is that there are too many of these characters. But the main reason is that since they cover only a small fraction (5% for a 30-key keyboard, 2% for a 40-key keyboard) of an average text, those codes for seldom used characters may be easily forgotten, therefore they should be consciously constructible. Since most of the outside characters are kanji characters, (leaving out a few exceptions of punctuation marks,) we may code them mnemonically using the features inherent in these characters, where the mnemonic codes are the codes of the basic characters. That is, the codes are constructed by a two-stroke entry code that indicates that an input sequence of an outside character has started, followed by the codes of two basic characters that express some feature of the character in question, making the number of strokes 6 in all. (By means of shift keys or hardware modifications, we might be able to omit the entry code.) In this way, we have coded 2000 of the outside characters as a start.

The mnemonic features of an outside character to represent it must meet such requirements as:

- 1) The mnemonics of the character can be easily recalled, or is recoverable from the character itself.
- 2) The introduction of additional characters to the character set will not require the modification of the whole coding system.

There are various features of characters that we might utilize for coding the outside kanji. Various kanji dictionaries use radicals (substructures of kanji), the number of strokes used to draw the character, or the reading (in phonetics) to index a certain kanji. Of them, the reading is not suitable, for a character may be read in several different ways, or several different characters may correspond to a reading. In addition, as the character to be typed becomes a less commonly used one, it will be less probable that the typist would know how to read it. The number of strokes is not usable either as a means of specifying a character because its exact value is not so readily perceivable. Most of the kanji characters are composed of two or more radicals. A radical may be a kanji itself, or be a substructure of the whole and may appear in a good many other kanji characters in common. By inspecting a complex and less used outside character, we can often agree upon a set of two

radicals which embody the essential features of the character. This identification process is a subtle and very much a subjective psychological one, heavily dependent on the past experience of individuals. Yet different individuals seem to have a nearly same set of choices for each character. We shall call these chosen characters component characters. When the choice is not unique, we use a reasonable number of alternatives as well.

All component characters we use for mnemonics are from the basic set. The codes made of these pairs may be classified into the four types given below, where W in (1) through (3) indicates that the whole of a component character is representing a part of the outside character. The P in (2) through (4) indicates that a part of a component character is representing a part (the whole in (4) only) of the outside character. "+" means a combination of components. "P-W" in (4) means that a component character W is eliminated from the other component character P and the remaining is the outside character. Alphabet letters below component characters are the codes for the characters.

Examples of codes for outside characters

	Outside Char.	Components and Codes	Other Components and Codes	
(1) W+W	銅	金 同 IT ;X	鉄 同 JJ ;X	(P+W)
(2) P+W	迷	道 米 MZ ;	米 通 ; HX	(P+W)
(3) P+P	酔	配 作 DC /W	酒 昨 VT ZN	(P+P)
(4) P-W	皿	悪 心 XP MN		

This way of coding may cause some conflicts in rare cases where the code typed in is not for the character intended, but for some other one. For these cases, we are leaving it at present to the typist to verify the character on a display, and correct it if necessary by trying another code. There are still a lot of possible codes unused, and other outside characters beyond the 2000th may be coded similarly. The 6 strokes necessary to type in a character might seem too many, but as we have seen, their appearance in text is not so often that the effect of the code length is not that critical to the overall speed. We place more importance on the good typing rhythm in 6-character codes, and the ease of the recall or reconstruction of codes.

7. Evaluation of the Code System

The ultimate test of a code system would be to actually measure its productivity on a real system, but we have not been able to go that far yet. In addition, those experiments would involve human factors problems not well understood even at present. One example is the problem of an objective measurement of mental fatigue.

Hence, we chose to show some statistical figures derived from some sample text.⁸ In the test, we considered the basic character set only, and treated the outside characters as if each consists of 2 invalid strokes. The QWERTY keyboard run on an English text is also made for comparison. Results are shown in table (7-1) and figure (7-2). Our code is referred to as "T-code" in the table. *LR-sequence length* in table (7-1) means the expected length of the alternate hand stroking starting from an arbitrarily chosen character of the text, counted by stroke intervals. Such expected value for a random sequence is 1.00, since

$$E[\text{length}] = (1/2)*0 + (1/2)^2*1 + \dots + (1/2)^{n*}(n-1) + \dots = 1.00$$

From the table, we see for our code that:

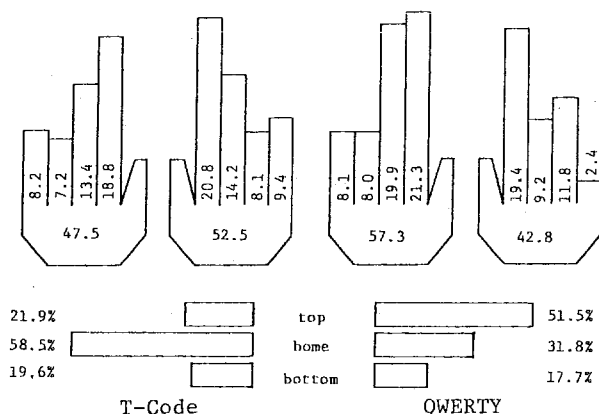
- 1) Hands are evenly loaded, slightly lighter for the weaker left hand.
- 2) Strokes are concentrated on the home row, so that the moving of hands from row to row is held minimum. Note that quite a different situation holds with the QWERTY keyboard, where more than half of the strokes fall on the upper row.
- 3) The loading of fingers is in a qualitative agreement with the conjectured strengths of the fingers. We feel, however, that index fingers (covering two columns each) may be loaded a little too heavily. It is possible to lighten their burden by weighing the ring and the little fingers more, but this is a trade-off problem, and we are unable to tell which is better at this moment. In any case, our code for Japanese gives much cleaner distributions than those of QWERTY. For example, the right ring finger on the latter is readily seen to be overloaded. (This finger covers keys "o", "l", and ".")
- 4) Our keyboard has attained a good low level of awkward sequence rate, which is about the same as for the improved English keyboard of Dvorak.
- 5) A high rate of alternate hand stroking has been obtained, but the reason for this is obvious, since it was our primary design objective. This may be further improved with keyboards having more keys, but that would also raise the rate of awkward sequences, as well as the use of the less preferred top row.

The efficiency of our code system is quite clear, in comparison with the QWERTY keyboard data. The figures attained even the level of the *Dvorak Simplified Keyboard (DSK)*, which is thought to be near optimal for English. The results are especially favorable in awkward sequence rate and LR-sequence length. The LR-sequence has far outlengthed even that of Dvorak. We believe that a further significant reduction of the awkward sequence rate (through the secondary modifications of the codes) is hard, if at all possible. Still, it may be beneficial to try to accommodate for the character pair distribution of Japanese texts, because in fast typing, the transition of every second stroke between characters might have a nonnegligible effect on speed.

As an attempt to see the effect of such factors, a computer simulation of typing motion was made. The aim of the simulation was to find the coordination between hand movements and finger stretches for key stroking, in order to model the most adequate overall typing motion. The designed model was intended to capture the significant features of real typing procedures where the typist would look ahead a number of characters (maybe taking words as units), and unconsciously hold the hands over a position that would require least finger

Table (7-1): Stroke Distributions.
(Space bar and top row excluded from QWERTY data.)

Hand Distribution				
	T-code		QWERTY	
Left	47.5%		57.2%	
Right	52.5%		42.8%	
Row Distribution				
	T-code		QWERTY	
Upper	21.9%		51.5%	
Home	58.5%		31.8%	
Bottom	19.6%		17.7%	
Finger Distribution				
	T-code		QWERTY	
	Left	Right	Left	Right
Index	18.8%	20.8%	21.3%	19.4%
Middle	13.4%	14.2%	19.9%	9.2%
Ring	7.2%	8.1%	8.0%	11.8%
Little	8.2%	9.4%	8.1%	2.4%
Rate of Awkward Sequences (Total, (Left:right))				
	T-code		QWERTY	
Hurdle	2.1% (2.0%:2.2%)		9.6% (5.4%:15.3%)	
Reach	5.2% (4.1%:6.2%)		8.2% (7.5%:8.7%)	
LR-sequence Length (Expectation 1.00 for random.)				
	T-code		QWERTY	
	7.69		1.09	



motions for the stroking sequence. The outline of the model is as follows:

A certain key is hit by the finger assigned to it, holding the hand over a certain position. The hand positions were quantized into 4 locations, each of them being the most convenient position to hit keys in a certain row, namely, the top, the upper, the home, or the bottom row. A finger may hit a key with no stretching effort by moving the hand over the row the key is in, or, it may stretch over to hit a key in the row right above, or right below, the row the hand position stands for, without moving the hands. The load of the whole process is the sum of (1) the load on the finger to stretch over, (2) the force to hit the key, and (3) the load on hand movement. The load parameters were adjusted so as to realize (1) a case where no finger stretching would occur, or (2) a case where no hand movements would occur (using the lower three rows), or (3) other cases in between. The hands may either (4) return to the home position (that is, the position over the home row) after a given blank period, or (2) stay in the position where the last key was hit. Also a mechanism to look ahead 1 to 6 characters was used, in order to plan for an optimal movement. Strokes continued on the same hand, and also those continued on the same finger are given extra load factors.

The results of simulations showed the effectiveness of DSK against QWERTY for English. QWERTY requires some 20 to 100% larger amount of hand motion when compared with DSK, depending on various parameter settings. The worst case occurred when the parameters were such that no finger stretches were allowed, no load was assigned to the hitting of the key itself, and the hand always returned to the home position after each stroke. The difference between these two became smaller as finger stretches were brought into the picture, and the force to hit a key was taken into account. The difference was about 20% at the least. This fact indicates that with QWERTY, a good typist would mostly hold hands over the upper row, rather than the home, since more than 50% of the strokes are on the upper row. Our code for Japanese texts gave about the same figures as those of DSK.

The effect of lookahead did not show a significant difference. We also tried a modified model that has a *built-in* lookahead ability, by assuming additional hand positions between adjacent rows, which serve as a transient position when moving from a row to another row, but this also gave results similar to the original version.

From these results, we may conclude that our code system, though it is still of an experimental status, would be able to lead to a good performance level in touch typing of Japanese texts.

8. Remarks on Implementation and Training

So far, we have been deliberately ignoring the problem of the attainability of touch typing. It may seem counter-intuitive that codes for so many characters will ever be learned for typing within a reasonable period of training. You might think this to be analogous to the training for playing the piano. So small a number of people will ever obtain the level of skill necessary to become a professional pianist. But this analogy is not appropriate, since there are millions of professional typists in the Western societies. This fact shows that the skill to touch type has not so much in common with piano playing.

But is this because the Western typewriter has a one-to-one mapping between the characters and codes -- namely, the keys?

9. Concluding Remarks

Again, the answer seems negative, as we see that the typist treats characters not individually, but as key stroke sequences by units of words. We may find a better analogy in Morse codes. Morse codes are not too complicated for an average person to learn. The transmitting and the receiving skill of the code can be learned by anyone in a reasonable amount of time. The trick for the best learning procedure there is in that the codes be learned as a conditioned reflex. It is an established fact that if one first learns the Morse codes mnemonically, he has to give up those mnemonics eventually in order to attain a good level of communication skill. The same holds for the touch typing of Japanese in multi-stroke codes.

One might be still bothered by the size of the code set. We have set an upper limit of about 1000 for this. Whether this is a reasonable choice is not so obvious, and the best and only way to see this is to actually test it. We are yet to carry out this training experiment.

There already exist some practical experiences by others with Japanese touch typing in almost mnemonic-free multi-stroke codes. Example learning curves are given in figure (8-1). We see that the speed of stroking progresses approximately at the same rate as that of English typewriting for the same amount of training.

Note that in Japanese, the unit of a concept that matches the English *word* consists of about two characters on the average, while the average length of an English word is about 5 strokes. Then with a 2-stroke code, about the same amount of information may be represented by the same number of strokes. Thus we conclude that Japanese touch typing will be as powerful as English typewriting in the handling of Japanese documents.

One important task is to develop an appropriate method of training. The period necessary for training will be greatly affected by the quality of the teaching method. We see that this is still a problem for Western typewriters after 100 years of history. This is not a trivial task, and if we are to organize a horde of typists as a professional institution, a careful and extensive effort must be addressed to this task.

We have proposed an input method and an associated coding system for the input of Japanese texts. The principal goal has been to realize an input system that would allow a high degree of touch typing. Our code system may not be optimal as yet. Nevertheless, we are convinced that it can attain a higher performance level than any other method proposed up to the present.

What we wish to emphasize is that if one wants to obtain good productivity (in any field), one has to pay for it, and when provided with well designed mediums, one will certainly get his investment for training amply paid back. The majority of the input methods for Japanese that are now receiving attention emphasize their easier accessibility by beginners. If only the realization of good sales is what matters, then that might be a better marketing strategy. But if the device is to take hold of a firm position in the society as a truly efficient tool of production, then it should not stay as a beginner's toy. It should be well worth several month's training to be able to prepare documents several times faster than handwriting.

With the advancement of technology, machines may go on evolving without bound. Then an efficient communication with machines would become a critical problem. It is high time we realize that one of the most difficult problems in technology is the design of interface between human beings and machines, and that should aim at well trained operators.

Acknowledgements: We would like to thank Dr. S. Kawai for helpful suggestions and discussions on our work. We are grateful to Mr. M. Mogaki for the preparation of the conversion table for Japanese character codes, which was of great help to our work. We also thank Mr. J. Jan for helpful comments.

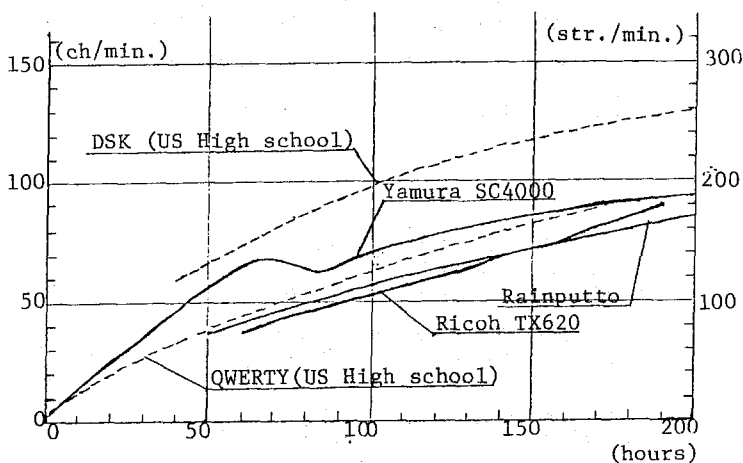


Figure (8-1): Learning Curve.
3 Japanese multi-stroke keyboards and
US high school students.

References

- [1] Hiraga, Yuzuru; Ono, Yoshihiko & Yamada, Hisao; "An Analysis of the Standard English Keyboard" (October 1980) Elsewhere in these Proceedings.
- [2] Yamada, Hisao & Tanaka, Jiro; "A Human Factors Study of Input Keyboard for Japanese Text" (December 1977) Proceedings of International Computer Symposium 1977, National Taiwan University, Taipei, Republic of China
- [3] Yamada, Hisao; "An Ergonomic Comparison of Kanzi Input Methods" (in Japanese, March 1980) Printing Information, Japan Association of Graphic Arts Technology, 17 (March 1980), pp.4-12
- [4] Yamada, Hisao; "A Historical Study of Typewriters and Typing Methods: from the Position of Planning Japanese Parallels" (February 1980) Journal of Information Processing, Vol. 2, No. 4, pp.175-202.
- [5] Dvorak, August; Merrick, Nellie L.; Dealy, William L. & Ford, Gertrude C.; "Typing Behavior, Psychology Applied to Teaching and Learning Typewriting" (1936) American Book Co., New York 521pp.
- [6] Dvorak, August & Dealy, William L.; "Typewriter Keyboard" (May 1936) U.S. Patent 2,040,248, 8pp.
- [7] Yamada, Hisao; "A Letter Selection Method by Typewriter Keyboard" (July 1979) Patent Application, No. 54-096033, 19pp. and 4 figs.
- [8] "Text Data from Three major Japanese Newspapers" (1966) The National Language Research Institute (1,500,000 Characters)

APPENDIX

CODE TABLE

Each of the 4 overall 3x5, and their component 3x5, blocks of characters represent key positions on one half of the keyboard. The 4 large blocks stand for the first stroke, and the small ones for the second. The headings of the large blocks indicate the hand sequence of the key pair. ◆ is the entry code for the outside characters.

R-L

歩キヤコナ 申わうあ本 店全バ後間	回せ出山金 スラ4こさ 持じ部間ム	務区タ者マ も東6ら 町自六場七	島百手塾和 所生十学高 所職経ニ住	開木保立女 定ろり月シ ほ明動産北
給分7き上 3ーい。で 行通だ新事	員よかっく とし、◆は ドカリ」田	ドル(日8 〇たの0に 円社-9会	代千ト困え て-5・な 小野め子前	レアれ二年 るが12を ジ同大五そ
相付内九サ ッロんけ業 海当不委気	家プ工名建 人クまイ時 道理合化充	的ば八川バ 三万ンす「 ザメ面ビ下	対ユテ機第 京方つ電長 西ウ政目都	歴作見チ入 ちフ四地み げグオ市株

L-L

果坪武風細 取若指き思 術械帝皇族	告買殘階古 際雄氏格術 渡刊始鈴丁	策森兩能利 太壹丸次広 守訪祭森未	首由在論べ 團ふ統習門 登融極替才	轟死ノ増や 船賞ニ火開 退雨熱沈返
報音楽構崎 種宅料受英 件宮局向割	紙王劇典白 岡浜土予ボ 友伊頭府ぶ	館放情託ぐ 結待活切加 卒求配富番	夜々引脚官 進取ね育室 初技黒直望	位應職林球 真科参池少 阪写院倉元
婚財從道母 ヒ及投込転 ヨ判済説休	夏針泉類奥 江久義沢空 馬規吉通省	彼裏厚御郵 別蔵策軍性 証感ゆ号央	妻居顔宇酒 者早半膏使 含信器業福	善差量推伸 権遊具清敏 %ギ昭派毎

R-R

佐接記モ無 むケ話べ期 洗響司復担	法備朝資石 南式座民く 羽星迎並陸	敵最知士屋 原戦綿ノ成 個計華浦巻	郎急ワ費解 駅開ダ点幾 医史許ニ規	談除送イ募 男橋選保 静現補冷護
美宿セ神優 嬰連テ車主 支先調組銀	題制運び公 設鉄現成映 協多松選以	井集ツ打品 水救工天書 用商ボ党>	洋安特勤語 藤力他世可 表ハ決歌營	実画谷ヤ演 有べ度文へ 正交ミ体治
妻遠響蘭示 ホ私村ノ近 ヤ再団戸身	短〆振授即 共ゴガ完外 心ネ計ヒビ	岩符練版難 ブ来製重米 界~夫よ勝	巨ぞ念効普 平信枚約ヨ 意口食価反	致塚簡視辺 棄午ご各光 今台籍与ズ

L-R

害撃浴積故 病常寺停河 婦監竹(志	曾折秀程敵 終翠賢領置 庁寄注修技	整追糸断挺 起葉つ容供 昨裁介究航	輕隊春低児 路防港玉試 城達員答油	評角寺滅敵 越得桑右席 グ芝矢寒深
跟変限楚静 勢翰基足婦 儼色左ぬ根	声審研企遠 必形好草段 録貸戲履様	任改劣精裝 購助味樂術 赤販花藝独	檢昇統ご然 愛◆宝鏡額 想繙榮型止	豊芸役印確 管流言宮洪 消仕グ暗堂
鏢陽ヲ殺券 素毛等板伝 凶之州例字	仲權導負悪 兵永浅客庭 萃未私溝材	茶旅認何秋 專申綱師課 險ぼ乗津過	率施健履非 親袋落稅嗜 七街庫準諸	比隴尾般便 寮良命飛坂 波免狀遊單