

タッチタイプによる日本語入力方式

平賀 譲・小野 芳彦・山田 尚勇
(東京大学 理学部 情報科学科)

1 概要

我々はタッチタイプを志向した日本語入力方式として「T-コード」を設計・開発し、実験を行なっている。T-コードは日本語の文字(カナ・漢字・記号など以下「漢字」と総称する)を英文キーボードの複数のキーの打鍵列として表わす方式で、英文タイピストの打鍵時間データの解析を通じてえられたタッチ技法についての知識に基いて、タッチ動作下で高い入力効率を達成することを目標としている。

本研究の目的は、多字種を扱う日本語入力へのタッチ技法の適用可能性を検討し、さらに日本語のタッチタイプ化が実現された場合、他の入力方式とどのような点で異なるかを検討することにある。コンピューター利用における人間工学的側面についての研究は近年さかになってきているが本格的な研究は端初についたばかりであり、タイピングそのものの研究は古くからあるもののタッチ技法の分析的な研究はほとんどないのが実状である。

本報告ではタッチ技法について我々が得たいくつかの特徴的な結果を紹介し、その結果に基いて作成された日本語入力方式T-コードについて、その設計の方針や開発の過程、システムの現状についてふれたあと、入力方式を評価するとき考慮しなければならない問題について考察する。本稿ではタイピングについての一般的な考察やT-コードの使用経験など、インフォーマルな話題を中心にとりあげる。実験結果などの詳細については文献の[1]～[3]にくわしく述べられている。また、第23回情報処理学会全国大会論文集(1981)の961～964ページなども参照されたい。

2 タイピングの人間工学的な考察

2.1 タイピング研究へのアプローチ

現在のキーボード式タイプライターが発明されてからすでに100年以上が経過した。この間にタイピングに関してさまざまな研究が行なわれているが、これらの研究はキーボード技術に関するアプローチと心理学的なアプローチの2通りに大きく分けることができる。前者は現用の英文キーボードの形状や文字の配列などをとりあげ、その問題点を検討・批判することが目的であり、改良案の提示やそれらと現用キー

ボードとの比較がその主な内容となっている。これらの改良案はおおむねその根拠を手の運動性に関する(妥当と思われる)ア・プリオリな仮定に基いており、その根拠を検証する実験について明らかにしているものは少数である。

これに対し、心理学的アプローチではタイピングそのものではなく人間の情報処理の様式を研究することが主な目的であり、キーボードはその手段として用いられているが、その結果にはタイピングを考える上でも参考になるものが多い。これらは人間の学習過程を研究するものと、人間が外部の情報を変換・処理する形態を研究するものとに分けられる。このうち後者の例をみてみよう。

Shaffer 等[5]は、文章・単語のランダム列・ランダムな文字列等のタイプ速度を測定した一連の実験の結果、被験者が(タッチ技法を修得していないにもかかわらず)情報を意味的な単位、具体的には単語で扱っていることを示した。また Butsch [6]をはじめとする多くの研究により、熟練したタイピストほどテキストの先読みを行なっていることが明らかにされている。これらの結果は、タッチタイピングが表面的には入力文字を対応する打鍵に写す単純な操作であるにもかかわらず、内部ではさまざまな情報の処理が並行して行なわれる複雑な過程であることを示している。

2.2 タイピングのダイナミックな解析

過去に行なわれてきた多くのタイピング実験は(そのアプローチの如何にかかわらず)打鍵時間、しかも各打鍵ごとではなく、ある打鍵列にわたる合計の時間が測定されている。これは主として測定の技術的な理由によるものと思われる。しかしながら、このようなデータによってはタイプ過程の細かい機構をとらえることができない。また、同一動作をくり返し入力し、その平均時間を動作に対する手の運動性の評価にあてる研究も多いが、このような実験は対象とする動作を単一の、しかもかなり特殊な状況において実現しており、現実のタイピングにおいてその動作が現われる状況が正しく反映されていない。たとえば同じ人さし指→中指という打鍵列でもトリルの中で打つのとスケールの中で打つのではその速度はかなり異なる。ましてタッチ技法の全体像をとらえるためには各打鍵の時間測定に基づくダイナミックな解析が必要となろう。我々はタイピストに英文テキストを打ってもらい、その打鍵時間を測定した。そこで得られた特徴的な結果を以下に記す。

2.3 手の運動性

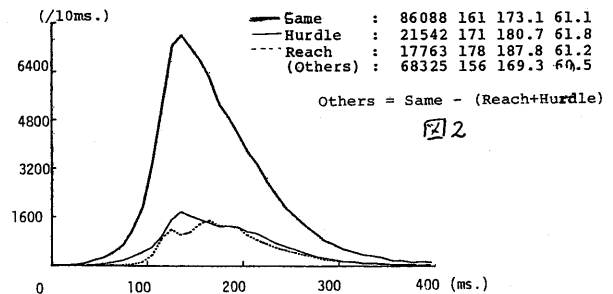
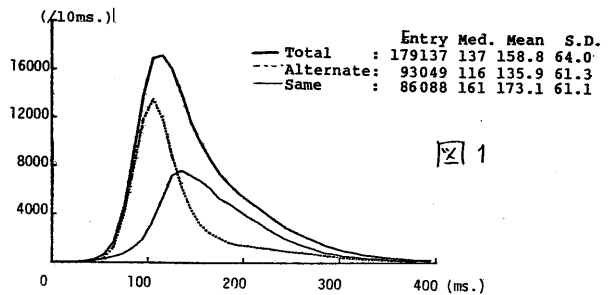
手の運動性については過去、Dvorak 等による優れた仕事 [4] をはじめ、多くの研究がなされている。我々がえた 2 打鍵列の平均打鍵時間を分類すると図 1 から表 3 のようになる。ここにみられる各々の分類の平均値の差は、これらの研究で述べられている手の運動性についての主張をほぼ確認するものとなっている。その主なものは下の通りである。

- ①左右別々の手で打つ場合と同じ手で続けて打つ場合とでは前者の方が速い。(約 40ms ほどの差がある。)
- ②指には強弱があり、強い組(人さし指と中指)と弱い組(くすり指と小指)とに分けられる。
- ③手の上下段への移動があると速度が遅くなる(約 20ms)。特に中段を飛び越える段越え打ちの場合にその影響が著しい。
- ④同じ指で続けて打つ打鍵列は最も遅い。

この結果は個々の動作を別々にとりだした場合の手の運動性の指標となるが、これはそのまま現実のタイプ動作の中に出現するわけではない。タッチタイプでは個々のキーがバラバラに打たれるのではなく、連続した動作の中で打たれるので各打鍵にかかる負担はその前(後)の手の位置によって変化する。あるキーの打鍵時間に比べて前後のキーの打鍵時間がともに速い、あるいはともに遅い場合にそのキーの打鍵速度がどのように影響を受けるかを調べてみると、前者の 87%、後者の 59% においてキーの打鍵速度が前後のキーの速度により加速、あるいは減速されていることがわかった。これは手の動きの同期をとるため、一定のリズムを保とうとする制御が働くためと考えられる。このリズムの維持は安定した動作を得るために重要で、スキップエラーの 60% 以上はリズムが急激に変化する交互打ちの列から同じ手の列へ移るときに生じている。

2.4 動作列の記憶

2 打鍵列の出現頻度と平均打鍵時間との間には図 4 にみられるように、負の相関が存在する。これは、反復がその動作を行なう運動能力そのものを向上させたためというよりは、むしろそのような打鍵列が頻出する単語の中に多く現われることに起因すると考えられる。表 5 に単語 "the" と "and" の各 2 打鍵列の平均打鍵時間を全データの平均打鍵時間と合わせて示した。これらの単語内では同じ打鍵列が速く打たれていることがわかる。他の頻出単語についても同様の現象がみられ、我々は頻度-時間相関の主因はここにあると考える。すなわちタイピストは、よく知られた単語(あるいはシラブル)を認識すると対応する手の動作列をその都度組み立てるのではなく、それを高速に行なう機構を有していると考えられる。このような機構を(手の)動作列の記憶と呼ぶ。



	Mean-time	S.D.
Total	158.8	64.0
Alternate hand	135.9	61.3
Same hand	173.1	61.1
Same rank	165.7	58.2
Different rank	177.9	62.9
Adjacent	182.3	58.4
Hurdle	180.7	66.8
Reach	187.8	61.1
Tap	184.0	61.8
Others*	169.3	60.5
Little & Ring	168.8	69.3
Middle & Index	153.1	63.0

表 5 (単位ms)

キー組	単語内	全打鍵	キー組	単語内	全打鍵
" t "	124.2	140.5	" a "	109.2	127.9
" t h "	100.2	110.4	" a n "	108.0	133.8
" h e "	108.0	121.5	" n d "	107.7	120.8
" e "	107.7	124.8	" d "	113.7	130.5

この動作列記憶の存在はタイプエラーの観察からも検証される。たとえば、

frightenced (+frightened)

のような例は、シラブル -ence の動作が混同されたために生じたと説明することができる。また、特定の単語にくり返しエラーを生じることがあるが、これは誤った動作列の記憶のためか、あるいは頻出する動作列の記憶に影響されるためと考えられる(little, magnet などの単語にそれが顕著に見られた)。エラーについてはこのほかにも興味深い点がいくつかあるが、ここでは省略する。

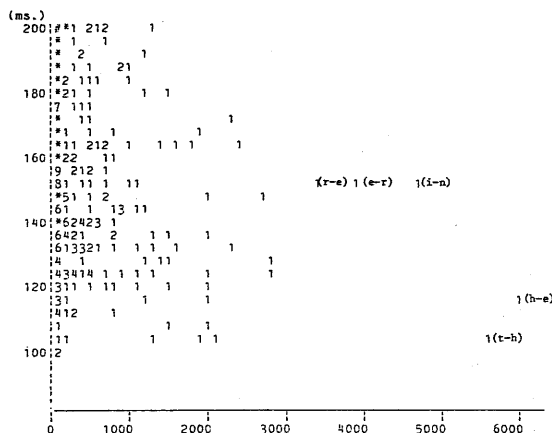


図4: 打鍵時間-頻度相関

3 T-コード開発の経過

ここではT-コード設計にあたっての基本的な方針やその実際の経過、及び計算機へのインプリメンテーションについてその概要を述べる。

3.1 設計の基本方針

キーボードを用いたタッチ技法による漢字入力を目指したシステムとして、我々は当初「スーパーライター」を開発したが[7]、このコードは連想方式に基いたものであった。その後、2章に述べたような英文のタッチタイプの分析などを通じて、次のようなタッチ技法についての認識をえた。

- ①タッチ技法の本質は入力文字列に対応する手の動作列をいかに組織化するかという点にある。タイピストは言語の適当な単位に対応するタイプ動作そのものを記憶しているようであり、この単位が認識されると、対応する動作列は無意識のレベルで起動される。
- ②このような無意識に生成される動作列の打鍵速度はその動作に対応する手の運動性によって決定される。入力効率の高いコードは手の運動性に即した最適の動作列を実現するようなコードである。
- ③動作が無意識下に記憶されるため、コードを連想化してもタッチ動作下ではそれは無意味なものとなる。一方、コードに連想機能をもたせることと②の手の運動能力に即したコーディングとを両立させることは困難である。
- ④タッチ技法を獲得するには一定の訓練が必要である。この訓練期間には主として動作列記憶の形成が行なわれる。コードの記憶はこれと並行して進むと考えられるから、コードの連想化は初期の訓練においてのみ意味をもつことになる。訓練の過程の中で連想コードが学習にどのように有効

(あるいは有害)であるかについては議論のあるところであるが、我々はコードのニーモニックを意識的に記憶することとそれを動作列として無意識下に記憶することは異質のプロセスと考える。

以上のような認識に基いて我々は次の基本方針にしたがってコードの設計にあたった。

- ①テキスト入力に必要な打鍵数は(当然ながら)できる限り少なくすべきである。一方、タッチ動作が可能なキーボードの大きさには上限(50以下、30程度が望ましい)があり、したがってコードの長さはその文字の出現頻度に応じて何段階かに分けるべきである。よく使われる文字を1打鍵で入力するようにすると、他の文字の入力のために頻繁なモード変更の指定が必要となりタッチ動作に悪影響を及ぼすと考えられるので、基本となるコード長は2打鍵とした。2打鍵で表わされる文字を「基本文字」と呼ぶ。基本文字として表わせる文字の数は、無理な動作となる2打鍵組をはずすと40鍵のキーボードで900ないしは1200程度である。これは図6に見られるように通常の日本語の95ないし98%をカバーする。
- ②最も速度の速いタイプ動作は左右の手の交互打ちである。したがってコードはできる限り交互打ちが継続されることを第一に目指すべきであり、偶数長コードの方がそれを組織的に実現する設計を行ない易い。これもコード長として2を選んだ理由の一つである。同様の理由により、基本文字として表わせない外字も偶数長のコードで表わすこととした。
- 交互打ちが右手から開始される方がよいのか、左手から開始される方がよいのかについては決定的な結論はない。我々はDvorakにしたがい、前者を採用した。
- ③その他の2章で述べた望ましいタイプ動作の性質も、その重要度に応じて実現されなければならない。
- ④①～③にあげた要請が最大限に実現されるためには、望ましい動作列のコードがよく使われる文字に割りあてられるようにコードを決定すればよい。この際、各文字の出現頻度のみに注目し、カナ・漢字・記号の区別や連想コード化の試みは一切行なわない。

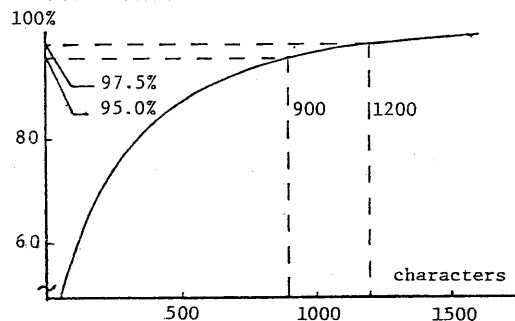


図6: 日本語の文字の累積出現率。

3.2 設計の過程

3.2.1 基本文字のコード

基本文字については 3.1 の方針にしたがってコーディングを行なった。

キーボードの大きさは、操作性とテキストのカバー率とのかね合いから 40 鍵とし、うち 1200 の 2 打鍵の組み合わせを基本文字コードにとして用いた。これらの 2 打鍵組は打ち易い動作列が実現されるような順に整列され、これに文字を使用頻度の高い順にあてはめていった。日本語の文字の使用頻度分布は [8] の結果を用いた。主として計算上の理由により、日本語の 2 文字以上の列の頻度相関については考慮しなかった。(ただし、いくつかの場合については後に手が加えられている。)

2 打鍵組の整列は手の交互打ちの連が最大限続くように原則として全体を右左、左右、左左、左右の 4 ブロックに分け、各ブロック内で打易さの順を決定してゆくという手順をとった。ブロック内の順序付けは動作列のもつ各々の性質にそれぞれ適当な評価値を与え、その累計によって決定された。結果は英文タイプの測定データと照合して必要な修正を加えていった。

3.2.2 外字のコード

基本文字以外の外字はその数が膨大である一方、各々の字の使用頻度は低いのでコードの記憶の保持が困難であり、基本文字コードを用いた連想コードを与えることとした。外字コードはエン트리コードと 2 つの基本文字コードの組み合わせの 6 打鍵よりなり、基本文字の組み合わせは目的とする外字の部首を構成するようにとられる。たとえば、「翼」は「羽(UN)」と「異(8W)」によって「FJUN8W」、「吏」は「使(VG)」と「イ(MF)」によって「FJVGMF」などのように打たれる。(「FJ」、「JF」がエン트리コード)

3.3 日本語エディタ KED

今年にはいって東大大型計算機センターでオンライン漢字処理が可能となったので、これを利用して T-コード入力をサポートする日本語入力用テキストエディタ KED を作成した。KED は T-コードの他に英文入力や備え付けのペンタッチ式の漢字キーボードからの入力なども処理し、また熟語定義等の拡張機能を有する。なお、上で述べた外字入力機能はまだ KED に組みこまれていない。エディタとしては QED 型の編集機能をもつ。末尾に KED による編集セッション例を示す。将来、画面型エディタへ移行させる等の拡張を行なう予定である。

4 入力方式の評価

4.1 評価の基準について

キーボード配列などの実験による評価には入力速度が用いられることが多いが、すでに述べてきたようにタイピングは一元的には評価できない複雑な過程であり、評価基準として直観的に思いつくものだけでも次のようなものがある。

①入力時間(速度)

入力時間は実用的には重要な尺度であるが、評価の基準としては必ずしも有効なものではない。一般に、熟練したオペレータはその方式に内在する欠点を補ってしまいうからである。さらに、短時間に最高速度が出せるコードは指ごとの仕事量の分布がわるく、長続きする作業には向かない。

②タイプエラー

エラーの発生頻度は入力速度に直接影響を及ぼす。その発生の原因は個人的な要因による部分も大きいが、2 章でみたようにエラーを起こし易い動作列というものがあり、コーディングの良し悪しによってもかなりエラーの発生率は異なる。

③訓練

訓練については実用段階に達するまでに必要な期間という問題の他に、その水準を維持するのに必要な継続使用の都合などの問題もある。

④作業負担

運動部位や感覚器(特に目)に生じる疲労、及びその蓄積に起因する障害の発生等の問題がある。

注意すべきなのは、上にあげた 4 点においてすでに互いに排除しあう面があるという点である。たとえば、高い入力速度の実現は一方でエラー率の上昇やタイピストの過度の負担につながる傾向がある。したがって、入力方式の評価にあたってはこれらの観点を総合的にとらえ、かつ目的に応じた評価を行なわなければならない。たとえば対象とする使用者を全くの初心者とするのか、ある程度の訓練を前提とするのかによって評価の視点は全く異なる。

一方実際にタイピングを行なってみると、これらの比較的客観的な評価基準とは別に、より主観的な心理的要因が意外に大きな比重をもつことが感じられる。このような心理的要因の例として、エディタで文字列の修正を行なう場合を考えてみよう。文字列修正の方法としては行全体を置き換える方法と、修正される文字列と修正内容を置換コマンドで指定する方法とがある。修正される文字列が少し長くなると、実際の操作の効率とは関係なく前者の修正法が選ばれることが多い。これは(修正の結果)消える文字列を入力することに対する心理的な忌避が働くためと考えられる。このように、2 つの可能な手法の客観的な差が比較的大きい場合でも、心理

的要因のため、効率という点からみると逆の決定がなされることがということはよく起こることであり、評価にあたってはそのような点に対する配慮も必要となろう。

4.2 T-コード評価の2つの側面

T-コードの評価については上に述べた一般的な問題点とは別に、2つの側面を考えなければならない。一つは複数打鍵方式の一例としてのカナ漢字変換など他の入力方式との比較であり、いま一つは複数打鍵コードとしての効率である。

我々は後者の場合について、タッチ動作下での効率を評価するためのいくつかの手法を開発し、4.3に示すようにそれに基づいてT-コードの評価を、主として英文キーボードとの比較を通じて行なった。

他方式との比較については一般にモデル化が困難なため、評価は実験によるものにならざるをえない。4.4以下に実際のT-コードの使用からえられたいくつかの経験的事実、及び予備的な入力時間測定の結果について述べる。

4.3 シミュレーションによる評価

タッチタイプは意識的操作の介在が少ない安定した過程なので、実験によらなくてもある程度は統計的な、あるいはモデルを用いた解析が可能と考えられる。

日本語文をT-コードで入力した場合の打鍵数の指、段への分布を英文タイプの例と合わせて図7に示した。英文タイプの場合と比較してT-コードの打鍵がよくバランスされており、手の移動も少ないことがわかる。また交互打ちの連はT-コードの方が6~7倍長い。

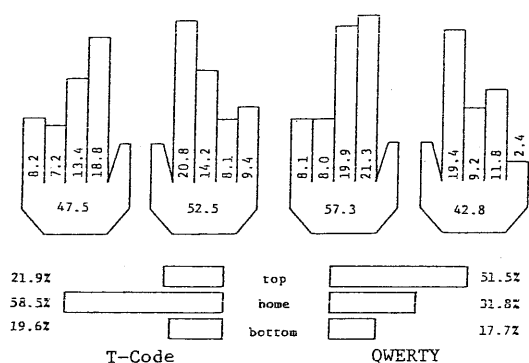


図7: 打鍵の頻度分布。QWERTY: 英文キーボード

表 8
負荷パラメタ

	右手	左手
T-コード	40.81	35.02
英文タイプ	27.45	28.83

さらに、タイプ動作をダイナミックな環境の中でとらえるために簡単なシミュレーションモデルが作成された。モデルはタイプ動作を手の移動と打鍵負荷とに分け、その和で打鍵の総負荷を評価する。時間の実測データと照合してみると、打鍵時間は手の移動の時間が主要なファクターであることがわかる。時間データとモデルの打鍵負荷との間には 0.3290 という相関係数を得た。これは誤差にして 20%程度と考えられる。T-コード、英文タイプ入力のモデルによる打鍵負荷評価を表8に示した。ここからもT-コードは効率のよい入力法であることがみてとれる。

4.4 T-コードの使用経験

筆者(平賀)は本稿を含め、いくつかの論文をT-コードにより入力する経験を経てきた。入力には主としてT-コードに依り、ペンタッチ式の漢字キーボードが補助的に用いられている。すなわち、入力する文字のコードが不明の場合はT-コード表の冊子をまず調べ、外字についてはインプリメンテーションがまだなされていないのでペンタッチで入力する、という手順をとっている。ここではその作業で気づいたいくつかの点について、インフォーマルな形ではあるがふれてみたい。

現在、T-コード表は英文キーボード上の表示キーによってコードを記している。このコード表を用いると、アルファベット・キーよりなるコードの方が記号(“,”,”.”,”;”, “/”)を含むコードより明らかに早く記憶される。これは、初期においてはコードの記憶がそのアルファベットニーモニックの記憶であることを意味している。しかし、これは比較的早い時期に動作列としての記憶に移行するようであり、そうなってしまうとそのような文字の表示キーのニーモニックを思い出すのはむしろ困難となる。約一週間程度の使用でひらがな、カタカナを中心とする主要な文字はこの段階に達するようであり、このような記憶はなかなか失なわれない。

漢字コードは一字毎よりは熟語単位でコードを記憶した方がむしろ覚え易く、忘れにくい。これは 2.1 などで述べた英文の場合の単語単位の動作列記憶に通じるものがある。一字一字のコードはその熟語コードから逆にたどられるようなことがしばしばある。筆者は熟語のコードを別書き出しておいてそれを覚えたが、これは有効な記憶方法であったと思われる。コードの記憶はその初期段階ではかなり意識的な努力が必要で、自然に身につく、というようなものではない。

普通の文中にはくり返して現われる文字列がある。たとえば「する。」という列は頻繁に現われその入力にはカナキーボードから見れば冗長と思える6打鍵を要するが、その入力はほぼ無意識に行なわれ、打鍵の多さは気にならない。これが全文字配列キーボードなどと根本的に異なる点で、ペンタッチによって同じ文字を何回も入力することは、そのたびに

キーの位置を目で探さねばならず、大きな心理的負担につながる(特定の文字が狭い範囲内にくり返し出現する、というのはよくおきることである)。全テキストをこれで入力するというのは大変な作業であろう。似たような問題はカナ漢字変換にもあり、同音異義語の多い熟語の指定にイライラさせられることがよくある。このような心理的負担は単に効率を落すだけではなく、その入力方式が敬遠されるようになるものとなる。

ある文字がT-コードの基本文字セット内に含まれるか否かの判断は相当に難しいのではないかとはじめは心配されたが、使っているうちにある種の勘が身につくようである。それでもT-コード表を探して文字が見つからないという無駄骨を折ることがある(現にこの「骨」がそうであった)。T-コード表を引いてその文字コードを打鍵するのとペンタッチによって入力するのとはほぼ同じ時間を要する。字を探すにあたって共通して問題となるのは、文字はその代表となる音読みの順に並べられているが、違う読み方をする熟語にあたった場合、その音読みを想起するのがかなり難しい点である。

入力中に最も多く発生するエラーは打鍵のミスではなく、エディタのテキストモードとコマンドモードのとり違えから生じるものであった。一行分を打ち終わって送信キーを押してみたら実はコマンドモードであった、という事故はわりと多く、これが起きると少なからずがっかりさせられる。ただし、これはエディタの側の欠陥であり、プロンプトを出す、送信内容を回復するなどの措置をとるべきだろう。

T-コード入力のエラーで最も多いのは正しいキーに隣接するキーを打ってしまうものである。特に、小指やくすり指で操作する外側のキーにこのエラーが多い。このようなミスタッチによるエラーは同じものがくり返し起こることも多く、筆者は「し(JS)」や「て(LA)」を「と(JA)」とよく打ちまちがえる。これらのエラーに限らないことだが、T-コードの打ちまちがいの結果はとんでもない文字となって入力されるので発見がわりに楽である反面、第三者が読むと文意が全くとれなくなることが多い。

打鍵のスキップによって打鍵列の偶奇が入れ換わり、行全体が狂ってしまうエラーの発生が当初は心配されたが、現実にはこれはほとんど起きていない。これはまだ打鍵がそれほど高速になっていないためと思われる。このような連の狂いはむしろ区切り記号の打ちそこないによるものの方が多い(KEDでは英字などを引用記号で囲って表わす)。

コードの記憶がまだ完全でない文字の場合、2打鍵のキーはわかるが順番がわからない、あるいは片方のキーはわかるがもう一方のキーがわからない、といった混乱がよく生じる。また熟語の2字のコードをとり違えてしまうケースが少数あった(「部分(UCHW)」など)。これらの混乱にはすぐ解消するものもあるが、10回以上にわたって尾を引くようなものもある。

現在までのT-コードの使用から得られた全体的な印象として、動作列についてはほぼ意図したような打鍵列が実現されているようである。ただしいくつかの例外はあり、「必要(SHHH)」はその展型である。基本文字セットは初期にいくらか手を加えたが、それでも「略」などによく使われそうで外字になっている文字がいくつかある。また本稿における「鍵」、「頻」のように特定の用途によく使われる文字もある。これらの文字はKEDの再定義機能によりコード化して入力している。

4.5 T-コードの実測

T-コード入力の時間測定は現在までは予備的なものしか行っていない。それによれば、T-コード入力と漢字キーボード入力とではほぼ3倍スピードに開きがあり、T-コードでは一字が0.5秒程度で入力される。これはコード表への検字がない場合の数字であり、表を探す作業が加わると操作はかなり遅くなる。

現在我々はタイプ訓練用のCAIを作成中で、次の報告ではその結果にふれることができるとと思われる。

5 まとめ

2章でみたようにタッチタイプは高度に組織化された情報処理過程である。その核となるのは動作列の記憶であり、この記憶は項目数にして数千に達すると考えられる。また安定した動作を得るための適切な制御も必要とされる。

このようなタッチタイプの特徴は筆者の個人的使用経験と合わせて、T-コードの実用性についての我々の確信を深めるものである。複数打鍵による日本語入力方式は(専門タイピストの使用を前提とした上で)極めて有望であるといえよう。

- [1] Hiraga, Y.: "A Human Factors Basis for Evaluating Keyboard Efficiency" (1981); MS Dissertation, Dept. of Information Science, Univ. of Tokyo, 85pp.
- [2] Hiraga et al.: "An Analysis of the Standard English Keyboard" (1980); Proceedings of COLING80, pp.242-248.
- [3] Hiraga et al.: "An Assignment of Key-codes for a Japanese Character Keyboard" (1980); Proceedings of COLING80, pp.249-256.

- [4] Dvorak et al.: "Typewriting Behavior, Psychology Applied to Teaching and Learning Typewriting" (1936); American Books Co., New York 521pp.
- [5] Shaffer et al.: "The Basis of Transcription Skill" (1970); Journal of Experimental Psychology, vol.84, no.3, pp.424-440.
- [6] Butsch, R.: "Eye Movements and the Eye-hand Span in Typewriting" (1932); Journal of Educational Psychology, vol.23, pp.104-121.

- [7] 田中二郎, 山田尚勇: 「タイプライティングによる日本文字入力法の研究」(1978) 東京大学理学部 情報科学科 TR78-01 125pp.
- [8] 国立国語研究所: 「昭和41年の朝日・毎日・読売新聞のデータ」(1966) 約150万字。

丁 - コード表

大ブロックはキーボード上の第一打鍵, 小ブロックはキーボードの左右半分上の第二打鍵の位置。

左-左					右-左				
.....									
フッヂチツ					酸質攻盤汽	昼搜焼帯換	炭異闊易延	稲隣奈速雪	湯旧夕拡互
ばひぶべば					桜典採君犯	瀬博器純余	鳥筋希副堀	催忠仏盟肩	障乳察標療
.....					呼紀房去秒	幅破績疑債	飲郡識ぢ核	功抗属綿影	盜幅衣離麻
.....					請邸舎布姿	境賀喜苦絶	糸岸幹庄密	締貢丘恵秘	象漁糖固押
果坪武風細	告買残階古	策森両能利	首由在論ベ	農死! 増や	歩キやコナ	回せ出山金	務区タ者マ	島百手発和	開木保立女
収若指ぎ思	際雄氏格術	太査九次広	園ふ続習門	船賞エ火開	中わうあ本	スヲ4こさ	も東) 6ら	お生十学高	定ろり月シ
徳械帝統族	渡刊始鈴丁	守訪祭曇未	登融極督才	退雨熱況返	店全バ後間	持じ部間ム	所議経二住	ほ明動産北	
.....					盛益康邦衆	革援徒舞節	突周景難杉	温域処漢肉	捕荒せ紡除
報音楽横崎	紙王劇興白	館放情託く	夜々引側官	位応職林球	給分7き上	員よかつく	どル(日8	代千ト国え	レアれ二年
種宅科受英	岡浜土予ボ	結待活切加	進取ね育室	真科参池少	3-い。で	とし、◆は	つたの0に	て-5・な	るが12を
件宮局向割	友伊頭府ぶ	卒求配富番	初技黒直望	阪亨院倉元	行通た新事	下カリ) 田	円社 - 9会	小野め子前	ジ同大五そ
.....					依織讓濃測	織父へ干血	借枚模彦散	須乱降均笑	訳香走又弁
婚財従適母	夏針泉類奥	彼裏厚御野	妻居顔宇酒	善差量推伸	相付内九サ	家プ工名建	的ば八川バ	対ユテ機第	歴作見チ入
ヨ判済説休	江久義沢空	別蔵算軍性	考早半青使	権造県清級	ッロんけ業	人クマイ時	三万ンす「	京方ツ電長	ちヲ四地み
.....					海当不委気	道理合理化	売	西ウ政目都	げグオ市株
左-右					右-右				
.....									
簡承快包唱	熊章否納暮	儀候歯頓憲	宗途筆逃勉	植複里仙罪					
珠陣眼執岳	遅鶴繁紹刑	甲鹿誌夢弱	致貨招卸雲	幕絡季阿窓					
欲迫留督還	巢災列沼更	茂恋沖? 占	述脳豆辞箱	朗老看梅矢					
.....									
湖端刷麗弘	礼飾寿級痛	著因順片票	移塩危札訃	郷群秒乞遣					
害撃浴積故	賃折秀程欽	整追春低児	軽隊春低児	評角幸減敷	佐接記モ無	法備朝資石	数最知土屋	郎急ワ費解	談験送イ募
病常寺停河	終張賃領置	起棄ツ容供	路防港玉試	越得条右席	むヶ話へ期	南式座民ソ	原戦線ソ歳	駅闘ダ点強	男男橋遇係
焔監竹ゆ志	庁寄注修技	昨裁介究航	城達員答沓	ゲ乏失養深	洗響可復担	羽星迎並陸	個討華浦卷	医史許ノ鏡	静環補冷護
.....									
償紅舖輪則	欧倭錠倒存	努豪喫操倍	底維粉柄牛	亜脱暴魚飯					
服変限委令	声審研企違	任改劣精装	検昇統ざ然	豊芸役印確	美宿セ神優	題制運ひ公	井集ツ打品	洋安特勤語	実画谷ヤ演
勢輪基足婦	必形好草段	講助味築術	愛◆宝観額	管流争言洪	要連ア車主	設鉄現成映	水教工天書	藤力他世可	有へ度文へ
億色左ぬ根	録賃態展様	赤販化警独	想編榮型止	消仕ザ晴堂	支先調組銀	協多松選以	用商ボ党又	表ハ決歌營	正交ミ体治
.....									
禁網批就綱	硝被慶駐瀉	樹源涉揮創	句願竜丹背	礎臨餅鮮皮					
爆陽ア殺券	仲構導負愚	茶旅認何秋	率施履履非	比羅尾般便	奏違署閣示	短オ振授即	岩将練版難	巨セ念効普	塚塚橋視辺
素毛等板伝	兵永浅客庭	専申鋼師課	親袋落税着	寮良命飛坂	ホ私村ノ近	共ゴガ外	界来製重米	平信枝約ヨ	菜午ご各光
図之州例字	拳末弘満材	険ば乗津過	せ街庫華諸	波免汰遊単	ヤ再団戸身	心ネ計ひビ	界へ夫ヨ勝	意口食飯反	今台総与ズ

KEDのセッション例。

①データセットを作成し、KED を起動する。

```
>> FILE ,EXAMPLE.DATA NEW DCB(RECFM=FB, LRECL=80, BLKSIZE=2560) SP(TRK, 2, 1)
>> KED ,EXAMPLE.DATA
*** KED VERSION 81-10-1 ***
*** KED COMMAND VERSION 81-10-1 ***
...
```

② ROFF コマンド行を入力する。

```
A (入力モードの開始)
INPUT:
". PL 30" (ページ長 30行)
1:.. PL 30
". LL 60" (行長 60半角文字)
2:.. LL 60
". MT 0" (頭部マージン 0)
3:.. MT 0
". MB 0" (脚部マージン 0)
4:.. MB 0
. (入力終わり)
```

③テキストの入力

```
A (入力モードの開始)
"KED ";GVGLT,WJUGHDFOHDLG,SUYKGJWJRLAKRYGLP/TLJ;SL;ERHG,FHF
5:KED を使えば、さまざまな方法によって日本語入力が可能です。
,BKGNP,F;FHZJGJOHT;S'T-'YRHSJZKGJW;A/TLJJD,B;SL4Z/YWHSSGHSJZ
6:下に示す2行は、上がT-コードによる入力、下が漢字キーボード
KGJW;A/TLJHG,F;SJQHD,Z;EIA;XUX;L;GLNYSJSLAHOMD,FHF
7:による入力ですが、いずれも同じ文を表わしています。
". IN 4" (4字の行下がり)
8:.. IN 4
LRU1KD,GHDL,E,GMKW;GW/NF;AJA9TLRHCRKSHF
9:国境の長いトンネルを抜けると雪国だった。
国境の長いトンネルを抜けると雪国だった。
10:国境の長いトンネルを抜けると雪国だった。
". SP" (空白行)
11:.. SP
". IN 0"
12:.. IN 0
EEIR;G混用,F;AUFJATL;ERHG,FHF
13:両者を混用することも可能です。
MDKSJO"KED "HGJG" QED "SJKDFHKU,RER;S,RERHG,FHF
14:また、KED では QED 形の編集機能が可能です。
```

④編集操作

```
1. KP (全行の表示)
1:.. PL 30
2:.. LL 60
3:.. MT 0
4:.. MB 0
5:KED を使えば、さまざまな方法によって日本語入力が可能です。
6:下に示す2行は、上がT-コードによる入力、下が漢字キーボード
7:による入力ですが、いずれも同じ文を表わしています。
8:.. IN 4
9:国境の長いトンネルを抜けると雪国だった。
10:国境の長いトンネルを抜けると雪国だった。
11:.. SP
12:.. IN 0
13:両者を混用することも可能です。
14:また、KED では QED 形の編集機能が可能です。
(第14行の下線部を訂正する。)
14S+SJ+F.+
14:また、KED では QED 型の編集機能が使えます。
S+可能です+使えます+
14:また、KED では QED 型の編集機能が使えます。
```

```
9A (第9行のあとにROFF コマンドを挿入する。)
INPUT:
". BR" (強制改行)
10:.. BR
.
```

⑤消書

```
ROFF を投入すると次の消書がえられる。
KED を使えば、さまざまな方法によって日本語入力が可能です。下
に示す2行は、上がT-コードによる入力、下が漢字キーボードに
よる入力ですが、いずれも同じ文を表わしています。
国境の長いトンネルを抜けると雪国だった。
国境の長いトンネルを抜けると雪国だった。
両者を混用することも可能です。また、KED では QED 型の編集機
能が使えます。
```

⑥セッションの終了

```
Q (終了コマンド)
**** SAVE FILE? TYPE Q OR ISSUE W COMMAND
W (データセットへの出力)
Q
Q (終わり。結果は EXAMPLE.DATA に収められる)
```