練習テキストから上達が計算できる日本文タイプ作業の認知的習得モデル†

小野芳彦††

運動や認知の技能習熟には，速度が練習回数のベキ乗に比例するというベキ法則 (Power Law) が広く認められる．日本文タイプ入力 T コードの練習過程では，練習の小単位である Drill について，その打鍵時間の上達が統計的にベキ法則に従うことが観測された．打鍵の時間を支配していると考えられる要素（文字の読み取り，文字コードの取り出し，手の運動）について，そのいくつかがベキ法則に従って上達するとしたモデルをいくつか作成した．運動の上達のみに焦点をあて，個々の文字の打鍵，打鍵対，文字と文字組，を上達の対象とするそれぞれのモデルでは観測値を近似することができない．しかし，認知的な処理にも焦点をあてて，文字から打鍵列への引き出しをも要素とし，そこに短期記憶の介在を含めると，少ないパラメータで観測値を近似できることがわかった．さらに，そのパラメータのひとつが短期記憶の容量として知られている 7±2 とも一致する．また，このモデルを支持するような個々の打鍵の速度の推移データも観測されている．

1. はじめに

本論文は，練習によるタイプ技能の習熟過程を説明するモデルについて論ずる．一般に，技能の習熟を練習回数と試行時間（あるいは速度）の関係としてとらえると，広い範囲の技能についてベキ法則 (Power Law) と呼ばれる関係が成立していることが実験的に確かめられている[1]．簡単な技能の習得においてだけではなく，複雑な技能の習得においてもベキ法則が成立する理由として，技能の構成要素となっている処理の個々の上達にベキ法則性を仮定するだけで全体の習得のベキ法則性を説明できるとする仮説を立てた．これを支持するデータを，われわれが開発し運用している日本文タイプ入力システムの練習過程の解析から得たので報告する．

ベキ法則が近似できる運動や認知作業は数多い．2数の和を計算するといった非常に単純なものから，タバコ巻き作業や裏文字の文章読み，さらには幾何の証明問題を解くといった非常に複雑なものまで近似できることが報告されている[2]．

タイプ作業の認知処理については，習熟した状態のモデルが Shaffer らによって作られている[3]．研究者によって細部は異なるものの，大略は入力・変換・実行の3つの処理と，間の作業バッファからなる．われわれは，タイプエラーの解析から変換・実行部間のバッファが両手に対応して2つ存在するというモデルをたてている（図 1）[4]．

ここで解析する日本文タイプ入力方式 T コード[5),6)]は，英文キーボードの2打鍵で日本文の1文字を入力するコード方式の入力である．本議論に関与する T コードの特徴は，練習段階の工夫により，読み取り文字列から打鍵列への変換を，文字の意味や文字からの連想を経ないで直接的に空間操作指令に変換するようにしていることである[7]．

タイプ作業においては，変換部が最も複雑な処理を行う．T コード方式では，この処理が1段の長期記憶の検索だけになるように設計しており，またそうなるように練習させている．カナ（あるいはローマ字）漢字変換方式では，文字の読みをカナ（あるいはローマ字）つづりにまず変換し，次に，それを指の操作指令に変換するという2段階の長期記憶検索処理となっている．連想コード方式でも，文字からの連想によるコード引き出しが，少なくとも練習の初期の段階では必要になっている．このため，カナ漢字変換などの変換部の挙動を，入出力だけから解析することは難しいと思われる．カナ漢字変換などの方式によるタイプ作業の練習については，練習の経過の観測報告はいくつかみられるが，練習のモデル化の研究はまだみられない．

英文タイプでは，変換部の入力と出力の対応が，練習の初期では1文字対1打鍵であるのに対し，熟練時にはシラブル対複数打鍵になっていることが観測されている．開始時の変換処理の内容が単純であるため，どのようにシラブル化が進行するかを数理モデルの上で説明することは難しい．このためであろうか，英文のタイプ作業も，練習の経過の観測や静的なモデル化

† A Cognitive Model for a Japanese Type Training in which Speed Advancement is Calculated from Training Texts by YOSHIHIKO ONO (International Research Center for Japanese Studies).

†† 国際日本文化研究センター

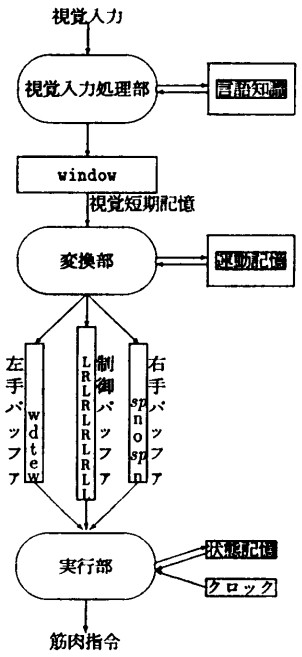

図 1 タイプ作業中の脳の情報処理のパイプラインモデル

Fig. 1 Pipeline model of information processing during typing task.

の研究は多く見られるが，習得のモデル化についての研究は例を見ない．

2. 観測データの要点

2.1 練習コード

われわれは，20人近い被験者について自作の練習システム[8]でのTコードの習得実験を行った[9]．被験者のうち，定期的に練習を行った者の打鍵速度の推移（上昇）を最初の数レッスンについて観察する．

練習コースはレッスンを単位として設計した．レッスンはいくつかのドリルに分かれている．各ドリルを一定回数（具体的には3回）繰り返し練習して，ひとつのレッスンを終了する．全ドリルの平均の打鍵速度が目標に到達したら次のレッスンに進む，というふうにコースを設定している．練習システムでは1打ごとの打鍵時間を計るようなことはせず，ドリル（drill）ごとにその時間を記録している．1打ごとの打鍵時間では細か過ぎてある程度まとまらなければ進歩のようすがわからないからである．

以下では，上達の特性をドリルごとにまとめたものについて見ることにする．

グラフ（**図 2**）は，いく人かの被験者の練習曲線で，縦軸に1打あたりの打鍵時間，横軸に各ドリルの練習回数を両軸ともlogスケールでとった．第一レッスンの8つのドリルを同一のグラフにプロットしてある．ただし，グラフの後尾が急激に落ちている部分は，かなり先のレッスンまで進んでから再度練習した場合の記録である．以下にいくつかの特徴を見て取ることができる．

2.2 ベキ法則

特性(1) 全体として見れば2変数は両対数軸で線形である．

これは運動や認知作業の学習について一般的に知られている関係と一致する．運動の認知作業の学習の進行のベキ法則とは

同一の動作を繰り返し練習すると，動作の実行に要する時間 E が練習回数の累計 t の関数として，

$$E=\beta t^{-\alpha} \qquad (1)$$

で近似できる

というものである．時間と速度のどちらをとってもベキ法則は成り立つのであるが，ここでは，和をとることが容易であり，以後の議論に都合がよい，時間のほうをとることにする．

(1)式の α は上達の早さを表すものであるから上達度と呼ぶ．また，β は第1回目の試行にかかる時間を表すものであるから初期時間と呼ぶ．

われわれの得た練習スコアではこの近似がかなりよく当てはまる．統計計算パッケージ S[10] で，ドリルごとの練習時間の推移を練習回数に対して両対数で線形回帰（linear regression）計算した．回帰の切片（intercept または第0係数）が初期時間 β であり，傾き（gradient または第1係数）の絶対値が上達度 α となる．繰り返し回数が 10 以上のものを対象とした統計解析では，異常に時間が大きい観測値（図で鋭くとがった山のようになっている頂上の点）を欠損値としてほんの少し除けばほとんどが 0.1% で有意な回帰計算となった．

以後の議論では，実際の練習スコアの代表値として両対数の回帰計算で求めた上達度と初期時間を使うことにする．初期時間として実際の第1回目の観測値を使わないのは，特定の測定値ひとつでは誤差の影響が大きいからである．

2.3 記号練習ドリルとその初期時間

特性(2) ある特定のドリルに初期時間が特に小さいもの（実線で示す）がひとつだけある．ほかのドリル（破線で示す）の上達曲線は狭

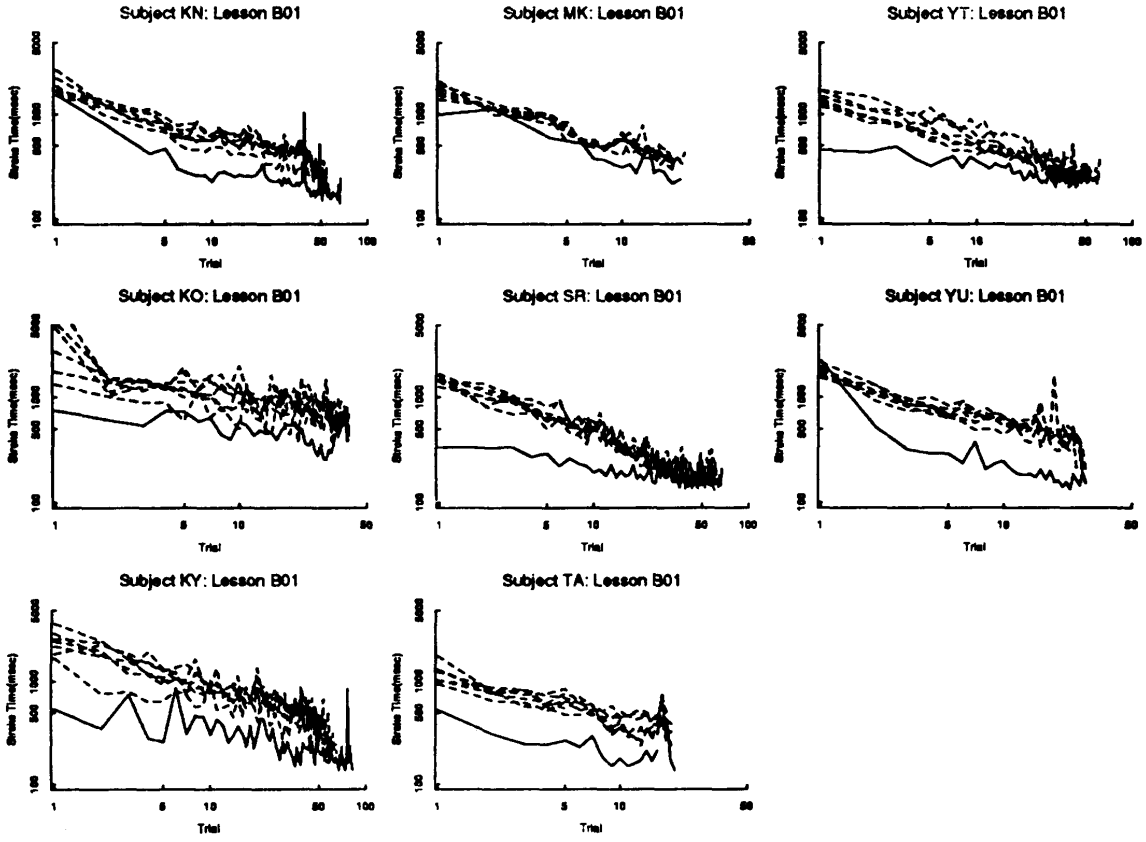

図 2 被験者ごとにプロットしたレッスン B01 のドリルの実行時間の遷移
Fig. 2 Log-log plots of drill stroke time and trials (Lesson B01).

い帯にかたまっている.

これは，第1レッスンに限ったことではない．レッスン B01* と B02 では，ドリル 1.1** の初期時間が他のドリルの初期時間に比べて有意に低い（**図 3**）．**表1**と**表2**に両レッスンの練習文とそのTコード打鍵列を示す.

両レッスンのドリル 1.1 は記号文字列の練習で，他のドリルの練習文が単語・句からなるのに比べて異質である．同様の記号練習ドリルはレッスン B04 のドリル 1.1 を始めとしていくつかあるが，記号練習ドリルのみが際立って初期時間が小さいという傾向は最初のふたつのレッスン以外には見られない（図3）．ただ，両ドリル 1.1 は同一文字の繰り返し回数が他のドリルに比べて大きく，これが，この特性(2)の原因であると考えられる.

2.4 練習の平滑化現象

特性(3) ドリル 1.1 の練習曲線の傾きがほかのドリルと同じであるものと，ほかより明らかに小さいものがある.

ということも図2から読み取れる.

上達度は，両ドリル 1.1 の初期時間に見られるような顕著な特徴を単独では示していない．しかし，全般的に上達度は初期時間と相関が高い．このことは，Y 軸に初期時間 (β) を取り X 軸に傾き (α) を取って各ドリルをプロットすると，$Y=aX+b$ ($a>0$) の直線上に並ぶことから見て取れる（**図4**ドリル 1.1 を含む）．これは，初期時間が小さい（すなわち最初から速い）ドリルほど速くなりかたが小さいということを示している．つまり，練習が進むにつれ，それぞれのドリルの平均打鍵速度が平滑化してほぼ同じになっていくことを示している．これを，**練習の平滑化現象**と呼ぶ.

特異な両ドリル 1.1 が2つのタイプになって図4に現れるのが特徴(3)である．ひとつは，他のドリルが

* 以後レッスンは名前で呼ぶことにする．名前は開発の経過から B*nn* か C*nn* となっている．また，数字は 01, 02 と増える順に練習させるが，これも開発の経過でレッスンの分割・統合・取替などを行ったため，025 などの挿入番号や欠番がある.

** ドリルは番号で呼ぶ．ドリルの番号は「1.1」のように「主.副」になっている．主番号が同じドリルは練習する目的文字が同じである．副番号はその連番である.

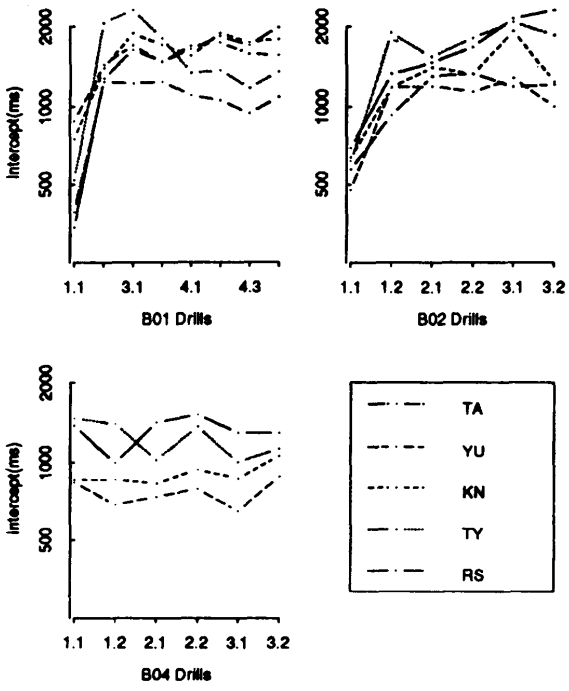

図 3 レッスン B01, B02, B04 のドリルの初期時間
Fig. 3 Intercept scores of subjects in B01, B02, and B04.

並ぶ直線から下左方向に大きく外れるタイプ（被験者 YU と TA）であり，これらは平滑化を起こしていない．もうひとつは同じ直線上にのるタイプ（被験者 RS と TY），すなわち，平滑化を起こすタイプである．ただし，同一の被験者（KN）に平滑化があったりなかったりすることも観測されている．

平滑化現象にはいくつかの解釈が可能である．まず，上達度が本質的に異なると率直に考えるのは妥当でない．最初遅かったほうが速かったほうを追い越すという現象があるとは考えにくいからである．この形は $E=\beta(t+c)^{-\alpha}$ の式に従うとした場合のプロットと同様の傾向を示す．つまり，なんらかの理由によって，初めから技能が一足飛びに進歩していたとの解釈が可能である．

もうひとつの解釈は，速い打鍵が他の遅い打鍵に引きずられて遅くなる，つまり，各打鍵をできるだけ同じ速度にしようとする（多分心理的であろう）抑制が働くというものである．2つのドリル 1.1 に平滑化が現れたり現れなかったりするのを説明するには，この解釈のほうがふさわしい．言い替えれば，平滑化がないのは指の運動能力の最大練習効果が出ている型であり，本来の上達曲線と考えられる．

表 1 レッスン B01 のテキスト
Table 1 Text of Lesson B01.

番号	ドリル名	テキスト
1	B01.1.1	、。、。、。、。、。 、、。、。、、。、、。。
2	B01.2.1	の、が、で、は、 ので、のは、のが、では、
3	B01.3.1	していると、していると、
4	B01.3.2	しているのが、いるので、 としている。
5	B01.4.1	なにを、ない、したのに、 でした。としたのでは、
6	B01.4.2	には、している。したい。 していないのではない。
7	B01.4.3	では、なにをしている、となる。
8	B01.5.1	はるになると、なのはながたのしい。

表 2 レッスン B02 のテキスト
Table 2 Text of Lesson B02.

番号	ドリル名	テキスト
1	B02.1.1	() () () () () (() () ()) (())
2	B02.1.2	(て) (に) (を) (は) して（しないで）、
3	B02.2.1	この、もの、どの、おのおの、 おる。こる。たどる。ともる。
4	B02.2.2	おもてでこどもがおどる。 どこにもおもいものはない。
5	B02.3.1	とらない。とる。とれ。 とられた、これらの、
6	B02.3.2	どれもしらないので、 これをとられてはいられない。

2.5 仮　　説

上に述べたように，Tコードの練習については一般にベキ法則が成立している．1ドリルの練習文全体を練習する動作の基本的単位とみなしたのでは練習による修得の過程を解析するのには複雑すぎる．1ドリルは複数の打鍵作業から成り立っており，さらに，その打鍵作業も同じ文字を打つのではなく異なった文字を異なった回数打鍵するものである．われわれは，1ドリルを分解したもっと細かい打鍵作業のそれぞれがベキ法則で上達するだけで，複雑な練習過程に対してもベキ法則性が成立すると予想した．予想が正しく，小さな単位が文字やキー位置に基づくものならば，それを用いて練習文の効率や習得の個人的特性を議論できることが期待できる．

まず，この仮説に基づくと，どのくらい実際の習熟曲線を定性的に説明できるかを議論する．

特性(2)が示すことは，習熟曲線をベキ法則で近似したときのパラメータ α と β が特異な第1ドリルを除

いてみなほぼ同じであるということである．このことは特徴として注目すべきことである．

まず，ほとんどのドリルの上達度αがほぼ一定であることは被験者の打鍵の基本作業の上達度がαとほぼ同じであることを示唆している．同時にどのドリルも同じ上達度の基本作業からなることを示唆しており，そのために上達度がそろうと考えられる．

また，ドリルの上達度がほぼ同じであるとするとドリルの難易は初期時間βだけによることになる．われわれは，指の運動の易しい順に文字を導入するという方針でレッスンを設計したので，初期のレッスンの文字の打ちやすさはほとんど同じである．もし練習テキストに現れる文字の練習の上達のみがドリル全体の練習の上達を説明できるならば，ドリルごとの難易がほぼそろうということがあっても不思議ではない．ただし，テキストをなるべく自然な表記の文にするという制約を課しているため，テキストに出現する文字の数をそろえるといった制御は行っていない．したがって，テキストの文字レベルでドリルの難易を論ずるとすれば，実際のテキストによる定量的な議論が必要となる．このような議論は次章で行う．

しかし，たとえテキストの文字レベルでドリルの難易が実際に異なっていたとしても，特性(3)の1解釈に示したように，打鍵速度をそろえようとする抑制の結果であるということもありうる．

以上のように，一部の例外的ドリルを除いて，何か基本的な作業，それも，たぶん練習テキストを構成する文字の打鍵がベキ法則に従って上達すると仮定するだけで各ドリルの上達を説明できる可能性を観測値は示している．

3. T コードの修得モデル

前章の議論で，テキストの文字構成からベキ法則を使っての習熟の過程が説明できる可能性を明らかにした．本章では，基本習熟単位を探るモデルを立てて，現実の測定データの定量的性質を説明することを試みる．ここでは，観測値からの単純な誤差計算でモデルの優劣を論ずるのではなく，前章で指摘した最初の2つのレッスンのドリル1.1の特異性が再現するかどうかを第1条件としてモデルの適否を判断することにする．

3.1 文字モデル

われわれは単純なモデルから始めて，その適合しない点を説明する機構を加えて改善し，妥当なモデルを構築するという方略を採用した．最も単純なモデルは個々の純粋な打鍵のみが上達の対象となるものとするものである．しかし，このモデルでは，キーボードのキーの数（われわれの場合40）だけの打鍵練習をして習熟すれば，練習したことのない文字でも高速に打鍵できることになってしまう．これは明らかに現実とは異なっており，最低限でも個々の文字コード（すなわち2打鍵組）が上達の対象であると考えなければならない．

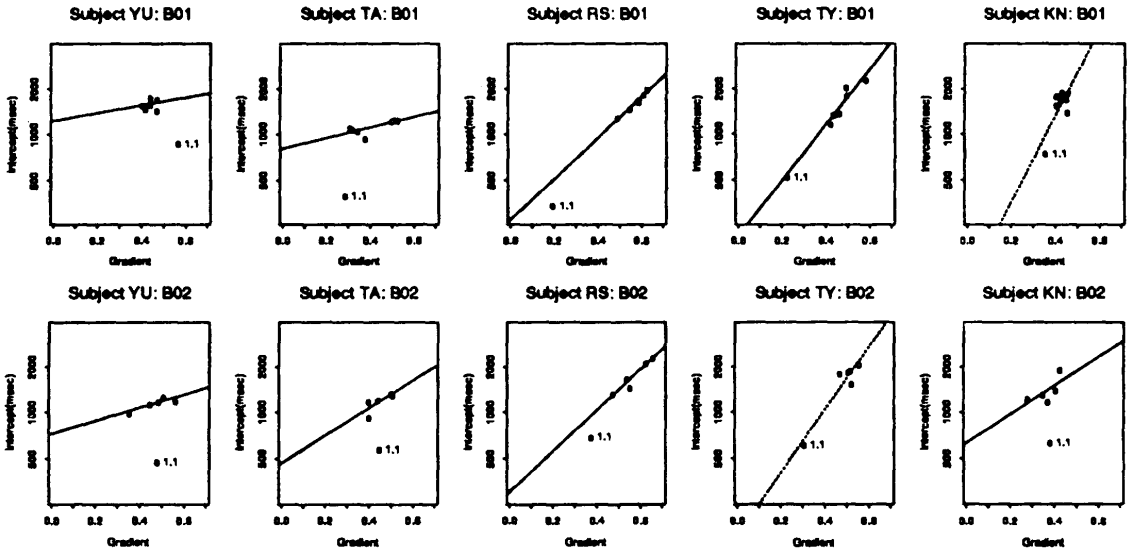

図 4 上達度と初期時間のドリルごとの相関
Fig. 4 Correlation between intercept scores and gradient scores.

3.1.1 文字モデルの基本設定

文字モデルとは，コードの学習が文字コードの打鍵のみに律せられるというモデルである．文字 ch の打鍵時間 E^{ch} は，この文字の過去の総練習回数 t の $-\alpha$ 乗 ($\alpha>0$) に比例するとみなすベキ法則を採用する．すなわち，

$$E^{ch}(t)=\beta t^{-\alpha} \qquad (2)$$

で計算できるとする．後のモデルとの統一を取るため，時間 $E^{ch}(t)$ は文字（2打鍵）を打つ時間ではなく打鍵単位に平均したものと定義する．

α と β が個々の文字コードで異なるとするモデルも考えられるが，ここではすべての文字コードについて同一であるとする．ここで適用する練習文においては，出現する文字のコード（打鍵組）に極端な難易の差がないことがこの仮定の妥当性を支持している．

このモデルで現実のテキストをシミュレートさせるには，レッスンやドリルをどのような順で何回ずつ練習するかのパラメータが必要である．ここでは標準設定のコースに従って各レッスン i を N_i 回練習したとする．打鍵時間の計算に必要なレッスン l ドリル d での文字 ch の出現数 $\delta^{ch}_{l,d}$ と，そのレッスンでの文字ごとの集計 σ^{ch}_l を，レッスン B01 と B02 についてのみ**表3**と**表4**の $\lambda=0$ の行に示した．以下ではレッスンの進み方を厳格に守って練習すると仮定して計算を進める．実際には先のレッスンを少しだけ練習したりとか，いくつかレッスンを戻ったりとかいう部分的な重なりがあるのであるが，順序の入れ替えで練習回数の累計には変化がなく，これらを無視しても大きな誤差は生じないことを確認している．

標準設定のコースに従った場合，レッスン l の j 回目の練習での d 番目のドリルが終った時点では各文字を $f^{ch}(j,l,d)$ 回練習している．ただし，

$$f^{ch}(j,l,d)=\sum_{i=1}^{l-1} mN_i\sigma^{ch}_i+(j-1)\sigma^{ch}_l+\sum_{k=1}^{d}\delta^{ch}_{l,k}$$

で計算し，m はこの場合3である．この式の第1項はレッスン l 以前に練習した分で，第2項はレッスン全体の $j-1$ 回の繰り返し，第3項は j 回目の練習のドリル1から d までの累計である．各ドリルでは $f(j,l,d-1)+1$ 回目から $f(j,l,d)$ 回目まで各文字を練習するから，平均打鍵時間は(3)式で計算できる．

$$E_{l,d}(j)=\frac{1}{n_{l,d}m_{ch}}\sum_{ch}\sum_{t=f^{ch}(j,l,d-1)+1}^{f^{ch}(j,l,d)}\beta t^{-\alpha} \qquad (3)$$

ここで，$n_{l,d}$ はレッスン l ドリル d のテキストの文字総数である．また，$f^{ch}(j,l,0)$ とは，このレッスンのドリル数を D_l とすれば，$f^{ch}(j-1,l,D_l)$ のことである．$f^{ch}(1,l,0)$ は $f^{ch}(3,l-1,D_{l-1})$ である．

表3 練習文字ごとの長期記憶から検索される回数（レッスン B01）
Table 3 The number of LTM-fetching per character (Lesson B01).

d(ドリル)	λ	$\delta^{ch}_d(\lambda)$
文字 ch		、 。 のがでははしていることなにをた
コード		jd hfkd ;shg jg js lahd ;aja lgkg ;gks
1	0	12 10
	1	9 9
(1.1)	2	2 2
	3	2 2
2	0	8 4 2 3 3
	1	8 4 2 3 3
(2.1)	2	5 4 2 3 3
	3	2 2 2 3 3
3	0	2 2 2 2 2 2
	1	2 2 2 2 2 2
(3.1)	2	2 2 2 2 2 2
	3	2 2 2 2 2 2
4	0	2 1 2 1 1 2 2 3 3 1
	1	2 1 2 1 1 2 2 3 3 1
(3.2)	2	2 1 2 1 1 2 2 3 3 1
	3	2 1 2 1 1 2 2 3 3 1
5	0	4 1 2 2 1 3 1 1 2 2 1 3
	1	4 1 2 2 1 3 1 1 2 2 1 3
(4.1)	2	4 1 2 2 1 3 1 1 2 2 1 3
	3	3 1 2 2 1 3 1 1 2 2 1 3
6	0	1 3 1 1 2 3 2 5 1 2 1 1
	1	1 3 1 1 2 3 2 5 1 2 1 1
(4.2)	2	1 3 1 1 2 3 2 4 1 2 1 1
	3	1 3 1 1 2 3 2 4 1 2 1 1
7	0	2 1 1 1 1 1 1 2 1 2 1 1
	1	2 1 1 1 1 1 1 2 1 2 1 1
(4.3)	2	2 1 1 1 1 1 1 2 1 2 1 1
	3	2 1 1 1 1 1 1 2 1 2 1 1
8	0	1 1 2 1 2 1 1 2 1 3 1 1
	1	1 1 2 1 2 1 1 2 1 3 1 1
(5.1)	2	1 1 2 1 2 1 1 2 1 3 1 1
	3	1 1 2 1 2 1 1 1 1 2 1 1
$\sigma^{ch}(\lambda)$	0	32 17 11 4 8 9 12 7 13 10 6 9 5 2 5
	1	29 16 11 4 8 9 12 7 13 10 6 9 5 2 5
	2	19 9 11 4 8 9 12 7 12 10 6 9 5 2 5
	3	15 9 9 4 8 9 12 7 12 9 6 8 5 2 5

$\lambda=0$ の行は文字出現の単なる累計となる．

3.1.2 文字モデルの適否

B01 と B02 の初期時間について適否を調べる．**図5**の点cは，文字モデルのシミュレーション結果を練習の軌跡とみなし，両対数型線形回帰で求めた切片（初期時間）をドリルごとにプロットしたものである．図を導いたパラメータは $\beta=2,000$ ms, $N_1=20$, $N_2=20$, $\alpha=0.3$ である．初期時間は B01 の全ドリルで似たような値をとっており，ドリル 1.1 の値が特に低くない．B02 においてはドリル 1.1 の初期時間がほかのドリルより大きいという実際とは逆の結果を示している．表2に示したレッスン B02 のテキストから明

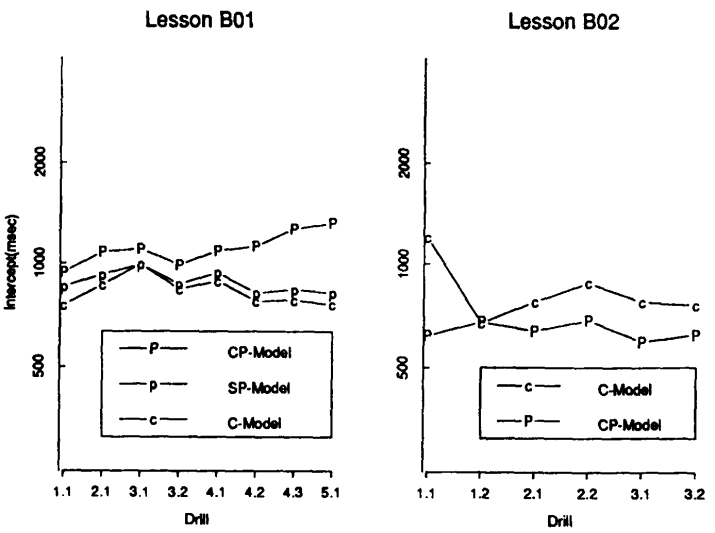

図 5 文字モデル，打鍵対モデル，文字組モデルのシミュレーション結果
Fig. 5 Simulation results of Character, Stroke-Pair, and Character-Pair Models.

表 4 練習文字ごとの長期記憶から検索される回数（レッスン B02）
Table 4 The number of LTM-fetching per character (Lesson B02).

d(ドリル)	λ	新しく習う文字の δ_d^{ch}																既に習った文字の δ_d^{ch}							
ch		、	。	の	が	で	は	し	て	い	る	と	な	に	を	た		（	）	こ	も	ど	お	ら	れ
code		jd	hf	kd	;s	hg	jg	js	la	hd	;a	ja	lg	kg	;g	ks		ke	id	uf	ia	kq	oa	ig	;e
1	0																	11	11						
	1																	9	9						
(1.1)	2																	2	2						
	3																	2	2						
2	0	1					1	1	2	2	1			1	1	1		5	5						
	1	1					1	1	2	2	1			1	1	1		5	5						
(1.2)	2	1					1	1	2	2	1			1	1	1		5	5						
	3	1					1	1	1	2	1			1	1	1		2	2						
3	0	4	4	5								4	1				1			2	2	2	3		
	1	4	4	5								4	1				1			2	2	2	3		
(2.1)	2	4	4	4								4	1				1			2	2	2	2		
	3	4	3	1								3	1				1			2	2	2	2		
4	0		2	1	1	1	1		1	2	1		1	1						2	5	3	3		
	1		2	1	1	1	1		1	2	1		1	1						2	5	3	3		
(2.2)	2		2	1	1	1	1		1	2	1		1	1						2	5	3	3		
	3		2	1	1	1	1		1	2	1		1	1						2	5	3	3		
5	0	2	3	1								1	1	4	1		1			1				3	3
	1	2	3	1								1	1	4	1		1			1				3	3
(3.1)	2	2	3	1								1	1	4	1		1			1				3	3
	3	2	3	1								1	1	3	1		1			1				3	3
6	0	1	1	1		1	1	1	1	3			1	2		1				1	1	1		3	4
	1	1	1	1		1	1	1	1	3			1	2		1				1	1	1		3	4
(3.2)	2	1	1	1		1	1	1	1	3			1	2		1				1	1	1		3	4
	3	1	1	1		1	1	1	1	3			1	2		1				1	1	1		3	4
σ	0	8	10	8	1	3	3	3	4	7	6	6	5	2	2	2		16	16	6	1	6	6	6	7
	1	8	10	8	1	3	3	3	4	7	6	6	5	2	2	2		14	14	6	1	6	6	6	7
	2	8	10	7	1	3	3	3	4	7	6	6	5	2	2	2		7	7	6	6	6	5	6	7
	3	5	7	4	1	3	3	2	4	7	5	5	5	2	2	2		4	4	6	6	6	5	6	7

$\lambda=0$ の行は文字出現の単なる累計となる．

らかなように，ドリル 1.1 以外にはレッスン B01 ですでに練習した文字が含まれている．モデルが文字の練習による上達のみに基づくものであるから，ドリル 1.1 の第 1 回目の平均打鍵時間がほかのドリルのものより大きくなるのは当然である．初期時間が第 1 回目の平均打鍵時間を大きく下回る要素を入れない限り，文字モデルで実際のデータを説明することは不可能である．

図を導いたパラメータをいくつか独立に変化させても上の傾向は変化しない．これは，文字モデルが実際と合っていないということを意味する．

3.2 打鍵対モデル

既習の文字を含まないのに初期時間が小さくなるという事実を説明するためには，

（a） 既習の文字がなんらかの原因（仕組）で既習とならないで，未習と同じスタートをする．

（b） 2 つのドリル 1.1 に急速に上達する仕組がある．

のどちらか，あるいは両方の要素をもつモデルを考える心要がある．

3.2.1 打鍵対モデルの基本設定

打鍵対モデルとは，文字の打鍵ではなく実際の打鍵の 2 つ組おのおのが習熟を律すると考えるものである．文字モデルで習熟の単位とした打鍵組を文字内だけでなく文字間も含めたうえで，文字モデルと同様の仕組で計算を行う．ここで導入した文字間の打鍵組は先の条件（a）を満たす．すなわち，前のレッスンで既習となっていた文字でも，文字間の打鍵組はほとんどが未習である．それに対して，単調な繰り返しの練習文である両ドリル 1.1 では，文字間の打鍵組が同じものとなり，習熟が早く進む．つまり，先の条件（b）を満たすものであると期待される．

ドリルの平均打鍵時間 $E_{l,d}(j)$ は（3）式の文字 ch に対する集計をすべての打鍵組 u,v に対する集計に置き換えればよく，

$$E_{l,d}(j)=\frac{1}{m(2n_{l,d}+s_{l,d})}\cdot \sum_{u,v}\sum_{t=f^{u,v}(j,l,d-1)+1}^{f^{u,v}(j,l,d)}\beta t^{-\alpha} \quad (4)$$

となる.

3.2.2 打鍵対モデルの適否

パラメータは，基本打鍵組の習熟パラメータ β と α，および，レッスンの反復回数 N_i であり，文字打鍵モデルと同じである．このモデルを B01 についてシミュレートした結果を図5に点pでプロットする．予想に反して，打鍵対モデルではドリル1.1の値が小さくならず，文字モデルとあまり変わらない結果しか得られない．

3.3 文字組モデル

このように単純な打鍵組モデルはまだ効果が少ない．考慮が不足していたと考えられる点として次の2つが考えられる．

（c） 文字頭の打鍵時間が文字中と同じであるとする仮定は実状と合っていない．したがって，パラメータが文字頭（α_0 と β_0）と文字中（α_1 と β_1）とで異なるものとする．β_0 のほうが大きな値を持つとして実際の観測値を説明できればよい．

（d） 文字間の打鍵組というのは被験者が直接意識しないのに，それを基本習熟単位としたことは合理的でない．直接意識される文字組（2文字組）がもうひとつの基本習熟単位になっていると考える．

3.3.1 文字組モデルの基本設定

ここでは，打鍵対モデルにこの両方の修正を加える．すなわち，文字中打鍵組は従来の文字打鍵モデルと同様の習熟をし，文字頭打鍵組はその文字組として習熟するとみなす．後者は，同じ打鍵組であっても文字組が異なれば別の経路で習熟が行われると考えるもので，もはや指の運動の習熟とはみなしていないことに注意してほしい．

ドリルごとの平均打鍵時間は

$$E_{t,d}(j) = \frac{1}{m(2n_{t,d}+s_{t,d})}\left(\sum_{ch}\sum_{t=f^{ch}(j,l,d-1)+1}^{f^{ch}(j,l,d)}\beta_0 t^{-\alpha_0} + \sum_{ch,ch'}\sum_{t=g^{ch,ch'}(j,l,d-1)+1}^{g^{ch,ch'}(j,l,d)}\beta_1 t^{-\alpha_1}\right) \quad (5)$$

で計算する．ここで，$g^{ch,ch'}$ は文字組のドリルごとの数え上げ $\delta^{ch,ch'}$ から f^{ch} と同様の方法で累計したものである．$\delta^{ch,ch'}$ は非常にスパースで，1レッスンについてそれを表示するだけでもページ数をくってしまうので，ここでは示さない．

3.3.2 文字組モデルの適否

文字組モデルの計算結果を図5に点Pで示す．

レッスン B01 での変化（図省略）は，$\beta_0<\beta_1$ とするほどドリル1.1の値が相対的に小さくなる．しかし，これでは観測事実と逆であり，したがって，現象を説明するのは他のパラメータであるか，モデルが正しくないかのどちらかである．

レッスン B02 では逆に $\beta_0>\beta_1$ とするほどドリル1.1の値が相対的に小さくなる（図省略）．さらに，ほかのどのドリルより小さいというわけにはいかないが，ドリル1.2より小さくなるという現象が，曲がりなりにもシミュレートできたのは B02 では初めてである．レッスン B01 での β の逆方向性に目をつむれば β を2つに分けたことは評価できると考えられる．

3.4 コードキャッシュ・モデル

文字モデルから打鍵組モデルまでは，打鍵や打鍵組の頻度の影響が打鍵速度を律するという考えであった．しかし，それではドリル B02.1.1 の特異性がまったく再現できないことが判明した．文字組モデルでは，文字頭の打鍵を運動の習熟ではなく一種の認知的な習熟であるとしており，わずかではあるがレッスン B02 をシミュレートできた．

以下に述べるコードキャッシュ（CC）モデルは，文字組モデルにならって認知的な習熟を最初から取り入れる．まず，最初の文字打鍵モデルに帰って，文字中の打鍵も文字頭の打鍵も文字コードが律すると考える．ただし，文字頭の打鍵にはコードを引き出してくるための認知的な作業が存在するとし，その分の時間を練習によって独自に習熟させると考える．ただそれだけでは，文字中と文字頭で習熟のパラメータが異なるとした文字組モデルと本質的な差が生じない．ここでは，コードの一時的記憶がコードの引き出しと関係するとして，ドリル1.1の特異性の再現を確かめる．

3.4.1 CC モデルの基本設定

コードキャッシュモデルでは，1文字の打鍵は，(Ⅰ)文字コードの引き出し，(Ⅱ)文字頭打鍵，(Ⅲ)文字中打鍵の3つの段階が直列に出力に現れるとする．すなわち，文字頭の打鍵時間は，(Ⅰ)にかかる時間と(Ⅱ)の打鍵時間の和が観測されるとする．それぞれの文字の打鍵で(Ⅱ)と(Ⅲ)は同じ時間と仮定する．さらに，コードの引き出しと2つの打鍵は独立にベキ法則に従って上達するとし，パラメータを(Ⅰ)は α_t, β_t, (Ⅱ)および(Ⅲ)は α_s, β_s とする．

ここまででは以前のモデルと本質的に同じになってしまう．これにコード引き出しのプロセスにおける次の仮定を加える．

- コードの引き出しは2種類あるとする．コード記

憶は通常は長期記憶から検索される．検索されたコード記憶は短期記憶に転送され運動指令に変換される．この変換後もコード記憶は短期記憶にしばらくの間は保持されると考える．したがって，短期記憶がコードを保持している場合には，長期記憶からの検索は行われず，その分(I)が小さくなると考えられる．ここでは近似的に0になるとする．

- コード記憶の引き出し過程の習熟は長期記憶が検索されるたびに起きるとする．逆にいうと，短期記憶に保持されたコードが再利用される場合には，コード記憶の検索の習熟が起きないとみなす．

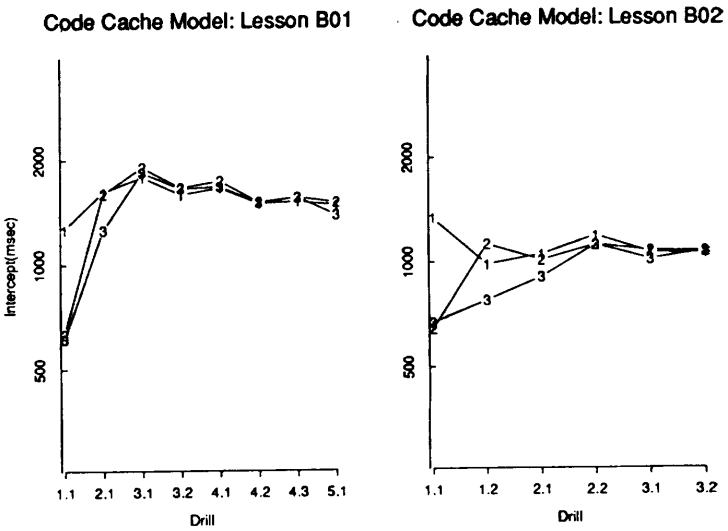

図 6 コードキャッシュモデルのシミュレーション結果
Fig. 6 Simulation results of Code-Cache Model.

短期記憶は容量が小さいという制限がある．そのため，多種のコードが混ざっている練習文の場合には長期記憶からの検索が頻繁に起こるであろうが，数少ないコードからなる練習文の場合には，前に検索されたコードを再利用する傾向が高いであろう．練習の初期段階ではコードの引き出しが文字頭の打鍵時間の大半を占めているから，この差が両ドリル1.1の特異性を生むと考えるのである．

大きさの制限付きの短期記憶を λ 文字分の大きさのキャッシュ・メモリと考えよう．キャッシュ上のコードの保持方式として，MRU (Most Recent Usage) をとる．すなわち，λ 文字分のコードがすでにキャッシュ上に保持されている状態で，新しい文字のコードを保持することになった場合，最も古い文字のコードを記憶からあふれさせるということである．

このアルゴリズムに従って練習文をスキャンすれば，個々の文字コードが長期記憶から検索されるかどうかがわかる．たとえば，ドリル B01.2.1 の1行目の練習文では，$\lambda \geq 2$ ならば「、」は1回しか検索されない．練習文でコードが長期記憶から検索される回数の個々のドリルについての累計 $\delta^{ch}(\lambda)$ とレッスンでの累計 $\sigma^{ch}(\lambda)$ を表3と4に掲げる．

ドリルごとの平均打鍵時間は

$$E^{\lambda}_{l,d}(j)=\frac{1}{2mn_{l,d}}\sum_{ch}\left(\sum_{t=h_{\lambda}^{ch}(j,l,d-1)+1}^{h_{\lambda}^{ch}(j,l,d)}\beta_f t^{\alpha_f}+\sum_{t=f^{ch}(j,l,d-1)+1}^{f^{ch}(j,l,d)}2\beta_s t^{\alpha_s}\right) \quad (6)$$

で計算する．ここで，f^{ch} は以前のモデルと同じで $\lambda=0$ とした $\delta^{ch}(0)$ の累計から求める．h^{ch}_{λ} は f^{ch} と同様の方法で $\delta^{ch}(\lambda)$ を累計したものである．

3.4.2 CC モデルの性質と適合性

λ を 1, 2, 3 の3通りについてシミュレートする(**図6**)．試したパラメータは基本として (β_f, α_f): (β_s, α_s)=(2s, 0.3) : (0.5s, 0.1) である．ここには示していないが，パラメータによる変化を見るためにそれぞれを変化させた (3s, 0.3) : (0.5s, 0.1)，(2s, 0.4) : (0.5s, 0.1)，(2s, 0.3) : (1s, 0.1)，(2s, 0.3) : (0.5s, 0.3) について調べた．

$\lambda=1$

B01 ドリル1.1の初期時間は他のドリルよりわずかに小さい．

B02 ドリル1.1の初期時間は他のドリルより小さくはならない．

$\lambda=3$

B01 ドリル1.1の初期時間が他のドリルより小さい．$\lambda=1$ の場合とは異なり，ドリル1.1と他との差はかなりある．

B02 ドリル1.1の初期時間が他のドリルより小さいかどうかはパラメータによる．しかし，際立って小さいということもない．パラメータが適当な場合には，B02.1.2 との差があまり大きくないことを除けば，まず適合しているといってよい．一部の被験者はこのようなB02.1.2 も多少低くなっているような分布を示している．

$\lambda=2$

B01 ドリル1.1の初期時間の他のドリルとの差は，大きくかつ際立っている．

B02 ドリル1.1の他のドリルとの差は，レッスン B01 ほどではないが大きくかつ際立っている．

さらに，ドリル B04.1.1の初期時間は特異に低くはない．また，初期時間と上達度の相関図（**図 7**）も，平滑化効果を起こさない上達を示している．

シミュレーションの結果では，$\lambda=2$ のコードキャッシュモデルが実測データをうまく説明している．

文字の打鍵コードが短期記憶の中でいくつのチャンクを必要とするかを考える．それぞれの打鍵がキーボード上の2次元座標に対応することから，また，2次元空間上にコードを表示する練習方法をとっていること[7]から，両座標軸の値，すなわち，段と列の情報がそれぞれチャンクとなっていると考えるのは合理的である．段は3段*，列は5列であり，通常の1チャンクが持つ情報量として適量である．ここでは，手の左右の情報を勘定に入れていないが，練習の初期には左右の順で打鍵するものしか練習しないので，特に打鍵ごとに独立したチャンクとして持つとしなくても問題はない．すると，1打鍵あたり2チャンク，1文字あたり4チャンクとみなせる．短期記憶の容量に関して，一般的にチャンク数は 7 ± 2（この場合の文字数に換算すると 1.75 ± 0.5）であるという知見[11]が得られている．われわれのモデルでのシミュレーションが，短期記憶の容量を文字数にして2とした場合でのみ測定データを再現できることは，モデルが心理的実在性の高い方向に進んでいることを示唆している．

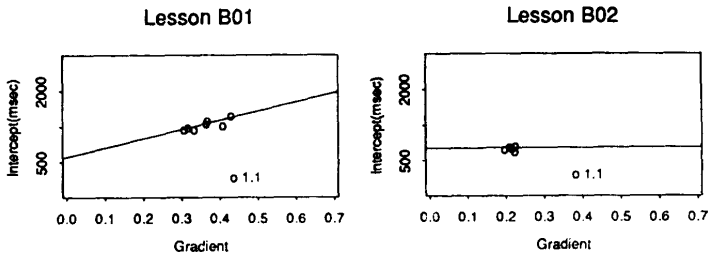

図 7 コードキャッシュモデルのシミュレーション（初期時間と上達度の相関）
Fig. 7 Simulation of Code-Cache Model (correlation between α and β).

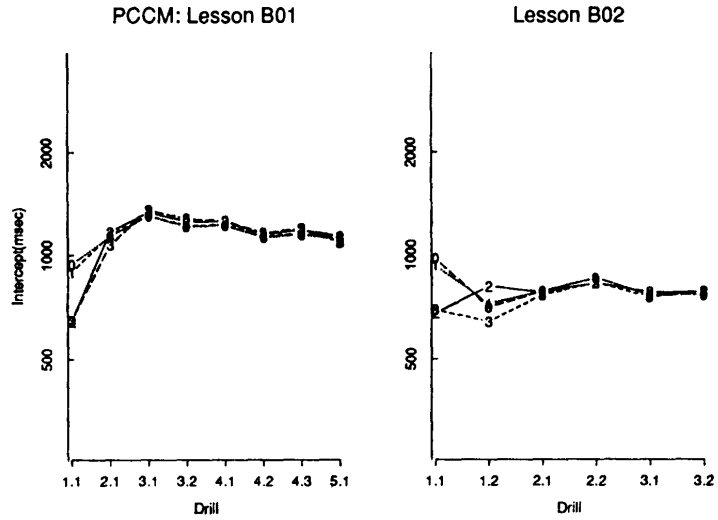

図 8 PCC モデルの初期時間の変化
Fig. 8 Intercept change (PCCM).

3.5 パイプラインキャッシュモデル

このモデルは練習の初期の状況をシミュレートするものであるが，上達に従ってどのような変化が発生しうるかを，前述の静的なパイプライン・モデルとの関連を含めて考える．

ベキ法則ではコードの引き出しがいくら上達しても，0にはならないから，このモデルでは常に文字頭の打鍵が文字中の打鍵より大きい．ところが実際の打鍵では，習熟が進むと，文字頭と文字中の双方の打鍵時間がかなりの文字数ほとんど同じになってしまう区間が多く現れる．これを，コードの引き出しが十分速くなって，打鍵時間の測定誤差範囲におさまってしまうと説明することもできる．しかし，毎打 500 ms から始まって 100 ms 程度で熟練するという観測を説明するには，α_f に比べて α_s が3分の1以下でなければならず，それでは現実の練習完了回数を遙かに超えてしまうことがわかった．

特にことわらなかったが，これまでのモデルでは，変換部と実行部が直列に処理され，入力処理部はそれらとは並列に処理を進めているとしてきていた．これ

* 最終的には数字段を含めて4段になるが，練習の初期には含めていないのでここでは3段とする．

は，初期段階では正しい処理モデルであるとみなせるが，熟練段階では，変換部と打鍵部も並列に処理が行われているはずである．

このことをモデルに組み込んだものが，パイプライン・コードキャッシュ (PCC) モデルである．打鍵コードの引き出しは，前の2打鍵の完了後ではなく，前の打鍵コードの引き出しの完了後に開始される．打鍵コードの引き出しが速くなると前の2打鍵の間に次の文字のコード引き出しが完了してしまう．これで文字頭・文字中の打鍵の平滑化を説明することができる．

打鍵時間は，前の2打鍵とコード取り出しの長いほうの累計の平均として計算する．シミュレーションの結果は図8に示すようにCCモデルと同じ傾向である．

4. 打鍵データの解析

われわれは，個人用にパーソナルコンピュータを使うように練習システムを再作成し，また，本文に述べた解析を基に，これまで使ってきた練習文の欠点と思われる点を改良した新しい文で練習を始めている．新たな被験者（1名）の練習では，内部クロック（10 ms の精度）を使って各打鍵時間を測定記録した．本章では，そのデータがモデルで予測される性質を持つかどうかを確かめる．

レッスン 1.5 には 11 個の「い」* が出現する（表5）．被験者はこのレッスンを 14 回続けて練習した．

図9は「い」の文字頭の打鍵時間を文字の位置ごとに集計し，平均の小さい順に分布をボックス図[10]で示したものである**．同じ文字であるにもかかわらず，文字の位置によって文字頭の打鍵時間に差があることがみてとれる．図9の右端に示した「い」全体の分布を示すボックスからわかるように，全体としては分布に極端な偏りはない．

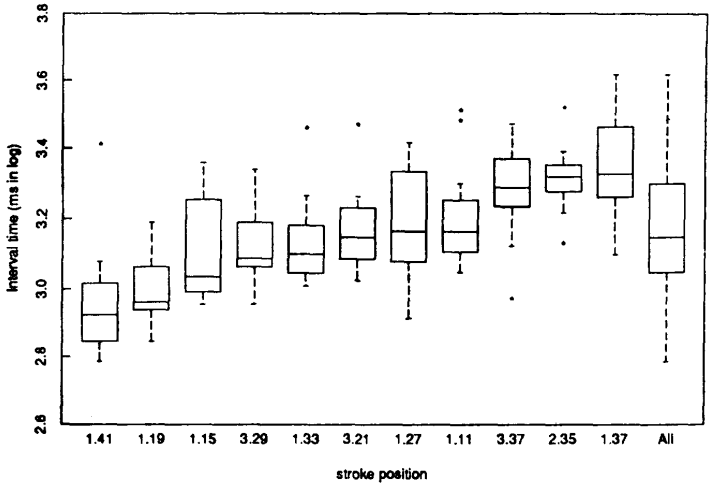

図 9 レッスン 1.5「い」の打鍵時間の分布
Fig. 9 Distribution of stroke intervals (Lesson: 1.5 (い hd)).

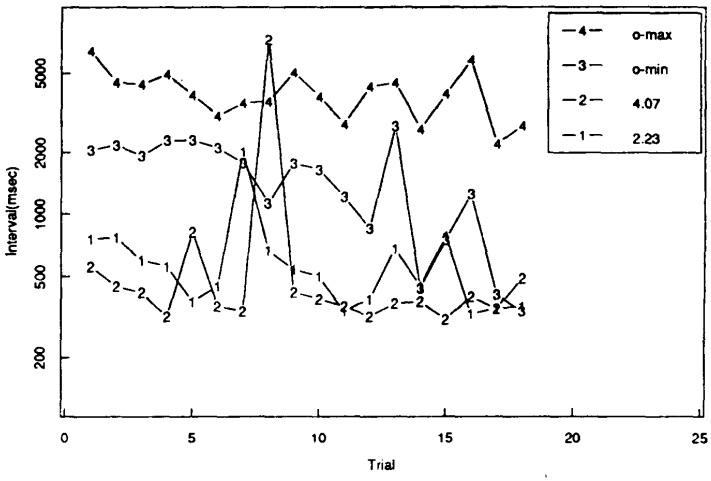

図 10 レッスン 1.5「ろ」の第1打鍵
Fig. 10 First stroke intervals (Lesson: 1.5 (ろ ps)).

表 5 新レッスン1.5のテキスト
Table 5 A new text (Lesson 1.5).

なにをしないでいたいの。ない。はい。いないがなし。しては、をした。なる。でる。いたがるとはるになると、たのしい。しているのがいるし

打鍵時間が最小の3つは直前の「い」との間に1文字が挟まっている．すなわち，CC モデルが短くなるはずと予言するシャッシュサイズが2の場合である．同じ現象は，レッスン2.4の「ろ」（図 10）など多くの場所でみられる．図 10 は練習回ごとの時間の遷移をプロットした．下の2本の折れ線は「いろいろ」というような繰り返し言葉の後の文字を打つ場合（文字間隔は2）の文字頭の打鍵時間で，上の2本は残り6カ所（文字間隔は7以上，または改行をはさむ）での

* 以後，文字の位置を，レッスンの何行目の何打鍵目かで示す．2数はピリオドでつないで表す．

** 箱の上端と下端が75%と25%値，中央線が中央値，をそれぞれ示す．ヒゲが 2σ 以内の上下限値を示し，星はそれよりはずれた値を個々に示す．

文字頭の打鍵時間の最大値と最小値をそれぞれ結んだ折れ線である．多少の飛び出た点を除けば，文字間隔2の場所での文字頭打鍵時間は残りに対して有意に小さいことがわかる．このように，多くのデータがCCモデルを支持している．

図9には文字間隔が2である場合がもうひとつ(位置 1.37) あるが，この場合の打鍵時間はむしろ大きく，モデルの予言に合致しない．しかし，これは，直前の文字が句読点の1つの「。」であるからということで説明できる．**図 11** は，レッスン1.5の全文字7種について文字頭（上半分のグラフ）と文字中（下半分のグラフ）の打鍵時間を平均した棒グラフである．集計は，文字の位置が行頭である場合，句読点の直後，その他の場合に分け，バーの色を順に薄くした．「で」を除けば，一般に句読点の直後は文字頭打鍵時間が長いという傾向が観測される．句読点は文あるいは句の終端記号であり，練習テキストの読み取りは句読点までとその後からに分けられるであろう．読み取りには一時記憶が必要だろうから，句読点の後ではいったんコードキャッシュがクリアされる可能性は高いと考えられる*．

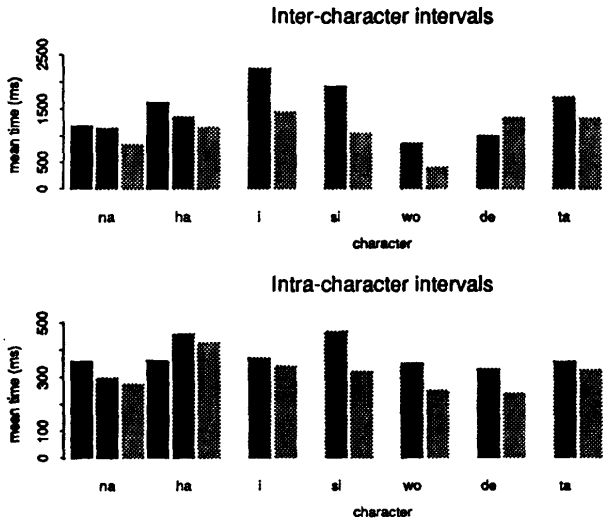

図 11 句読点の後の打鍵時間の平均（レッスン 1.5)
Fig. 11 Mean intervals after **kutôten** (Lesson: 1.5).

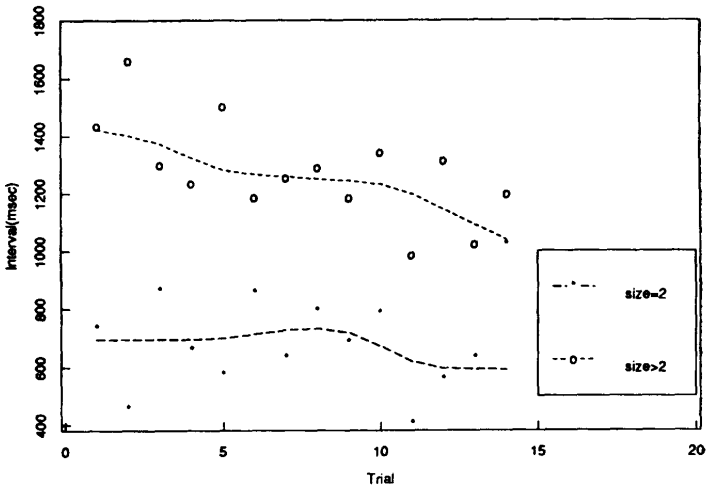

図 12 推定のコード取り出し時間の推移（レッスン 1.5)
Fig. 12 Transition of difference between 1-st and 2-nd intervals (Lesson: 1.5).

CCモデルでは，一時記憶からのコードの取り出し時間を0と仮定している．しかし，実際のデータはそれが長期記憶からの取り出し時間よりも短いものの，有限の大きさをもっているらしいことを示唆している．CCモデルでは，文字頭打鍵時間から文字中打鍵時間を引くとコード取り出し時間が得られると仮定している．この計算は当然ながら個々の実際の打鍵に当てはまるわけではないが，統計的なふるまいとして平均について成り立つ可能性がある．**図 12** は，レッスン1.5での推定コード取り出し時間の平均を計算し，練習回数ごとにプロットしたものである．練習テキストの文字間隔が2である場所とそれより大きい場所とで分けてある．図が示すように，推定されるコード取り出し時間は，平滑化すれば練習回数に従って上達を示しており，しかも，文字間隔が2より大きいほうが上達が大きい．これらは，CCモデルを裏打ちする観測であると考えられる．

5. おわりに

われわれが到達した (P) CC モデルでは，短期記憶の役割が重要であった．とくに，ふたつのドリル 1.1 の練習が高速であるのは，短期記憶からのコード取り出しがあるためである．しかし，コード獲得が促進されないこのようなテキストは練習としては不適格である．同様に，英語の練習テキストとして “asdfg asdfg …” のような繰り返しが不適格であるのも，構造の単調な “asdfg” が1チャンクを作ってしまうと考えれば，CC モデルで説明がつけられる．

* ただし，元の課題 B01.1.1 のような句読点だけの練習では，このようなコードキャッシュのリセットが1文字ずつ起きると考えるべきではない．

テキストの設計にもわれわれのモデルが有効である．効率のよい練習のためには新たに練習する文字の密度を高くすべきであるが，このために短期記憶にコード記憶が残らないようにしなければならない．モデルから得られる知見は，同一の文字の間隔を3以上にすれば，初期の練習においては十分であることが予想される．これに沿って，練習文字の密度が高い新しい練習テキストを現在作成している．さらに，最適なテキストの設計や練習回数の設定についても，モデルを用いて計算することが可能である．

CC モデルから PCC モデルへの移り変わりについては，われわれは，徐々に移行していくのであろうと考えているが，それを検定するだけの統計的精度がデータにはない．また，ごく初期のコード表の見方を理解する過程などで，コード記憶以外の処理が行われている可能性も高いが，これらもその存在を練習の観測値から確認することは難しい．むしろ，ベキ法則がきれいに成立していることから，これらの処理が入力部で並行して行われることの可能性が高いと考えられる．いずれにしろデータ精度が得られない以上，これ以上の議論は控えるべきであろう．

われわれの議論は単純な処理の上達のベキ法則性に基礎を置いている．その処理として変換部および実行部で行われる単位動作を考えてみると，これらは広い意味で記憶の強化である．記憶の強化とその逆の忘却においても，ベキ法則がみられることはよく知られている[2),11)]．タイピングのみならず一般的な認知・運動においてベキ法則が成立する理由が，基本とする処理が記憶の強化に還元されることにあると考えることは妥当なことではなかろうか．

謝辞 本研究は，東京大学理学部情報科学科に在任中，同所の山田尚勇教授を中心とする日本文入力システムの研究・開発プロジェクト[6)]の一部として行われたものです．本研究をご指導くださった山田教授に感謝いたします．また，初期の練習システムを作成してくださった平賀譲氏，NEC PC 9801 用の練習システムを作成してくださった鈴木浩之氏，菊池徹氏，中山健氏，有益な助言をいただいた山田研究室の方々，および，Tコードのタイピストとしてご協力くださった皆さんに感謝します．

参考文献

1) Crossmann, E. R. F. W.: A Theory of the Acquisition of Speed-Skill, *Ergonomics*, Vol. 2, pp. 153-166 (1959).
2) Anderson, J. R.: *Cognitive Psychology and Its Implications*, second ed., A Series of Books in Psychology, Freedman, New York (1985).
3) Shaffer, L. H.: Intention and Performance, *Psychological Review*, Vol. 83, No. 5, pp. 375-393 (1976).
4) Okadome, T., Ono, Y. and Yamada, H.: Pipeline Model of Information Processing during the Typing Task, *Proceedings of International Conference on Computer and Communications (ICCC '86)*, Beijing, pp. 60-101, China Computer Society (Oct. 1986).
5) Hiraga, Y., Ono, Y. and Yamada, H.: An Assignment of Key-Codes for a Japanese Character Keyboard, *Proc. 8th Intern. Conference on Computational Linguistics (COLING-80)*, Tokyo, pp. 249-256 (Sep.-Oct. 1980).
6) 山田尚勇：専任タイピスト向きタイプ入力法の研究経過，コンピュータソフトウェア，Vol. 2, pp. 344-354 (1985).
7) 小野芳彦，橋田浩一，鈴木浩之，山田尚勇：日本文入力のための位置対応型コードの表示方式について，第 31 回情報処理学会全国大会論文集，pp. 1397-1398 (1985).
8) 平賀 譲，小野芳彦，山田尚勇：日本語タッチタイプ入力コード用の練習システムの作成，第25回情報処理学会全国大会論文集，pp. 1087-1088 (1982).
9) 小野芳彦，平賀 譲，山田尚勇：日本語タッチタイプ入力コード用の練習システムのテキストについて，第 25 回情報処理学会全国大会論文集，pp. 1085-1086 (1982).
10) Becker, R. A. and Chambers, J. M.: *S—An Interactive Environment for Data Analysis and Graphics*, Wadsworth Advance Book and Software, Belmont (1984). (渋谷政昭，柴田里程訳：Sシステム，I，II，共立出版，東京 (1987).)
11) Klatzky, R. L.: *Human Memory: Structures and Processes*, second ed., Freedman (1980). (箱田裕司，中溝幸夫訳：記憶のしくみ，サイエンス社，東京 (1982).)

(平成元年9月4日受付)
(平成2年6月4日採録)

小野 芳彦（正会員）

1951 年生．1974 年東京大学理学部卒業．1976 年同大学院理学系研究科修士課程修了．1978 年同博士課程中退．同年東京大学理学部情報科学科助手．1989 年国際日本文化研究センター助教授．理学博士(東京大学)．日本研究における計算機支援についての研究を行う．日本文入力やユーザ・インタフェースの観点から，文科系諸学での計算機利用法に興味を持つ．JUS，日本ソフトウェア科学会各会員．