8 Adaptive Teaching Machines

A steady-state system, used in connection with a task such as teleprinting (Pask *et al.* 1969a), speed reading or various tracking and compensation activities, has a training function in its own right since it keeps the subject (student or trainee) poised in an operating region, where the problems posed by the skill are difficult enough to be taxing without becoming unintelligible; as they would be in extreme overload. In particular, it is sometimes possible to choose ξ so as to maximise the overall rate of adaptation. As a specific example, consider the following *code* learning or *transformation* task which has been used widely in experimental studies and which underlies the simulation described in Chapter 6.

Two versions of the task were employed; one with a six-fold input (Pask and Lewis, 1967) and one with an eight-fold input (Pask and Lewis, 1968). The latter is described below.

1 Code Learning and Transformation Tasks

A subject is presented with an effectively unlearnable sequence of inputs, spaced equally in time, which are configurations of up to four illuminated signal lamps, selected from a group of eight lamps. Before the experiment begins, he is told that a transformation rule, Ω, relates the group of signal lamps to a row of eight response buttons; he knows this rule and is able to recite it. Given an input, x, he is required to solve the problem of producing a response, $y = \Omega(x)$, within a deliberately restricted interval, δt. This response is complex, for he must press several buttons to denote the transformation of the signal configuration.

The size of the signal configuration, the value of δt and the inter-trial interval, Δt, are chosen so that (a) a fully proficient subject can perform the skill accurately, but (b) a novice finds the problems posed by x in the context of Ω unsolvable (e.g. his uncertainty regarding the value of y when x is given, is such that he is grossly overloaded). Typically, $\Delta t = 5{\cdot}5$ s, and $\delta t = 4$ s.

In these conditions a novice can neither perform the skill nor learn to

perform it unless he is provided with external assistance. If a marginal amount of external assistance is provided by a steady-state regulator, then the subject is forced to exhibit various problem-solving behaviours which would, in more relaxed circumstances, be unobservable.

In order to start a learning process, the regulator must reduce the difficulty of the situation. It does so by 'simplifying', or 'partially solving' the problems posed by the inputs. A combination of two simplifying procedures is used for this purpose.

One procedure involves the reduction of the number of lamps in a signal configuration. The other consists in providing cue information that partially specifies the correct response. Both of these procedures, as well as their combination (to form a conjoint simplifying operation), have been checked empirically, that is, the percentage of correct responses elicited by a sequence of more simplified input problems is never less than the percentage of correct responses elicited by a sequence of less simple input problems and, with the exception of a fully proficient subject, it is greater. The snare to be avoided is that some intuitively acceptable simplifying procedures suffer from an 'inversion' noticed by van der Veldt (1928, cited in Woodworth, 1950). If a subject is taught to respond to a complex stimulus by way of simple exemplars, the subsequent presentation of a less complex stimulus may give rise to more, rather than less, difficulty. The subject acquires a perceptual motor organisation applicable to the complex situation; the organisation apposite to less complex situations has decayed, since it amounted only to cognitive scaffolding needed to reach and deal with the complex problems.

Specifically, if m designates the number of signal lamps in an input configuration, then at any trial the equipment quasi randomly and equiprobably selects one of the roman numbered columns in the table shown and four levels of simplification are obtained by selecting one row (ordered in decreasing difficulty, η, from top to bottom) where the letters designate signal lamps.

m	I	II	III	IV	V	VI	VII	VIII
4	*abch*	*abdc*	*bdce*	*cedf*	*egdf*	*gehf*	*gafh*	*gahb*
3	*abc*	*bdc*	*dce*	*def*	*efg*	*gfh*	*hga*	*ahb*
2	*ac*	*bd*	*ce*	*df*	*eg*	*fh*	*ga*	*hb*
1	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*

The other part of the simplifying operation provides cue information through a further row of lamps adjacent to the response buttons and within the interval, δt, in which a response can be submitted and accepted. Cue information partially specifies a correct response, $y = \Omega(x)$. It is provided if a variable $z = 1$ and is not provided if $z = 0$. Specifically, if $z = 1$ and if x is an m-lamp configuration, then after the subject has selected $m = 1$ correct members of y, he is automatically provided with a specification of the remaining selection that he must make in order to achieve a correct response.

These procedures are combined to provide eight degrees of difficulty indexed by the variable, η. The combination of procedures is determined by the listing shown below; values of η being determined by the steady-state control mechanism. As an initial condition for the system $\eta = 0$

Value of η	8	7	6	5	4	3	2	1
Value of m	4	4	3	3	2	2	1	1
Value of z	0	1	0	1	0	1	0	1

The response, y, is complex and is obtained as follows. The subject responds by selecting a configuration, y, of response buttons and is required to select the particular configuration $y = \Omega(x)$. If Ω is a permutation of signal lamps with respect to response buttons and if the input at the nth trial, $x(n)$, has m components, then a correct response, $y(n) = \Omega[x(n)]$, also has m components. In order to count as a correct response, $y(n)$ must be completed within the interval δt during which $x(n)$ is exposed; and the response buttons are automatically rendered ineffective after δt has elapsed.

However, the response buttons need not be pressed simultaneously and the subject is permitted to construct y in any order he pleases. To avoid an unwanted load on his immediate memory capacity, each response button is associated with a 'lock' relay which is energised if the corresponding button is pressed and remains energised until 0·5 s after the end of δt. Whilst the relay is energised, an indicator adjacent to the response button is illuminated and the set of illuminated indicators provides a visual record of y. The interval δt is terminated by the disappearance of the input signal and the coincident illumination of 'knowledge of results' indicators, in register with the response buttons, that indicate the correct response $\Omega(x)$. This presentation persists for 0·5 s. after the end of δt (so that it coincides with the image of y thus allowing the subject to compare $\Omega(x)$ and his actual response, y). Coincidentally, the subject receives an auditory and visual signal (buzzer and lamp) to indicate whether his response is completely correct (all response selections correct) or mistaken. At an instant, $\delta t + 0{\cdot}5$ s after the appearance of the stimulus, the 'lock' relays are de-energised and the indicators are extinguished. There is a rest interval of 1 s and the next stimulus is presented $\Delta t = 5{\cdot}5\ \text{s} = \delta t + 0{\cdot}5\ \text{s} + 1\ \text{s}$ after the last.

A proficiency index, ρ, is calculated by averaging correct response frequencies, either latency weighted or not, over coherent blocks of trials.

As in the last chapter the steady state control mechanism adjusts η to maintain a null-point value of ρ, at or near to a null-point value ξ.

In a skill of this type, many ξ values are stable, provided they are within a certain range. But, regarding the system as a tutorial arrangement, the preferred value is one which maximises the rate of adaptation or satisfies a similar terminal performance condition (near to the limit at which the steady-state regulation has nearly exhausted the variety it is in a position to introduce); for example, when $\eta = 8$ and when $\rho > 0{\cdot}8$. Call the number of trials needed (by a given individual) to reach this terminal performance condition, T. The mean values of T are determined for groups of students (ten in each group for the data shown below) working under different, but stable value of ξ and the T means are compared in order to ascertain the most 'effective' value of ξ which minimises the value of T.

	T(mean)	T(standard deviation)
$\xi = 0{\cdot}55$	247·4	38·34
$\xi = 0{\cdot}65$	386·5	68·78
$\xi = 0{\cdot}75$	503·4	79·89

Using Jonckheere's (1954) trend test, the evident trend in $T(\xi)$ is significant at the 0·1 per cent level ($0{\cdot}001 > p$). Further reduction in ξ leads to instability in the steady-state control system. Thus, for a skill of *this* kind, adaptation rate is maximised by choosing ξ near to the overload point and thus presenting problems that are as difficult as possible, though not unintelligible. The adjustment is a typical 'tuning' operation (in the sense of Chapter 4, Fig. 32).

2 Adaptive Regulation

Such a choice of ξ relies upon an assumed constancy of the operating region boundaries between individuals. In general, this assumption is untenable; moreover, it is usually untrue that an optimum ξ remains fixed, even for one individual, during a training session.

In order to estimate and establish effective values of ξ under *these* circumstances, it is possible to use adaptive machines that operate as local optimisers. Such a machine selects values of ξ to determine the stable boundaries; it selects *test values* of ξ, within these boundaries, and, by successive sampling, picks whatever value maximises a function such as the (approximated) rate of increase in η with ξ (and thus ρ) held at the currently selected test value. From this point onwards, the current value of ξ is maintained for most of the time at the 'best' value so far discovered, but some time is allocated to sampling performance under a *test value* which is incrementally distinct from the current value. If the current value is superior, then an increment in the opposite sense is selected as the next test value and if not the next current value becomes the test value. The process is iterated and is chiefly limited, in practice, by the number of trials over which the rate of change in η can be estimated. Such a device is shown in Fig. 75. This, and

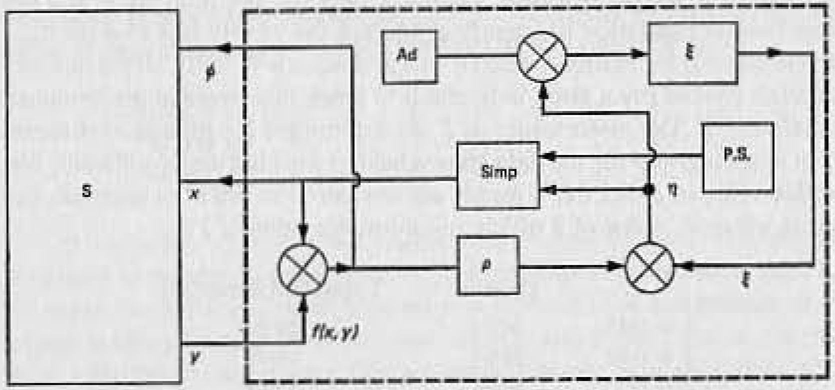

Figure 75 Adaptively controlled system: x, input; y, response; ϕ, knowledge of results; P.S., source of problems; Simp, simplification; $f(x,y)$, correct response function (for example, 1 if $\langle x,y \rangle \in \Omega$; 0 if not); ρ, performance index; ξ, steady-state criterion; η, difficulty index; Ad, adaptation rule; s, student; $\otimes$, comparators, as in control engineering notation.

other more complex arrangements (described in the following sections) were programmed on a special-purpose computer (Plate 5) as well as being built into numerous experimental or industrially used equipments (Plates 6 and 7).

2.1 Two variations on this theme indicate the scope of adaptive regulation and the adaptive machine as a training mechanism. One variation is based upon the code-learning or transformation skill of section 1, and the other upon a multidimensional steady-state system, exemplified by a trajectory recognition and target interception skill, briefly described in the next section.

2.2 The cellular display of Plate 6a consists in a matrix of signal lamps which are turned on and off at variable intensities to form a confusing or 'noisy' background. Trajectories of illuminated signal lamps, moving from left to right across the display, can be superimposed upon the 'noisy' background at various intensities and thus (if other things were equal; for example, if all trajectories were equally recognisable) would correspond to varying degrees of prominence of discriminability.

There are eight classes of trajectory, differing in their pattern of motion. Members of each class are sampled equiprobably, but each class of trajectory may be presented with higher or lower intensity (the lower the intensity for the ith class, the greater is η_i).

The trainee is required to intercept the target responsible for a trajectory by using one or, if need be, both of two vertical interceptors that move up the right-hand side of the display. He is warned if a target *may* be in the offing but is *not* told which class of trajectory will characterise its motion. This information, which is essential in order to intercept the target at all and which must be received as early as possible to achieve a provident interception, is garnered from observing the first phase of a trajectory and combining the clues so obtained with background knowledge about the possible classes of trajectory (in certain variants of the task, also from positional information, furnished by the cue lamps bordering the display, about regions where targets are likely to be found).

Training[1] begins with $\eta_i = 0$ for all $i = 1, \ldots, 8$. A proficiency measure ρ_i is computed with respect to each trajectory class as a function of success and the number of interceptors used up at a trial. The resulting ρ_i values form the input to eight subcontrollers that adjust the η_i, to maintain $\rho_i = \xi$. With $\eta_i = 0$ the task is quite easy to perform. At the upper limit of loading and it is impossible for anyone to maintain perfect and provident interception.

The adaptive form of this block of eight steady-state subcontrollers contains an additional overall controller which prescribes separately determined values; ξ_i, to maximise the expected rate of increase in the product of the stabilised η_i. It is shown in Fig. 76. In fact, it is difficult to operate the system in a stable manner without the adaptive control loop. Typical data are shown in Fig. 77.

2.3 An obvious extension of the code learning or transformation task described in section 1 is to require a student to deal with two or more coding rules; say Ω_1 and Ω_2. For example, the tenure of rule, Ω_1 or Ω_2 is alternated, haphazardly, from one *block* of six or eight trials to the next (though the student is informed which code rule is applicable at any trial *block*).

In changing the transformation or code rule, it is, of course, necessary to change the steady-state controller for there is no reason whatever to suppose that Ω_1 and Ω_2 are homogeneous. Thus there are two subcontrollers, one

1. Several variations on this training system have been used from time to time. For example, in one of them, the relative frequency of appearance of members of the trajectory classes is regulated, as η_i, in place of display intensity.

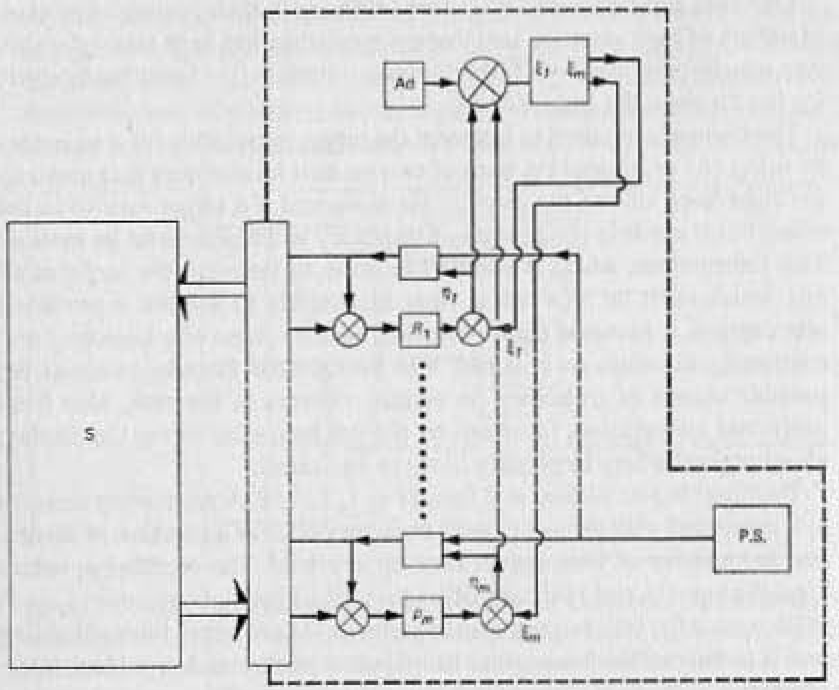

Figure 76 Multidimensional adaptive system. Note single problem source (P.S.). The system has m controllers and one overall controller. Apart from the subscripts, the notation is as presented in Fig. 75.

for Ω_1 and one for Ω_2; one varies η_1 to maintain $\rho_1 = \xi$ and one varies η_2 to maintain $\rho_2 = \xi$.

Under these circumstances, especially if the rules are chosen to have a few features in common but most features distinct, there is pronounced interference which hampers learning to the extent that most students are unable to acquire the skill unless the time constraints are relaxed. However, with the same time constraints imposed, the skill is learnable if the student is allowed to choose the rule he will concentrate upon during a trial block (in a pause prior to presenting the block inputs) under the caveat that he must ultimately alternate the code rules in a 50 per cent fashion.

Data from two subjects learning in this condition are shown in Fig. 78. The shaded region between the curves for η_1 and η_2 is an index of the degree of interference between acquiring Ω_1 competence and acquiring Ω_2 competence and the code rule applied over a given trial block is shown on the upper line (shaded for Ω_1 or blank for Ω_2).

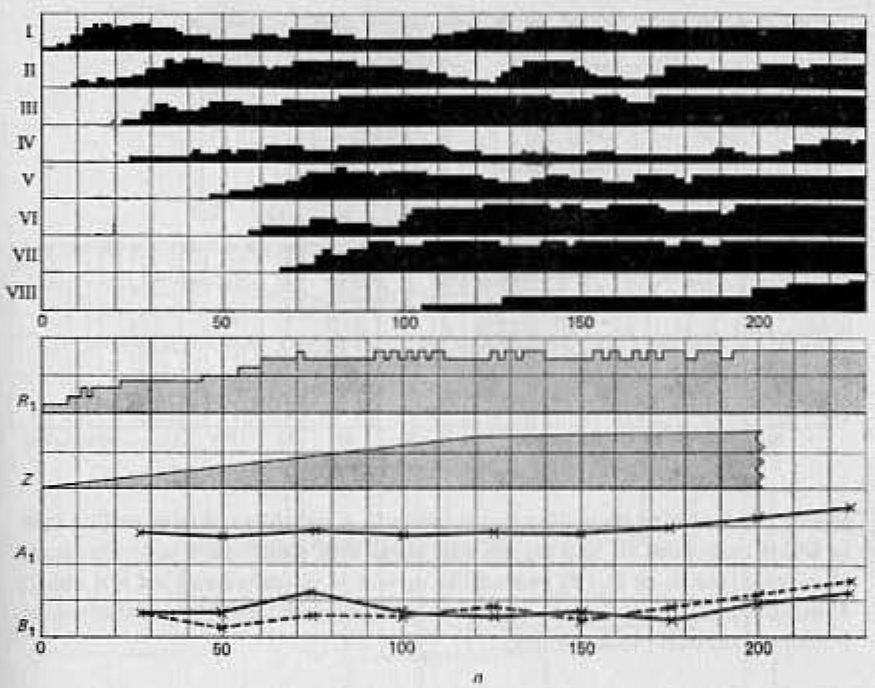

Figure 77 Learning curves for one subject (see text): I to VIII, values of η_1 to η_8; R_1, mean of the η_1; Z, product function used as control variable; A_1, mean value of the ρ_1; B_1, differential mean ρ values for double interception (solid line) and single interception (dashed line); n, trial number.

2.4 The adaptive form of this training system is shown in Fig. 79. The overall controller does not change the steady-state parameters, ξ, of the subcontrollers (though it *could* do so, in principle). Instead, it executes a simple selection strategy, which consists in assigning probabilistic weights π_1 and $\pi_2 = 1 - \pi_1$ to the selection of Ω_1 (and its subcontroller) or Ω_2 (and its subcontroller) at the beginning of a trial block. In the typical, simplest case examined, the value of

$$\pi_1 = \frac{\eta_{max} - \eta_2}{(\eta_{max} - \eta_1) + (\eta_{max} - \eta_2)}$$

is applied as a biassing input to a quasi-random selection mechanism (since there are only two subcontrollers and $\pi_1 = 1 - \pi_2$ this input is sufficient). In general, an adaptive controller of this kind furnishes $M - 1$ biassing inputs and a selection is made between $M > 2$ subcontrollers; for example,

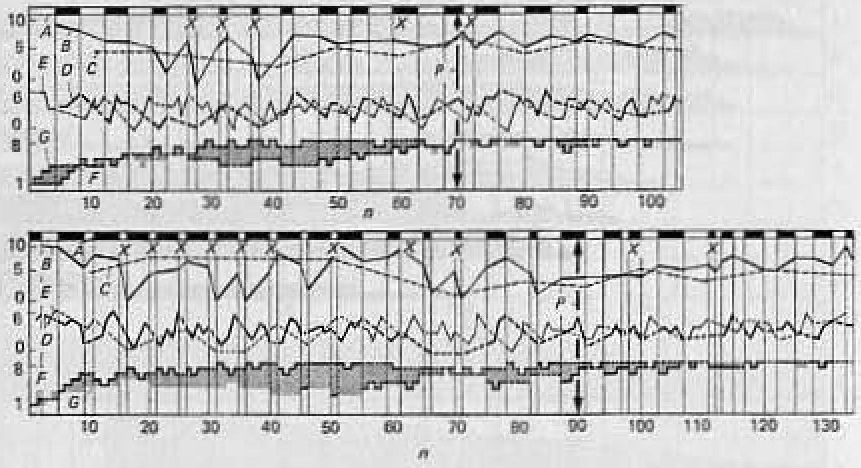

Figure 78 Learning curves for two subjects in a transformation or coding task in which code rules Ω_1 and Ω_2 are alternated with steady-state subcontrollers: *A*, current rule Ω_1 or Ω_2; *B*, probability $\pi_1 = 1 - \pi_2$ (calculated but not used); *C*, mean ρ; *D*, ρ_1; *E*, ρ_2; *F*, η_1; *G*, η_2; *P*, point at which definite strategy of alternation is adopted; *n*, trial block number.

in Pask (1964b). That reference describes an adaptive system for training in electronic fault detection.

Some typical data are shown in Fig. 80, which should be compared with the curves shown in Fig. 78. The salient difference, apart from the more rapid learning in the adaptive system, is the suppression of interference.

2.5 One further extension of the system is worth examination. There are good reasons to believe that the 'free learning' of Fig. 78 is ineffective because students are substantially unaware of their own ability to assimilate the subskills learned under Ω_1 and Ω_2 or to integrate them into one coherent skill, as they must do to perform effectively. For *all* students there is a breakpoint, shown as 'P' in Fig. 78, when a definite strategy is adopted after which the student does come to terms with the task. Some students (the lower graph in Fig. 78 for example) recognise that a strategy is an *operational* requirement very tardily, though most people are willing to pay lip service to it; that is, they talk of 'planning' and the like. Incidentally, it does not follow that people with the most readily verbalised 'plans' are the most effective learners. They often tackle the task with spirit and high motivation but are no more successful, on average, than individuals who have plans that *evolve* in the course of learning and cannot be neatly described (at any rate, before the event). These comments are supported by a specific sub-study described in a technical report (Pask, Mallen and Scott,

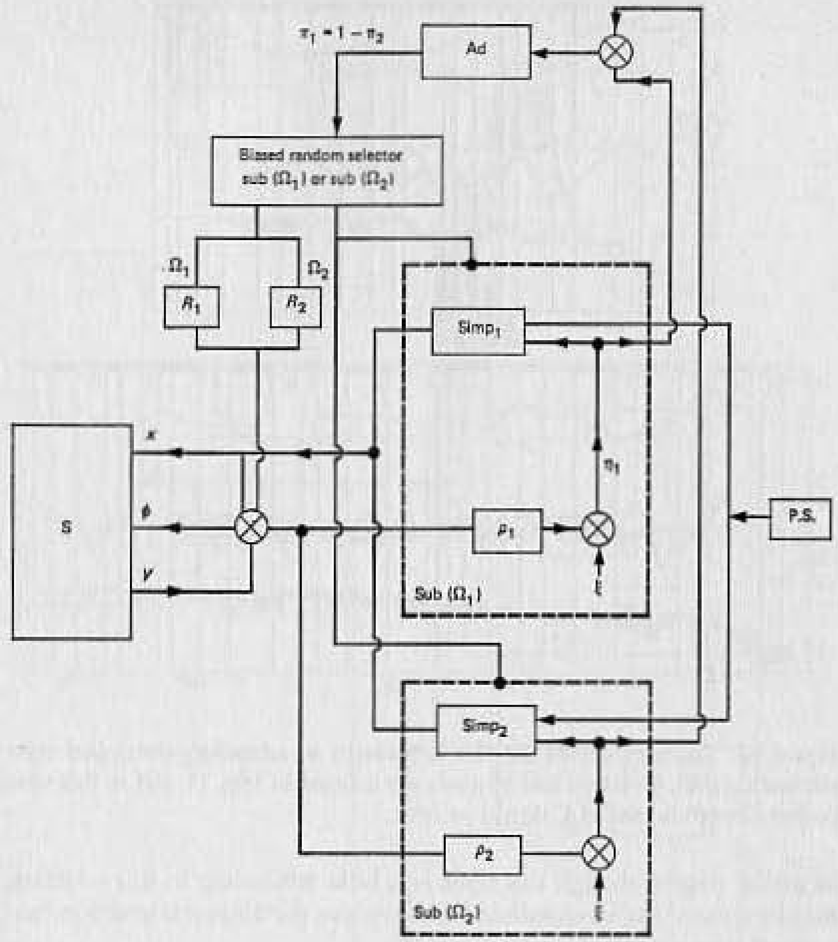

Figure 79 Code learning with subskill alternation. Subcontrollers for each subskill are enclosed in dashed lines and designated sub (Ω_1) and sub (Ω_2). Ω_1 and Ω_2, coding rule inscriptions as function tables; symbol ●, selection as alternatives, one and only one at once. Other notation is as presented in Figs. 75 and 76.

1968), carried out in connection with the overtly strategic learning task discussed in Chapter 11.

On the other hand, though learning in an adaptive system is impressively fast and free from interference (Fig. 80), this effect is most certainly not eliminated (for example, the right-hand graph). At least some of the residual interference is due to the participant interaction, noted in the last chapter, and the following arrangement (called an 'adaptive metasystem'

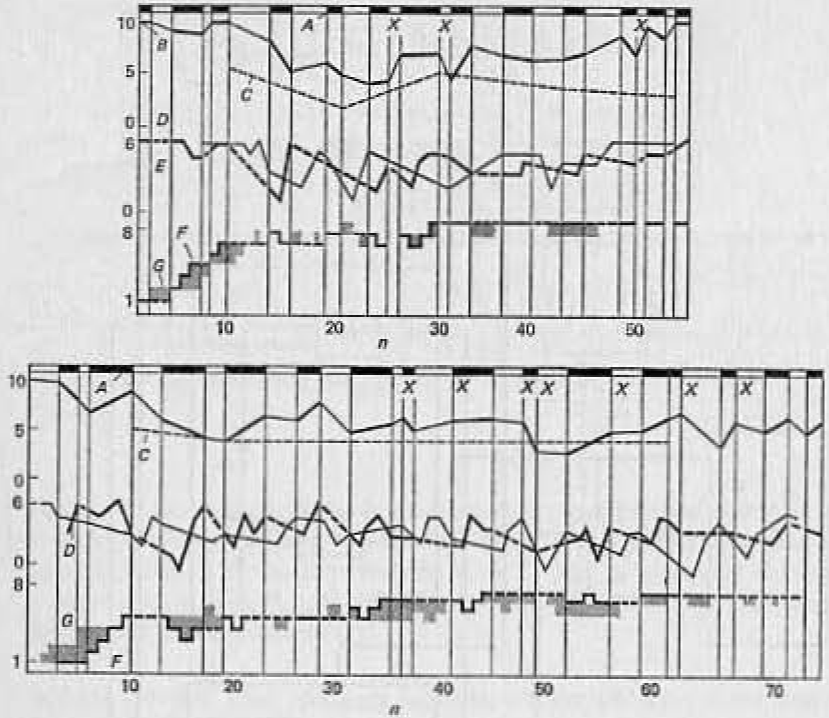

Figure 80 Learning curves for two subjects in an adaptively-controlled code alternation skill. Notation and symbols are as used in Fig. 78, but in this case probability (*B*) is used as a control variable.

in earlier papers, though the name is a little misleading in this context), may be regarded as an expedient for converting participant interaction into a 'legal' interaction.

The arrangement brings about a compromise between the student's choice of the code rule to work on and the overall controller's selection. For each trial block, here indexed n, the overall controller calculates the dominance function

$$Domin = \sum_i p_i \, . \, \eta_i (1 - \|\pi_i - \tfrac{1}{2}\|); \; i = \{1,2\}$$

with the properties that for $n = 0$ *Domin* $= 0$ and that for $n = T$ (the terminal condition, here requiring proficiency in performing a hybrid or amalgam of both subskills) *Domin* $= 1$. As suggested by its name, *Domin* determines the overall controller's degree of dominance over the student. At the beginning of each trial block, the student is allowed to submit his *proposal* for which subskill shall be rehearsed (either deterministically by

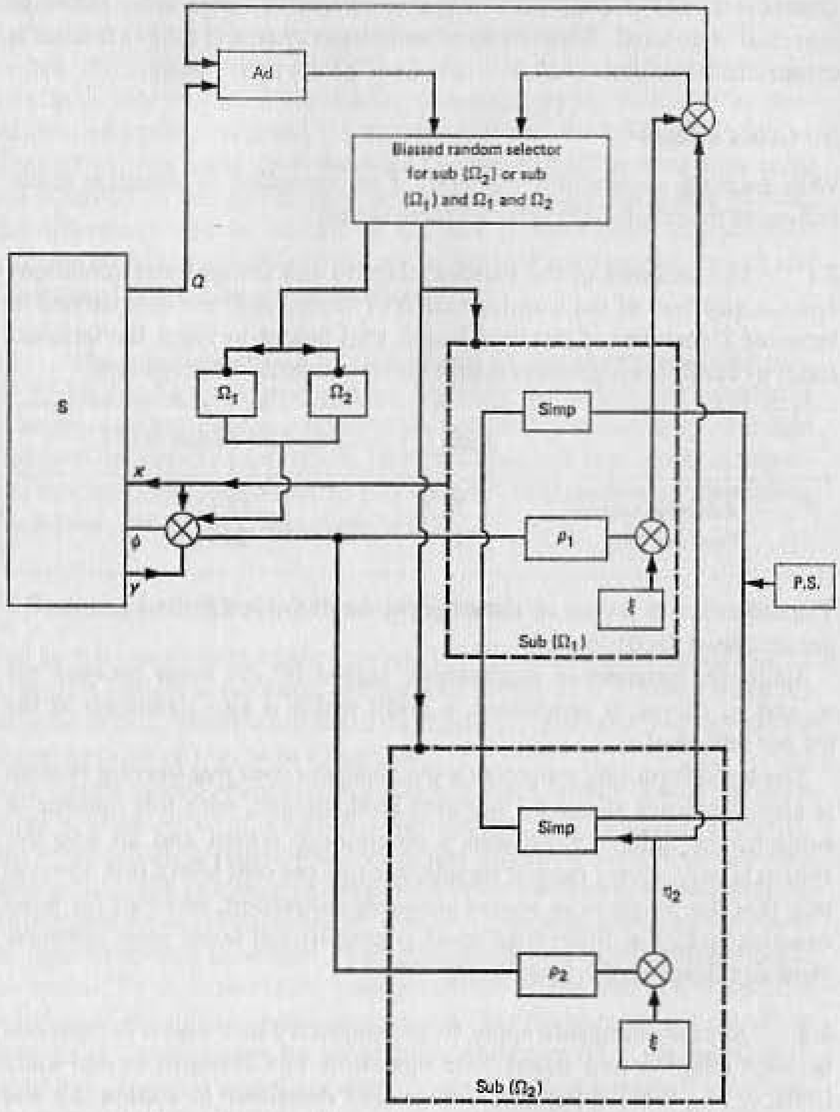

Figure 81 Adaptive control and compromise (or metasystem) solution: *Q*, student's selection; other symbols are as used in Fig. 79.

pressing a button or probabilistically by setting a confidence estimation scale expressing his belief that Ω_1 or Ω_2 should be rehearsed). The weight given to his proposal depends upon the prevailing value of *Domin*. If *Domin* $= 0$, the student's choice or his probabilistic bias is decisive; if

Domin = 1, the overall controller's bias; in the latter case, alternate rehearsal is ensured. Between these values the system (Fig. 81) reaches a compromise solution.

3 Gross Effects

With training applications in mind, it is interesting to compare crude indices of the relative efficiency of these systems.

3.1 The influence of the various adaptive and compromise conditions upon acquisition of the transformation or coding skill are summarised in terms of T (number of constant-length trial blocks to reach the terminal state) by the following figures (again for ten subjects in each group).

	Mean (T)	Standard deviation (T)
Adaptive system	76·2	13·14
Free learning	116·4	20·81

The difference in favour of the adaptive condition is significant at the 0·1 per cent level ($0{\cdot}001 > p$).

Similarly, interference suppression, judged by the areas between the η_1 and η_2 curves, is suppressed, a result which is also significant at the 0·1 per cent level.

The trend favouring *compromise* over *adaptive* over *free learning* systems is also significant at the 0·1 per cent level, though, with this number of subjects, the difference between a *compromise* system and an adaptive system is only of very modest significance (0·5 per cent level). It is however true that the *compromise* system enhances motivation, and that for more complex tasks the differential result is unequivocal (even gross measures yield significances of 0·1 per cent).

3.2 Similar comments apply to the trajectory task and a comparison between adaptive and steady-state operation. For example, in one study (Pask *et al.*, 1967) using the arrangement described in section 2.3 and groups of eighteen and twenty-five subjects (with n = number of trials) we obtained the figures shown below.

	Mean (T)	Standard deviation (T)
Steady-state	558	86
Adaptive system	1225	215

The difference in favour of the adaptive process is significant at the 0·1 per cent level on a non-parametric test (Mann–Whitney U).

Here, there is also data on long-term retention (over a few weeks) which is substantially perfect. Very similar comments apply to the other procedures and conditions noted in the discussion, and the difference in favour of the adaptive mode is rarely less than 1 per cent significant, even for small and relatively *ad hoc* studies that have been carried out from time to time. The differences tend to increase as the task is made more complex (for example, by introducing predictive and positional cueing information) and if individual differences are pronounced.

3.3 The transformation and code-learning tasks closely resemble real-life keyload skills (shift positions or duplicate keyboards are analogous to multiple codes). Not surprisingly, the laboratory results tally with field studies of devices like SAKI (Pask, 1958, 1961) when it is properly incorporated in a training scheme and to very detailed field studies of teleprinting (Pask *et al.* 1969a; Scott and Pask, 1973).

4 Some Inherent Limitations of the System

Just as the steady-state control system (or the steady-state technique, in general) is subject to the limits summarised in Fig. 32 of Chapter 4; so the *adaptive* control system is subject to the limits of Fig. 33 and the *compromise* system to those of Fig. 34 in Chapter 4.

At a more mundane level (though the argument reflects exactly the same ideas) a single variable regulator relies upon a rank ordering of inputs; that is, the problems engendered by a higher value of η must be consistently more difficult than those engendered by a lower value of η. Consistency, in the required sense, is established by checking that there are no inversions of the type mentioned in section 1 and intuition is fairly often misleading in this matter. By the same token, a many-variable system relies upon a partial ordering of difficulties (in the same sense). The primary criterion may be taken as an ordering over the problems of each subskill in respect of each variable, η_i, together with the possibility of embedding subsets of problems, characteristic of the integrated skill, in a further ordering system (Gaines, 1972). The chief difficulty occurs due to the fact that a skill can be assembled in many legitimate ways out of its subskills; different methods can be adopted by different students or the same student on different occasions.

If a skill does have these very restricted properties it is called a structured skill (Pask, 1964b; Lewis and Pask, 1965) and it turns out that the most interesting tasks do not belong to this category. For all that, the multi-variable adaptive system and the compromise system have a flexibility that a steady-state control system does not have; the regulator is *not* confined to compensating for changes of attention that are localised, effectively, in one η related universe (like the regulator in Chapter 4, Fig. 32). In a non-trivial, though still very special sense, the regulator may deal with changes of attention over several distinct universes (Chapter 4, Fig. 34).

5 Self-Organisation and Uncertainty Regulation

This section is devoted to the following theoretical notions.

(a) The adaptive regulator (including the *compromise* case of section 2.5), when it is interacting tutorially with a student, is an hierarchically organised system with *Lev 1* and *Lev 0* as levels of control. It is also part of a learning and teaching system in which the learner cannot be fully disentangled from the teacher (Chapters 5 and 6).

(b) A steady-state regulator and the system it controls fit into a degenerate form of this paradigm.

(c) Performance indices and difficulty indices may be expressed as selective uncertainty/information indices.

(d) As a result of this, the adaptive/compromise systems are seen to be self-organising systems in Von Foerster's (1960) sense (Appendix A).

(e) The steady-state system is a degenerate form of self-organising system.

(f) It is possible to interpret the selective uncertainty/information indices as indirect estimates of subjective uncertainties and to replace them (if the sampling conditions of Chapter 1 are satisfied) by direct estimates. Due to the nature of perceptual motor skills, it is usually impracticable to institute a change to direct estimation. But the possibility of doing so is important since direct estimates underlie the uncertainty regulation procedures, apposite for the intellectual and strategic learning to be discussed in Chapter 11.

5.1 The constructions promised in (a) are shown in Figs. 82 and 83, where the hierarchical components may be conceived as finite-function machines (Chapter 2) or, in the lower boxes, aggregates of these. A more realistic interpretation, and a slightly more liberal one, is obtained (as promised in Chapter 5) by regarding the constituent finite automata as fuzzy automata or fuzzy algorithms (under execution in a generally non-numericised manner).

5.2 For the steady-state system, the last clause (fuzzy non-numericised execution) is excluded; that is, such a process cannot be *observed* or incorporated into the regulator design.

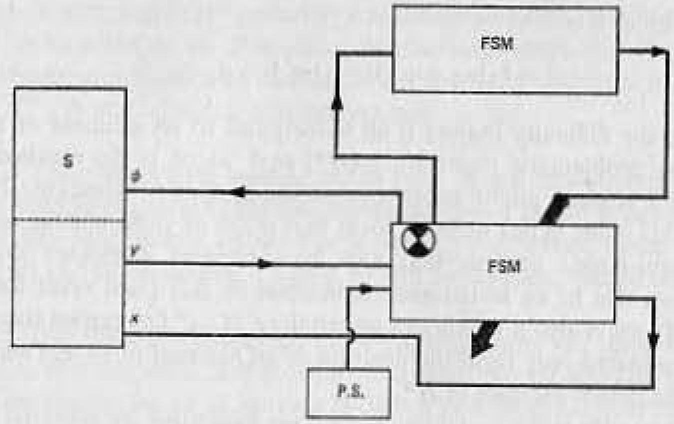

Figure 82 A finite-function machine construction for the adaptive teaching/learning system. Symbols and other notation as used in Chapter 2. *Lev 0*, *Lev 1*, levels in control hierarchy; FSM, finite-state machine; P.S., problem source shown as input to lower FSM. Dotted line across student indicates that he may be (minimally) imaged as a pair of FSMs.

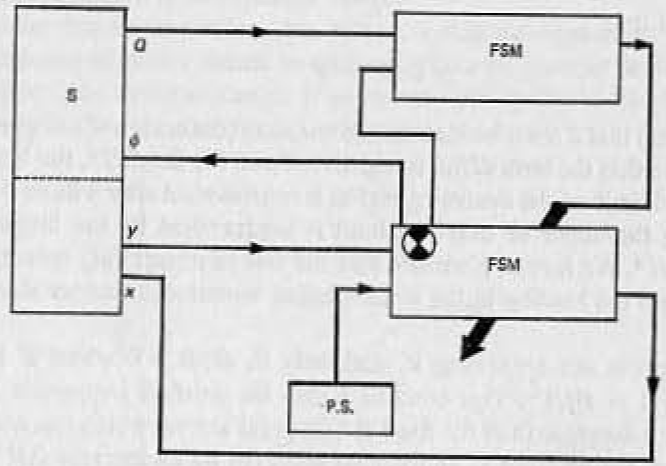

Figure 83 A finite-function machine construction for the compromise teaching/learning system: Q, student selections; *Lev 0*, *Lev 1*, levels in control hierarchy. Other notation is as used in Fig. 82.

5.3 A common feature of all the performance indices ρ is that, with respect to a *given goal* (to bring about $\mathscr{R}$, for example), they correspond to selective uncertainty/information indices (as in Chapter 1). The general form of the index is:

$$H = (\rho \log \rho) + (1 - \rho \log 1 - \rho).$$

Similarly, the difficulty indices η all correspond to an estimate of the structural or problematic uncertainty (H^*, say) which is the maximum uncertainty a student might experience at the instant in question. The scaling of this index is not defined, from that point of view, but the consistency requirement ensures that any instantaneous increment ΔH^* (or $\Delta\eta$) gives rise to an instantaneous increase in ΔH (vice versa for a decrease). There is also a maximum uncertainty $H_{max}{}^*$ (corresponding to η_{max} and evaluated only from the position of an external observer) which limits the maximum increase in H^*.

For systems with many variables (ρ_i, η_i) these quantities are vectorial (as in multivariate information theory: Chapter 1).

Let J represent a transmission (estimate in terms of selective work: Chapter 1) so that J is interpretable as the time integral, over a unit such as a trial block, of an adaptation rate $-dH/dt$. On response to a transient ΔH^*, or for fixed H, the student's loading (Pask and Mallen, 1969), and an *indirect* index of the subjective uncertainty a student experiences, is proportional to

$$H = H^* - J$$

We posit (a) that J must be maintained constant (the notion of an operating region) so that the term dH/dt is negative. Thus, for fixed H^*, the lower or underload limit on the operating region is contravened after a finite time interval. (b) the upper or overload limit is contravened by too large an increase in H^*. We further postulate that the rate of uncertainty reduction depends upon the loading in the single-peaked monotonic manner shown in Fig. 84.

The system is self-organising if, and only if, $dz/dt > 0$ where Z is a redundancy $1 - H/H^*$. This condition may be satisfied (Appendix A), either (i) by a decrease in H for fixed H^* (case (a) above) which has necessarily limited tenure since H approaches zero, (b) by an increase ΔH^* in H^* with H constant (which corresponds to an increase in η). However, provided that ΔH^* is not too large, this increase is compensated by learning and represented as an increase in the rate integral J, or (c) a system may be self organising due to the coordinated or coupled regulation of H and H^* (H changing as a result of the student's *goal-directed* adaptation and H^* due to the tutor or regulator).

This is a macroscopic picture of the tutorial operation in an adaptive system which is designed to maintain a constant loading or a constant value of H, by a variation in H^* guided by a model in the subcontroller that represents the student as an H-reducing mechanism (within the limits of his operating region), and a further model in the overall controller that adjusts η (or H^*) to maximise the H reduction rate.

5.4 The steady-state system, on its own, is still self-organising but is degenerate to the extent that the overall controller model is missing. As a result, self-organisation occurs due to the stepwise juxtaposition of operations (a) and (b) in section 5.3 (not, except by accident, due to (c)).

5.5 One difficulty is that all those identifications rest upon a presupposition that responses are goal-directed and that, on *average*, reduction in uncertainty is due to an increase in *correct* certainty. Another difficulty is that, though the sense (+ or −) of the change in the estimated variables (H, H^*) is veridical, their absolute values are only calculable in respect of an external observer, not in respect of the student.

Both difficulties can be surmounted if the quantities H and H^* are regarded as *indirect* estimates of underlying subjective uncertainty/information/indices and if the values of these subjective indices are used in the control equations or in regulator design.

Under this interpretation, the system is still self-organising. However, the indirect estimates should be replaced whenever *possible* (in the sense of Chapter 1) by direct estimates. If so, values appropriate to the student can

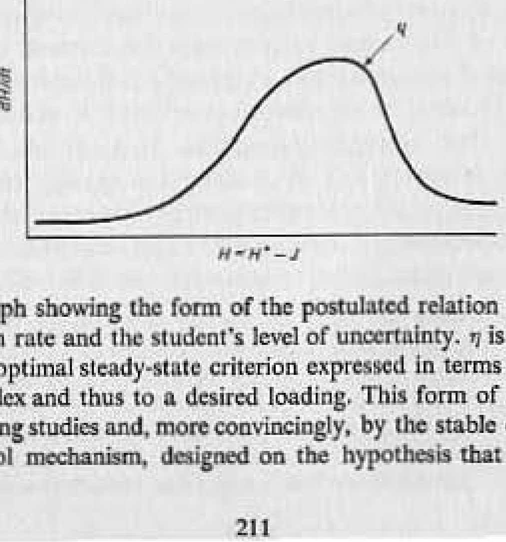

Figure 84 Graph showing the form of the postulated relation between uncertainty reduction rate and the student's level of uncertainty. η is a point corresponding to the optimal steady-state criterion expressed in terms of uncertainty/information/index and thus to a desired loading. This form of relation is supported by loading studies and, more convincingly, by the stable operation of an adaptive control mechanism, designed on the hypothesis that such a relation exists.

be attached to the terms cited and it is also possible, under these circumstances to obtain an index of *correct belief* (Chapter 1). The reality or the relevance of the self-organisation is guaranteed by any scheme that renders any decrease in H, which is used as a *control signal* by the regulator, contingent upon either an increase in correct belief or a constancy of correct belief.

In practice, schemes of this kind work very effectively. They have a much wider field of tutorial application than the domain of structured skills discussed up to this point.

9 Descriptions of Processes

A description, of any sort, entails a language in which the description is expressed and which can be interpreted by someone or something. For instance, one can describe, in natural language, the events going on in one's room or one's mind, in the street, in an aquarium or in a motor car's engine. The widespread idea that a static skeleton of things must first be described (e.g. the pistons of the engine, its cylinders, crank shaft, and so on), and that events become obtrusive later as happenings to *things* is a misconception.[1] Often this approach is convenient and logicians have achieved formal elegance by adopting it. But there is nothing in man or nature to enforce this world view and (as particle physicists, amongst others, have found) it is sometimes better to notice events first and to build things, either real or fictional, around them.

The outstanding distinction between a static description and the description of a process lies in the capabilities assumed to exist in a user (recall, from Chapter 1, that there is only *potential* information in a description, i.e. literally it *has* a pattern, it does not *have* an information). Both static patterns and processes may represent classes ordered by the writing and connection rules of a language or functions and relations. For example, a table specifies a relation and abstract automata may be represented by a table (for instance, *Ru* of section 1.2 in Chapter 2). But if a pattern is depicted then the processor that uses it must be equipped with an order of execution or a specially operated clock (a seeking-out program) that reconstitutes the table in its own storage and refers to it. If a process is represented (the specification *Rul* in contrast to the specification *Ru*, for example), then order of operation is carried as part of the pattern; it prescribes the process to be executed directly and it may prescribe or permit many execution orders.

1. Little or nothing is said, in this volume, about the description of static objects and patterns; still less concerning the important topic of description, *efficiency*. Without an *efficient* description, most problem solving operations and control operations are utterly impracticable. The reader is referred to Banerji (1970, 1971) for a comprehensive as well as elegant discussion of the field and the most lucid available account of what a description *is*. Perhaps the best technical accounts of pattern recognition, learning, and search are found in Ivahnenko and Lappa (1967) and Fu (1968).