SESSION 3

PAPER 5

LEARNING MACHINES

by

DR. A. M. ANDREW

BIOGRAPHICAL NOTE

Alexander M. Andrew, National Physical Laboratory Teddington. Age 33. Studied Mathematics and Natural Philosophy at Glasgow University, later doing research in Physiology. Spent 18 months with Warren McCulloch's group at M.I.T., working on visual pathways in the frog. Recent activities have swung away from biology, but this will be remedied.

LEARNING MACHINES

by

DR. A. M. ANDREW

SUMMARY

THE application of learning machines to process control is discussed. Three approaches to the design of learning machines are shown to have more in common than is immediately apparent. These are (1) based on the use of conditional probabilities, (2) suggested by the idea that biological learning is due to facilitation of synapses and (3) based on existing statistical theory dealing with the optimisation of operating conditions. Although the application of logical-type machines to process control involves formidable complexity, design principles are evolved here for a learning machine which deals with quantitative signal and depends for its operation on the computation of correlation coefficients.

---

1. INTRODUCTION

In this paper learning machines will be discussed with particular reference to industrial applications, but it is believed that the arguments have a bearing on the study of human and animal learning. The extent to which parallels between brains and machines may profitably be drawn has been discussed by many writers including MacKay *(ref. 1)* and Sluckin *(ref. 2)*, and it is clear that extreme caution is required. It does appear, however, that valuable biological experiments are likely to be suggested by a study of the difficulties encountered in attempting to make practical applications of learning machines as similar difficulties have presumably been met and solved at some stage of biological evolution.

1.1 *The definition of learning*

Thorpe *(ref. 3)* provisionally defines learning in animals as "that internal process which manifests itself as adaptive change in individual behaviour as a result of experience." By its inclusion of "adaptive" this definition implies that the learning process is necessarily associated with goal-directed behaviour. The goal may be the acquisition of food when hungry, or drink when thirsty, or the satisfaction of an instinctive drive such as reproduction or care of the young. The animal's behaviour may be interpreted as an attempt to maximise some function of its sensory inflows. Wiener *(ref. 4)* uses the term "affective tone" to describe this function; Selfridge *(ref. 5)* refers to a similar function as "hedony".

In the case of a learning machine to control an industrial process the quantity which it recognises as "hedony" and tries to maximise will be a pre-arranged function of a number of quantities associated with the process. These will include measures of yield and quality of the product and the costs of running the process. The learning machine will keep modifying its policy of control by trial-and-error with the aim of increasing the amount of "hedony". This is, of course, essentially what a person does in acquiring skill in a control task.

1.2 *A type of industrial application*

Since chemical processes lend themselves to automation more readily than do processes involving the handling of solid objects, attention has been paid particularly to the possibility of using a learning machine to control a chemical process. Consequently it will be assumed in the following that the learning machine must deal with quantitative rather than logical information, but may, nevertheless, be able to make binary decisions and to change its own internal connections.

Figure 1. represents a process controlled by a learning machine. The quantities a, b and c may be temperatures, pressures, flow-rates or other variables associated with the process. The learning machine effects control by varying the quantities d and e, which may control valve settings or heating power applied to parts of the process. In order to compute the degree of goal-achievement or hedony, h, the output of the process is measured and probably tested for quality by suitable transducers. The inflow of raw material may also be monitored.

The arrangement of *fig. 1* is considerably different from an ordinary servo-mechanism employing negative feedback. In a servo-mechanism the quantity which indicates the degree of goal-achievement, namely the output error, can be used directly to control the active element. In the arrangement of *fig. 1*, on the other hand, it is unlikely that the quantity h can be used directly to control the process as there will usually be a considerable delay between the occurrence of a variation in d or e and the appearance of the resultant change in h.

In general the only way in which the process can be controlled is by letting d and e be controlled as functions of a, b and c. The quantity h can only enter into the control system in a more subtle way. The mathematical functions relating d and e to a, b and c must contain a number of

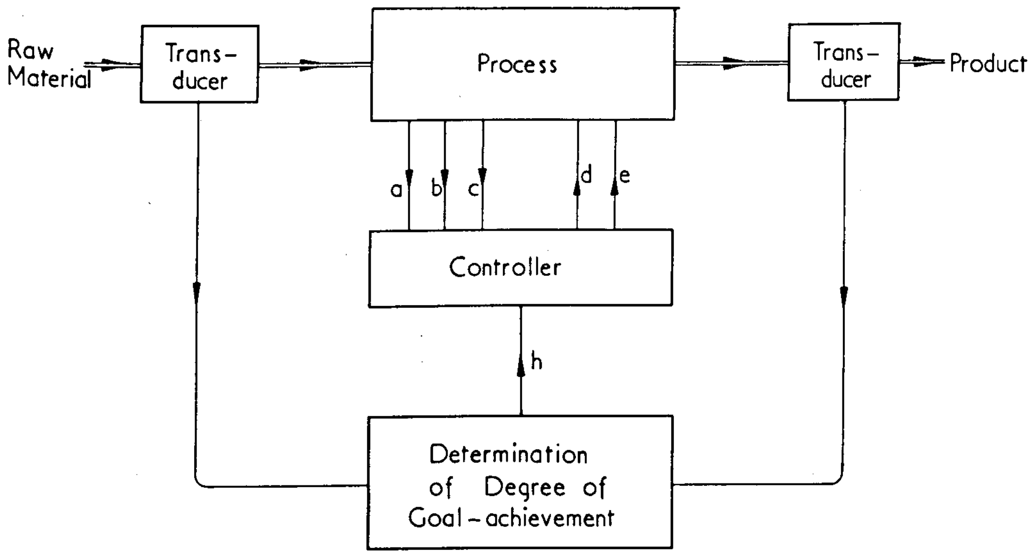

Fig. 1

parameters which can be altered in a trial-and-error fashion so as to maximise *h*. At any instant, therefore, the effect of a high value of *h* is to help set the seal of approval on the forms of the control functions which were operative before it occurred, while the effect of a low value of *h* is the opposite. By such trial-and-error procedures the control functions can be made to approach their optimum forms. It is perhaps worth observing at this point that the optimisation is not necessarily restricted to finding the best values of parameters in the functions; it is possible to devise optimisation procedures which can alter the function more fundamentally, as will be described later.

Although a distinction was drawn between servo-mechanisms and the type of arrangement shown in *fig. 1*, this could represent a self-optimising servo in the case where *a*, *b* and *c* are functions of the input and output quantities of the servo-mechanism, and *h* is some time-averaged monotonically-decreasing function of the servo error.

It is expected that learning machines will allow efficient automatic control of processes which are not amenable to precise mathematical analysis. The use of a learning machine is, in fact, an empirical method for approximating to the optimum control function. Other empirical methods are possible, involving the continuous recording of all relevant quantities associated with the process and the periodic analysis of records by means of a computer. The method employing a learning machine is attractive insofar as trial-and-error adjustment can proceed continuously, and consequently the optimum control function should be approached more rapidly than by other methods. This relatively rapid approach to the optimum could be particularly valuable if the process was influenced by some fluctuating quantity which could not be monitored, but which influenced the optimum control functions.

The principle of operation of a learning machine as described in the foregoing is very similar to Ashby's principle of ultrastability, *(ref. 6)*. Three different approaches to the problem of building a learning machine will be discussed. These appear, at first sight, to have little in common but will be shown to be closely related. Then, certain of the principles which were involved in these approaches will be further elaborated to show how it is hoped to build a practical learning machine suitable for industrial use.

2. FIRST APPROACH TO THE DESIGN OF LEARNING MACHINES. THE CONDITIONAL PROBABILITY COMPUTER

The ideas of probability theory must obviously be involved in any empirical approach to process control, since the aim is to maximise the probability of the desired goal in the future. Since a certain amount of information about the process is available to the controller at any time (in the form of measures of a, b and c in *fig. 1*, for instance) the probability levels which are important are the *conditional* probabilities which are computed subject to the information being as indicated.

A computer has been built by Uttley *(refs. 7,8,9)* and Russell which is specially designed to evaluate conditional probabilities. Some further developments have been described by Russell *(ref. 10)*. This type of computer has a number of input channels which can be activated singly or in groups. The signals in the channels are necessarily in dichotomous ("yes-no") form; each channel at any instant is either activated or not activated. The channels are connected to a classification system (see Uttley *ref. 11*) of a kind which is shown in *fig. 2*. for the case of a 3-channel machine. The capital letters A, B and C have been used to represent the signals in the three channels; the convention will be adopted here of using capital letters for dichotomous quantities and small letters for continuously varying quantities.

The classification system can activate a unit for every possible group of input channels. A machine with n input channels must have $2^n - 1$ units.

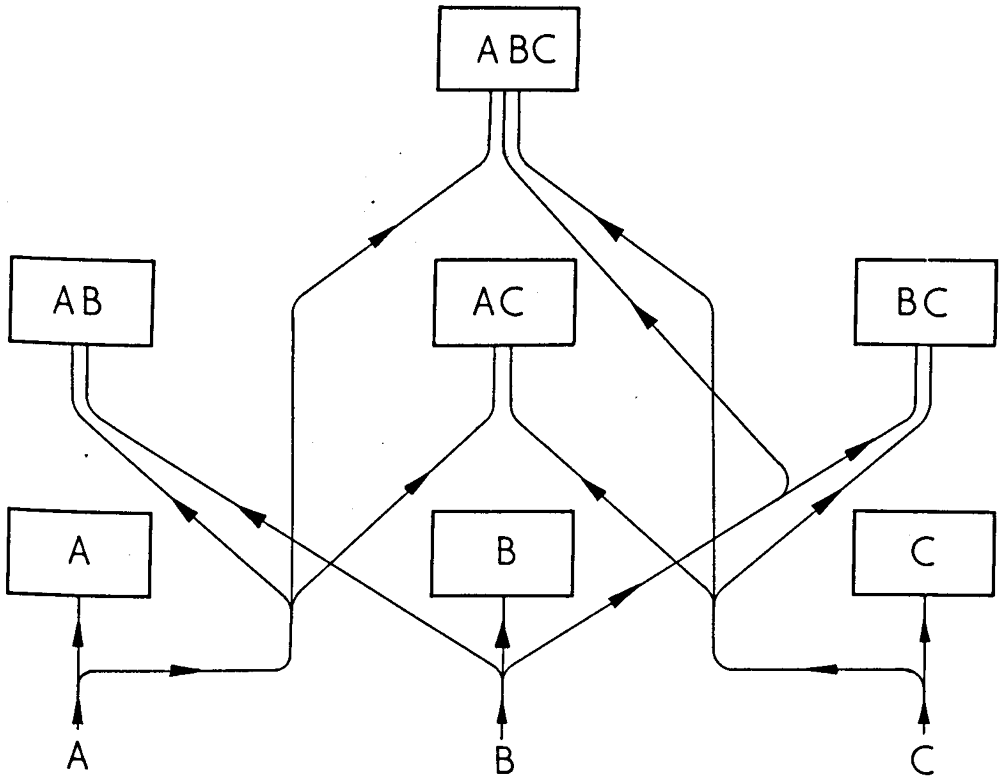

Fig. 2

The practical conditional probability computer which was built by Uttley and Russell has five input channels and consequently 31 units. This computer will be exhibited at the Symposium. In order to compute conditional probabilities, each of these units counts the number of times it is activated. In the existing computer this is done in an analogue fashion by arranging that the charge on a capacitor is altered every time the unit is activated. Then the conditional probability of activity in any particular channel, given activity in some other channel, or group of channels, can be computed as the ratio of the counts stored in a particular pair of units.

Suppose, for instance, that the *A* and *B* channels are active. Then the conditional probability of activity in *C* is given by:-

Probability of *C*, given *A* and *B* =

$$p_{AB}(C) = \frac{\text{count stored in } (ABC) \text{ unit}}{\text{count stored in } (AB) \text{ unit}} \qquad (1)$$

When the A and B inputs of the conditional probability computer are activated simultaneously, it computes and quantity $p_{AB}(C)$ and if there are D and E channels it similarly computes $p_{AB}(D)$ and $p_{AB}(E)$. The computer could have been made so as to indicate the actual values of these probabilities, but it in fact determines whether or not they exceed a predetermined threshold value. If this value is exceeded an inference is made of activity in the corresponding channel or channels C, D or E. The output signals of the computer consist, therefore, of inferences of activity in the same channels as are used to convey the input signals.

2.1 *Goal-seeking by conditional probability*

The conditional probability computer was not designed for the type of industrial application which is considered here, and it is not inherently goal-seeking. There are three main ways in which it can be modified to become goal-seeking. These are illustrated in *figs.* 3, 4 and 5 respectively.

In *figs.* 3 and 4, one of the input channels of the computer is connected so that it is activated when there is an indication of success from the

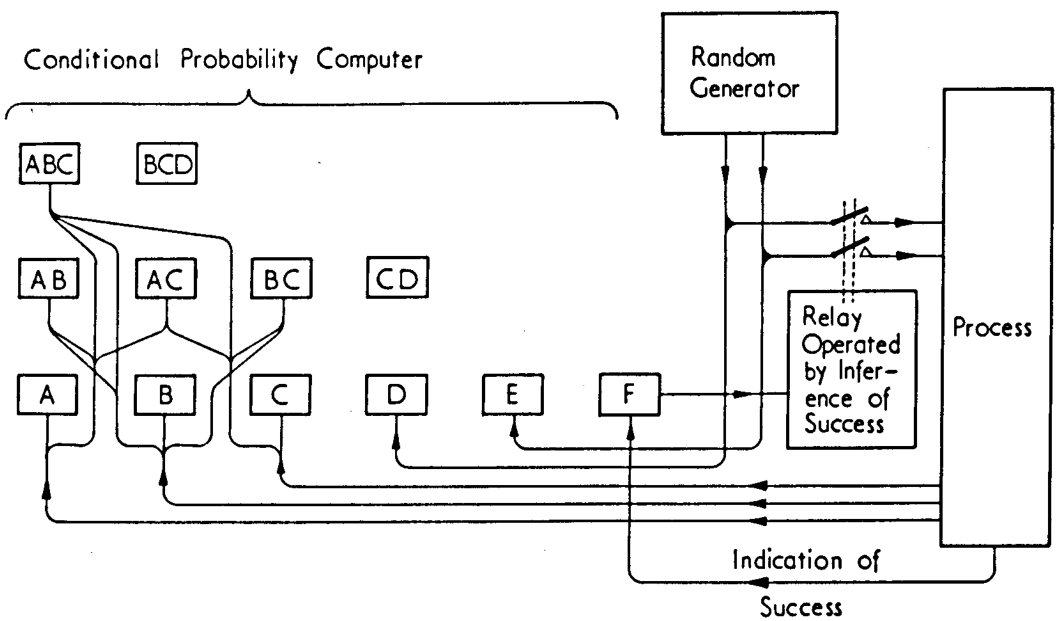

Fig.3. First Type of Goal Seeking

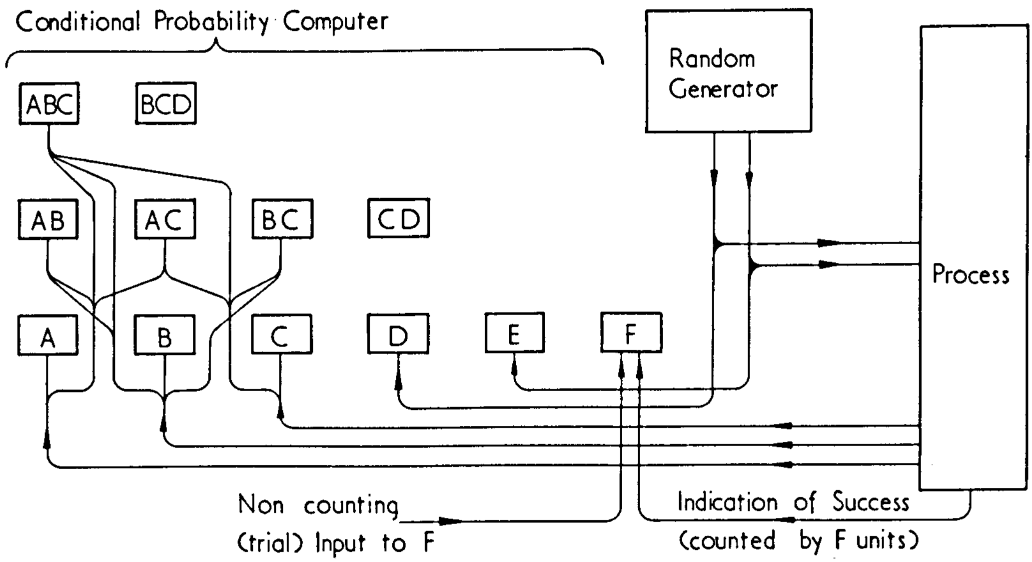

Fig.4. Second Type of Goal Seeking

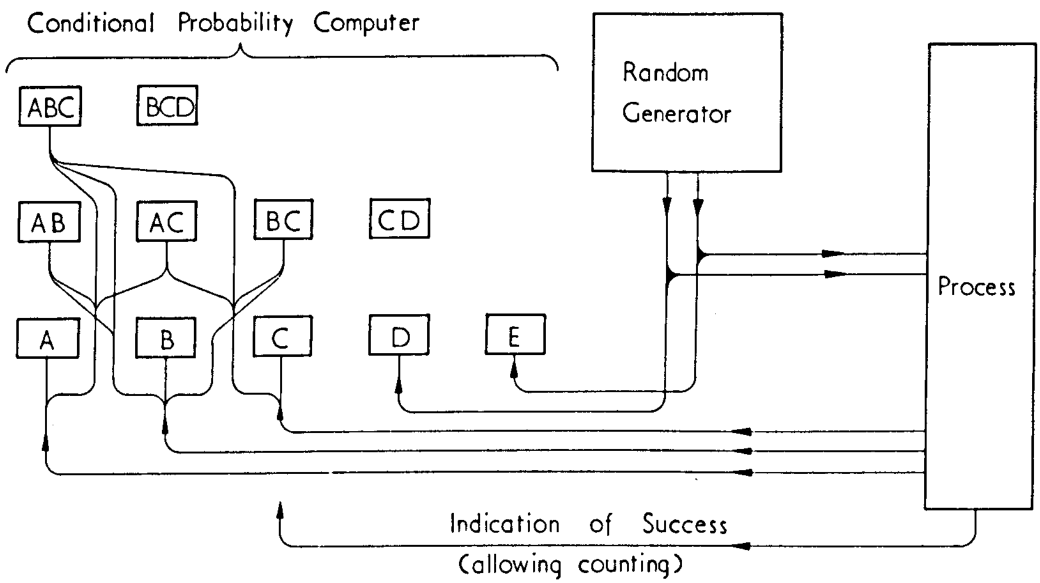

Fig.5. Third Type of Goal Seeking

process. (It is assumed in these arrangements that the indication of success is a yes-no signal, and not a continuous variable like *h* in *fig. 1*). In *fig. 3* the random generator applies different patterns of activity to the *D* and *E* channels, until the whole pattern of activity applied to the *A, B, C, D* and *E* channels is one which the computer associates with activity in channel *F*, which represents success. When *F* becomes active by inference, the relay closes and allows the pattern of activity presented to *D* and *E* to be applied to the process. In order that the computer can learn it is necessary that the relay closes sometimes when an inference of activity in *F* has not been made; it might be arranged, for instance, that if the computer does not make an inference in *F* within a certain time of random searching, the relay, in any case, closes.

In *fig. 4* the pattern of activity in channels *D* and *E* is actually inferred by the computer. It infers the pattern of activity in these channels which it has come to associate with the pattern existing in the *A, B* and *C* channels, and with activity (representing success) in the *F* channel. The input to the *F* channel which is used to invoke inferences is applied in such a way that it does not affect the counts on the units of the computer. If this was not done, the computer would come to associate success with all possible patterns on the *A, B* and *C* channels in combination with an absence of activity in *D* and *E*.

In *fig. 5*, the indication of success is not actually applied to a channel of the computer. Instead, it is arranged that the units of the computer are only allowed to count if there is an indication of success. By this means the computer can learn to associate favourable patterns of activity in *D* and *E* with the patterns which occur in *A, B* and *C*.

The types of goal-seeking shown in *figs. 4* and *5* (2nd and 3rd types) are essentially equivalent to one another. In *fig. 5* the indication of success allows counting in all units, while in *fig. 4* it allows counting in those units which have a connection to the F channel. But since, in *fig. 4*, it is only the units having a connection to the *F* channel which are important in making inferences, *figs. 4* and *5* are essentially the same.

Figure 5, however, indicates how the principle can be extended to the case where the indication of success is a continuous variable such as *h* in *fig. 1*. The indication of success in *fig. 5* could be such a quantity, and might even take negative as well as positive values. The amount by which the counts in the units are altered would be proportional to *h*.

In spite of its rather severe limitations due to having only five on-off channels to convey information to it, the existing conditional probability computer has proved remarkably versatile. When coupled to simple pieces of external apparatus it provides a number of interesting and amusing demonstrations.

3. SECOND APPROACH TO THE DESIGN OF LEARNING MACHINES. FACILITATION OF SYNAPSES

A second approach to the design of learning machines is suggested by the idea that biological learning is due to changes in the properties of synaptic connections between neurons. A learning machine might consist of a number of units capable of activating each other through some kind of linkage, where the properties of the linkages are modified as part of the learning process. There would be a variable quantity associated with each linkage and the effectiveness of the linkage in transferring the activity from one unit to another would depend on the current level of this quantity. The values of these quantities would be continuously modified as functions of the activity of the network.

MacKay *(refs. 1, 12, 13)* has discussed the possibility of building networks in which the linkages do not behave in a deterministic manner, but have a certain probability of transferring activity. In this case the probability of a transfer would be governed by the current level of the variable quantity associated with the linkage. An alternative to MacKay's scheme would be to let the occurrence or non-occurrence of a transfer depend absolutely on the current level of the quantity, according to whether it was greater or less than some threshold value.

In order that this kind of network may learn, the variable quantities associated with the linkages must be suitably modified as functions of the activity of the network. In an application where the network is required to be goal-seeking, an obvious rule for modifying these quantities would be the following:-

If a transfer of activity in a linkage is followed by an indication of goal-achievement, let the effectiveness of the linkage in transferring activity be increased. If the transfer is not followed by goal-achievement, let the effectiveness of the linkage be reduced. An extension of this rule which would probably produce faster learning would be the following:-

> If simultaneous activity in the two units which are coupled by a linkage is followed by goal-achievement, let the effectiveness of the linkage be increased and vice versa. According to this latter rule the effectiveness of the linkage is altered whether or not the simultaneous activity in the units was due to this particular linkage. (It is assumed that the linkage can transfer activity in one direction only).

Other possible rules for varying the effectiveness of the linkages might be devised. It is sometimes suggested that in the nervous system those synapses which are frequently activated become more effective. This would provide a mechanism whereby the system would learn with practice, but it is difficult to relate the principle to goal-directed activity. In any case, recent work by Professor Delisle Burns at McGill University indicates that repeated activation of a synapse has the effect of reducing its effectiveness rather than increasing it.

It has been assumed in the foregoing that a linkage either transfers activity or does not, but a network could contain units in which activity is produced by a summation of effects of a number of linkages, although any one linkage would not produce activity by itself. Such units must certainly be incorporated in any network intended to serve as a model for the nervous system. In the present treatment, however, it will be assumed for simplicity that the transfer of activity by a linkage is an all-or-nothing process.

3.1 *Conditional probability computed by linkages*

Although the conditional probability computer in its present form does not operate by altering the properties of the linkages, a computer could be designed to work in this way and to show equivalent behaviour. In the computer as previously described the storage of information is in the units themselves which count the number of times they are activated non-inferentially.

The storage can equally well be in the linkages but it is necessary in that case to have a larger number of storage locations. For a computer having n input channels, it is necessary to have about 2^n locations if the storage is in the units and $n/2$ times as many if it is in the linkages.

However in the various modifications of the computer which make it goal-seeking, many of the storage locations can be eliminated in either case. It can be shown that, for a goal-seeking computer, the number of storage locations is much the same whichever way the computer is organised.

For a non-goal-seeking conditional probability computer having storage in the linkages, every unit corresponding to a group of channels must be capable of being activated inferentially through a linkage from every unit corresponding to the same group of channels with one channel eliminated. For instance, the unit corresponding to *ACD* must be connected through linkages to the units *AC*, *CD* and *AD*. The unit corresponding to *AB* must be connected through linkages to the units *A* and *B*.

Figure 6 shows two alternative circuits which could be incorporated in the linkages to compute conditional probabilities.

For the purpose of explanation it is assumed that one of these circuits is incorporated in the linkage which can activate the *AB* unit from the *A* unit. In the circuit of *fig. 6* (a), the switch S_1 is normally in the position shown, but goes over to the right for a short time whenever the unit *A* becomes active due to activity in the *A* channel - not by inference. If the *B* unit (and hence the *AB* unit) are similarly active at the same time, the switch S_2 is in its upward position and connects the capacitor of value kC to the supply voltage V_0. On the other hand, if the *B* unit is not active, the switch S_2 connects the capacitor of value kC to earth, and it is discharged. It is clear that if event *A* is *always* accompanied by event *B*,

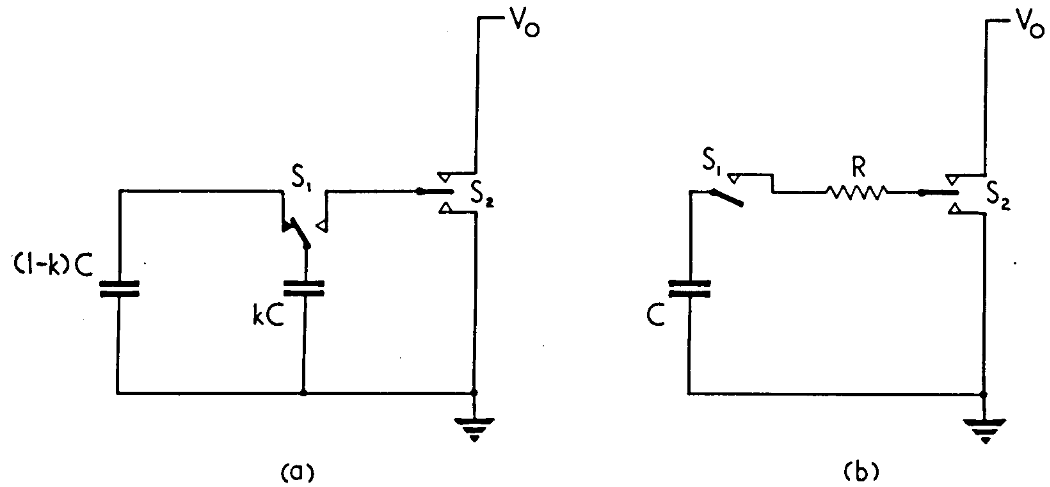

Fig.6

the voltage on the capacitor approaches V_0; if the event A is *never* accompanied by event B, the voltage approaches zero. It can be shown that if V_n is the voltage on the capacitors after the nth occurrence of activity in A,

$$\frac{V_n}{V_o} = (1-r)\left\{ \delta_n + r\delta_{n-1} + r^2\delta_{n-2} + \ldots\ldots \right\} \quad (2)$$

where $\delta_i = 1$ or 0 according to whether or not B was active on the ith occurrence of activity in A.

For the circuit of *fig.6(a)*

$$r = 1-k \quad (3)$$

and normally $k \ll 1$

The expression on the right hand side of Eq. 2 is a reasonable one to take as a "running value" of the conditional probability of B given A. Hence, $\frac{V_n}{V_o}$ as produced by the circuit of *fig.6(a)* is a suitable quantity to take as such a "running value". The behaviour of the circuit *fig.6(b)* is similar. It is required that the switch S_1 in this case closes for a definite time t_1 every time A occurs. Then if the time constant of R and

C is equal to t_2, it can be shown that the circuit of *fig. 6(b)* also conforms to Eq. 2, but in this case

$$r = \exp\left(-\frac{t_1}{t_2}\right) \tag{4}$$

In order that a computer incorporating one of these circuits may make inferences, it is only necessary that the voltage V across the capacitor to the left of the circuit is applied to a comparator which determines whether or not V/V_0 is greater than the chosen threshold value of conditional probability. If V/V_0 is greater than the threshold, the linkage produces inferences. In the case considered above, for instance, if V/V_0 is greater than the threshold, and if input A becomes active, the linkage will energise unit AB by inference (assuming input B is not active and hence unit AB is not activated directly). When unit AB is activated by inference, the computer indicates an inference of activity in channel B.

Circuits other than those in *fig. 6* can be devised to compute conditional probabilities. For an industrial application the storage element would probably not be a capacitor since this would not provide long term storage. The use of magnetic cores in an analogue fashion, as has been done by Pittman *(ref. 14)* is a possibility.

3.2 *Equivalence of approaches 1 and 2*

Since it is possible to design a conditional probability computer which depends for its operation on the modification of linkages, it is considered that approaches 1 and 2 to the design of learning machines are essentially equivalent. *Figure 7* shows a type of learning machine which could have been evolved through either approach. It is essentially a rearrangement of the 3rd type of goal-seeking shown in *fig. 5*, for the case where the conditional probability computer depends on the modification of linkages. (It does, however, have a classification system for the A, B and C inputs which differs from that in *fig. 2*. Instead of an AB unit, there is an $A.B.{\sim}C$ unit, active when the input pattern is "A and B and *not* C". There is no special merit in the system of *fig. 2* in the case where the data in channels A, B and C is reliable, so that it is never necessary to make inferences of A, B or C).

The random generator in *fig. 7* can either be connected to the D and E channels, corresponding to the random generator in *fig. 5*, or it can be applied to the linkages themselves making them behave like the probabilistic connections discussed by MacKay.

4. LIMITATIONS OF LOGICAL-TYPE DEVICES FOR PROCESS CONTROL

The learning machines considered in the last two sections dealt with yes-no information; the input and output channels were either active or inactive. A computer controlling a process, however, must utilise information coming from transducers measuring temperatures, pressures, flows, etc. and it is unlikely that satisfactory control could be effected by a system which treated these measures as yes-no quantities. (The rather clumsy terms "yes-no" and "logical-type" are used here in preference to "binary" in order to avoid possible confusion with pulse-code modulation).

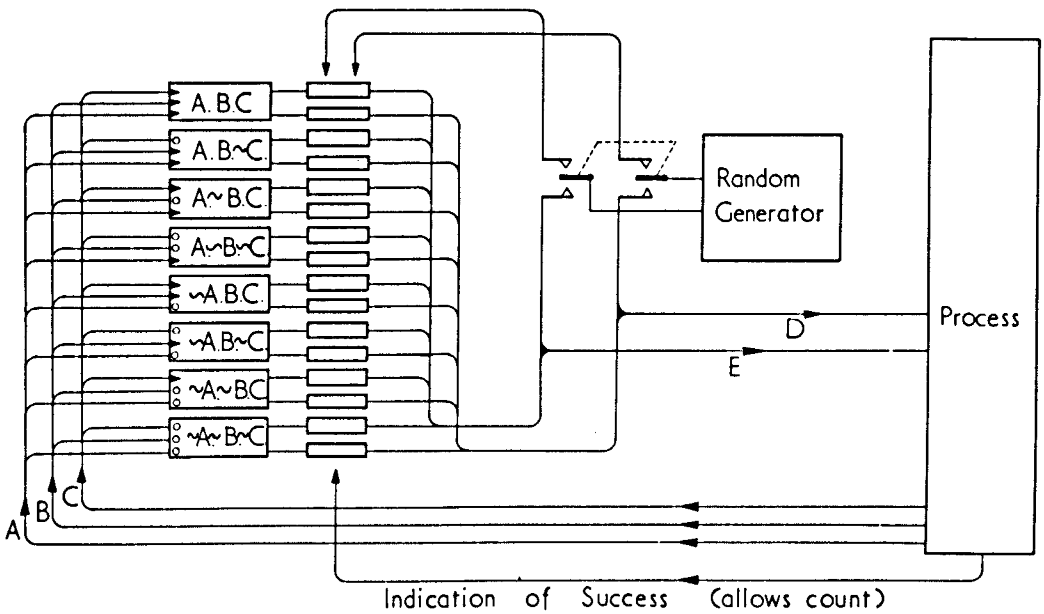

Fig. 7. Modification of Third Type of Goal Seeking

A possible way of extending logical-type machines to take account of quantitative information would be to code the measured quantities in binary form and then to use several of the yes-no channels to convey the information from each transducer. In this way it is possible, in principle, to arrange for a logical-type machine to deal with any sort of input information.

Unfortunately the complexity of the learning machine which is required becomes prohibitive. It would be reasonable to represent each measured quantity by five binary digits. To control a dynamic process, the learning machine must take account of past as well as present values. In order that the machine may take account of values at two-time-displacements as well as the current value, it is necessary that each measured quantity connects with 15 yes-no input channels. Hence, the total number of yes-no channels by which information is fed to the learning machine must be 15 times the number of transducers, and is, therefore, prohibitively large.

4.1 *Achieving simplification*

There are several ways in which some simplification might be achieved. It is possible that in certain applications some of the 15 yes-no channels corresponding to an input quantity could be eliminated. It might sometimes be sufficient to represent the current value of a quantity by five binary channels, and its derivative by two, making seven in all. Simplification in this way is undesirable, however, since information thrown away can never be regained and any reduction in the amount of information supplied to the learning machine can only restrict the range of control functions which it is able to apply.

4.2 *Use of sub-goals*

Two further ways in which some simplification can be achieved are by the use of sub-goals and by interpolation. Any application of a learning controller to a process of any complexity would involve the use of sub-goals. Parts of the process would be linked to a sub-section of the controller in the general manner represented by *fig. 1* but the quantity recognised as "hedony" and maximised by this part of the controller would not be the same as the overall goal of the complete controller. In controlling a chemical process, for instance, the overall goal would be expressed in terms of the yield and quality of the product, but a sub-goal might be to maintain a particular temperature-distribution in a fractionating column. If the sub-goals were determined in advance by the designer of the controller, the use of sub-goals would be a departure from the learning principle. In fact, however, the sub-goals could be modified in a trial-and-error fashion by that part of the controller (which may by termed the "master" controller) whose goal is the overall goal of the process.

A complicated controller might actually involve a hierarchy of learning machines of which one was the "master" or first-order controller. This would have, as its goal, the overall goal of the process, and it would set the goals for the second-order controllers, some of which would, in turn, set goals for third-order controllers and so on. The use of subsidiary goals can lead to a considerable reduction in the complexity of a learning controller.

4.3 *Use of interpolation*

The other principle which could be used to reduce complexity is interpolation. It was assumed earlier that five yes-no channels are needed to convey the signal from a transducer, or fifteen to convey the current value as well as the values at two time-displacements. The system of interpolation which is proposed would let each signal be represented by a smaller number of yes-no channels, which might be two to represent a single signal, or six when time-displaced values are represented. Then the two yes-no signals can define four levels within the range of variation of the measured quantity. Suppose, for example, that the measured quantity is a temperature which can vary between 40 and 100 degrees. Then 40 degrees would be represented by 0,0 in the yes-no channels, 60 degrees by 0,1 in them, 80 degrees by 1,0 and 100 degrees by 1,1. At any instant in time it will generally be found that the temperature is not exactly 40, 60, 80 or 100 degrees. The learning machine must decide what action it would take for the nearest of these values of temperature above and below the given value. For example, if the temperature was 63 degrees, the controller would work out control actions which would be appropriate if (a) the value was 60 degrees and (b) the value was 80 degrees. Then a weighted average of (a) and (b) would be taken, with more weight given to (a) than to (b) since 63 is closer to 60 than to 80. This weighted average would be the control action exercised by the controller.

In the general case, where a number of variables are represented by the activity in two yes-no channels, a more complicated form of interpolation must be used. Let n be the number of variables, including time-displaced versions. Instead of determining what the control action would be in each of two situations, it is now necessary to determine what would be the control action in each of 2^n situations. Then a weighted average is taken of the control actions indicated for each of the 2^n situations. The weighting given to any indication would be governed by the degree of approximation of the corresponding situation to the true situation.

When interpolation is used, the process by which the learning machine adjusts the storage in its units or linkages must be modified to take account of values of the variables which are not exactly represented by the yes-no channels. For example, consider again a variable taking the value of 63 degrees, where 60 degrees is represented by 0,1 and 80 degrees by 1,0.

Then the storage locations which can be modified as a result of the activity involving this measure will include those which would have been modified had the value been 60 degrees as well as those which would have been modified if the value had been 80 degrees. The former group will be modified more than the latter, since 63 is closer to 60 than to 80.

By the use of sub-goals and interpolation, a considerable saving in complexity is possible in the application of a logical-type learning machine to process control. However, the necessary complexity is still high and it appears that a more elegant solution can be reached by departing from the logical mode of operation.

4.4 *Time taken to learn*

The reduction in complexity which is achieved by the use of sub-goals and interpolation is valuable insofar as it makes the learning machine more nearly feasible from the point of view of cost and size; it is also valuable in reducing the time taken for the machine to learn. A logical-type learning machine having, say, 30 yes-no inputs would have to learn how to behave in each of the 2^{30} situations which might arise (except for those which are physically impossible). Since 2^{30} is approximately 10^9 and since each situation must arise a number of times before the machine learns how to deal with it, it is clear that the learning process will take a long time. Even if unlimited resources were available for the construction of a learning machine, therefore, the complexity would still have to be kept within bounds in order that learning would not take too long.

4.5 *Numerical estimates of complexity*

It is instructive to calculate the number of units which are needed in a learning machine of the logical type having various numbers of yes-no input channels. In *Table 1*, let n be the number of input channels. The values shown are multiples of 6 or 15, since these are the numbers of channels required to represent one variable in the cases of operation with and without interpolation.

The second column in *Table 1* shows 2^n. This is the minimum possible number of storage locations in a learning machine of the type shown in *fig. 7* having n yes-no input channels. Actually, the number of locations could only be as small as 2^n in the case where the controller had only one yes-no output channel through which to control the process.

Since the number of storage locations tends to be large, it is desirable to consider a cheap form of storage such as magnetic tape. The quantities held in the various storage locations could be represented by binary digits recorded on tape. This would require at least 10 digits per location (see below). The number of digits is therefore 10×2^n and is shown in column 3 of *Table 1*. Column 4 shows the length of magnetic tape which would be needed assuming a packing density of 100 digits per inch.

TABLE 1

Note: 1 light year = 5.878×10^{12} miles

No. of channels n	Min. no. of locations 2^n	No. of binary digits 10×2^n	Length of magnetic tape
6	64	640	6 inches
12	4,096	40,960	34 feet
15	32,768	327,680	260 feet
24	1.6×10^7	1.6×10^8	25 miles
30	10^9	10^{10}	1,600 miles
42	4×10^{12}	4×10^{13}	6,300,000 miles
60	10^{18}	10^{19}	0.27 light-year
72	4×10^{21}	4×10^{22}	1,074 light-years
84	1.6×10^{25}	1.6×10^{26}	4,295,000 light-years

It is clear that the logical type of learning machine becomes prohibitively complex when the number of yes-no channels is more than about 20, and even for 12 channels the complexity is considerable. It is important to remember that the number of storage locations as shown in column 2 is a minimum based on the assumption that the output is by a single yes-no channel. In general the number will be considerably greater, and the figures in columns 3 and 4 would be multiplied correspondingly.

4.6 *Retrospective indication of goal-achievement*

A source of complication which has not hitherto been mentioned is the fact that it will not generally be possible to assume any particular value for the time-lag between a control action by the controller (see *fig. 1*) and the appearance of an effect in the indication of "hedony". For most processes, in fact, the effect of a control action at any instant will be spread over a period of time, and the nature of the effect may be different at different times within this period. A learning machine used to control such a process must have several quantities stored in each "storage location". In the learning machine of *fig. 7*, for instance, each of the linkages would store a number of quantities which might be (a) a quantity relating to the probability of success immediately following a control action involving the linkage, (b) a quantity relating to the probability of success 30 mins. after such an action and (c) a quantity relating to the probability of success 60 minutes after the action.

The values of 30 and 60 minutes for the lags have, of course, been chosen arbitrarily; in a practical application the values would be chosen with regard to the time-constants of the process. It would be advantageous to store many more than three quantities and the ideal would be to store a continuous function of time. Here as elsewhere, a compromise is necessary to keep complexity within bounds.

Returning to the learning machine of *fig. 7*, the way in which learning would take place would be as follows.

A suitable memory, such as a continuous loop of magnetic tape, would be incorporated. This would contain, at any instant, a record of the input and output signals of the controller over the previous 60 minutes of operation. When a "success" signal was produced by the process, the quantity described under (a) would be modified in those linkages which had been active immediately previously, and simultaneously the quantity described under (b) would be modified in those linkages which according to the record had been active 30 mins. previously and similarly for the quantities described under (c).

The "decision" of a linkage whether or not to give an output at any instant when its input is activated must depend on a function of the three quantities listed under (a), (b) and (c). The function must be such that the linkage would become active in the case where one of the three quantities indicated that activity in the linkage was usually followed by good results in 0, 30 or 60 minutes; and the other two of the quantities made no significant indication. On the other hand the function must also be such that the linkage *would not* become active if one of the quantities indicated that activity in the linkage was consistently followed by lack of success after some interval. This latter requirement would hold, even when the other stored quantities indicated that good results were usually obtained after other time intervals.

It was because of the need to retain several quantities in each storage location that the number of binary digits to represent a storage location in digital form was taken to be as many as ten (see *Table 1*). If only one quantity was stored three or four binary digits would be sufficient.

It is clear from the foregoing that a logical-type learning machine suitable for process control is likely to be of formidable complexity. There is, however, no reason for restricting learning machines to the logical type of operation. In fact, it will be shown that machines which deal with quantitative information (by means other than converting it to binary form) will probably prove more suitable for application to process control.

Logical-type machines have been discussed at length for three reasons. These are (a) because much of the previous discussion of learning machines by other writers has been devoted to machines of this sort and (b) because it is not impossible that logical-type machines may find practical applications even though they are apparently less suitable for chemical process control and (c) because quantitative learning machines can be regarded as a development from the logical type. The following section will be concerned with (c). It will be shown how an attempt to reduce the complexity involved in building a practical logical-type machine can lead to a scheme for a quantitative learning machine.

5. EXTENSION TO QUANTITATIVE OPERATION

Two types of classification system have been mentioned, namely the type shown in *fig. 2*, in which a given pattern of activity in the input channels can activate a number of units, and the type used in *fig. 7*, in which each unit corresponds uniquely to a particular pattern of activity. A possible way of achieving some simplification would be to omit some of the higher order units in a classification system of the type of *fig. 2*. Suppose this is carried to the limit where there is no classification at all but only direct connections to the input channels. (That is to say, all the units in *fig. 2* except those in the lowest row have been eliminated).

Suppose further that the input channels are connected through linkages to the output channels in the way that the units are connected in *fig. 7*. In this the outputs of the different linkages which affect a particular output channel are combined in an "or" fashion. In the simplified learning machine which is now proposed it is obvious that if the outputs of the linkages were combined in an "or" fashion the machine would not be able to make full use of its input information, since, as soon as the output channel was activated by activity in one input channel, it would become independent of all the other input channels. If the outputs of the linkages are summated instead of being combined in an "or" fashion, the machine can make better use of its input information. For a machine using summation, the quantity or quantities stored in the linkages would determine the magnitude of the contribution of the linkage to the summation. Thus the linkages would have a different function from that considered previously, which was to determine from the stored quantities whether or not activity should be transmitted at all. For the machine using summation, an obvious extension is to let the inputs to the units be quantitative instead of yes-no signals, and to have the output signals of the units proportional to the inputs. The ratio between the output and input signals of a linkage would be a function of the quantities stored in the linkage, which are modified by the learning process. The controlling action of a learning machine, as now proposed, would be represented by a set of equations of the form

$$d = La + Mb + Nc \qquad (5)$$

where d is an output signal, and a, b, and c are three input signals of the controller, as shown in *fig. 1*. L, M and N depend on the quantities stored in the appropriate linkages of the learning machine. In practice a "constant term" K would also be included. K would also be modified by the learning process. Equation (5) becomes

$$d = K + La + Mb + Nc \tag{6}$$

Equation (5), of course, simply expresses d as a linear function of a, b, and c. The aim of the learning process is to adjust the values of K, L, M, and N so as to make this linear function approximate as closely as possible to the optimum control function for the process.

5.1 *The use of correlation*

The way in which the parameters K, L, M and N could be modified is by making experimental variations in their values while the process is running, and correlating these variations with the variations in the degree of goal-achievement or "hedony" represented by h in *fig. 1.* If there is a significant positive correlation between the variations of one of the parameters, say M, and the variations in h, the learning machine must make an increase in the value of M. Conversely, if significant negative correlation is observed, the machine must reduce the value of M.

5.2 *Correlation as an extension of the conditional probability principle*

This process of self-adjustment by correlation will be discussed in more detail in the next two sections. It is interesting to note that the computations carried out by the linkages in *fig. 7* are essentially correlations. The "transfer function" of each linkage (corresponding to M above) can take the value 0 or 1 in *fig. 7.* Counting in the units is allowed, only when a "success" signal is present. This means that the dichotomous "transfer function" is multiplied by the "success" indication, which is a dichotomous hedony signal, and a time-weighted average of the product is computed in the linkage. The time-weighted average of the product is essentially a correlation between transfer function and hedony. Strictly speaking, the quantity computed is not the normalised correlation coefficient defined by

$$r = \frac{\Sigma XY}{\sqrt{(\Sigma X^2 . \Sigma Y^2)}} \tag{7}$$

since in this definition the quantities X and Y must consist of fluctuations about mean values. The quantities "hedony" and "transfer function", which may be represented in their dichotomous form by H and M, do not necessarily consist of fluctuations about means.

However, the quantity computed is of the form

$$r = \Sigma H . M . W(t) \tag{8}$$

where *W(t)* is a time-weighting function, Equation 8 essentially represents a correlation (see Licklider, *ref. 15* for a discussion of running correlation functions). Hence the use of correlation in a quantitative learning machine is a direct extension of the principles devised for logical-type learning machines.

6. THIRD APPROACH TO THE DESIGN OF LEARNING MACHINES. EVOLUTIONARY OPERATION

Any attempt to apply learning principles in industry should certainly utilise the large amount of statistical theory which has been discussed by Box under the heading of Evolutionary Operation *(ref. 16)*. The theory is described in greater detail in Chapter 11 of the book edited by Davies *(ref. 17)* and in other publications by Box and his colleagues. *(refs. 18, 19, 20, 21* and others*)*.

Box and his co-workers are concerned with the determination of optimum conditions for a process where the variables are quantities such as temperature, pressure, time of reaction, proportions of constituents and so on. *Figures 8* and *9* show two ways in which the yield (or "hedony") of a process may be plotted as a function of these variables. Box *et al* describe empirical procedures which can be used (without

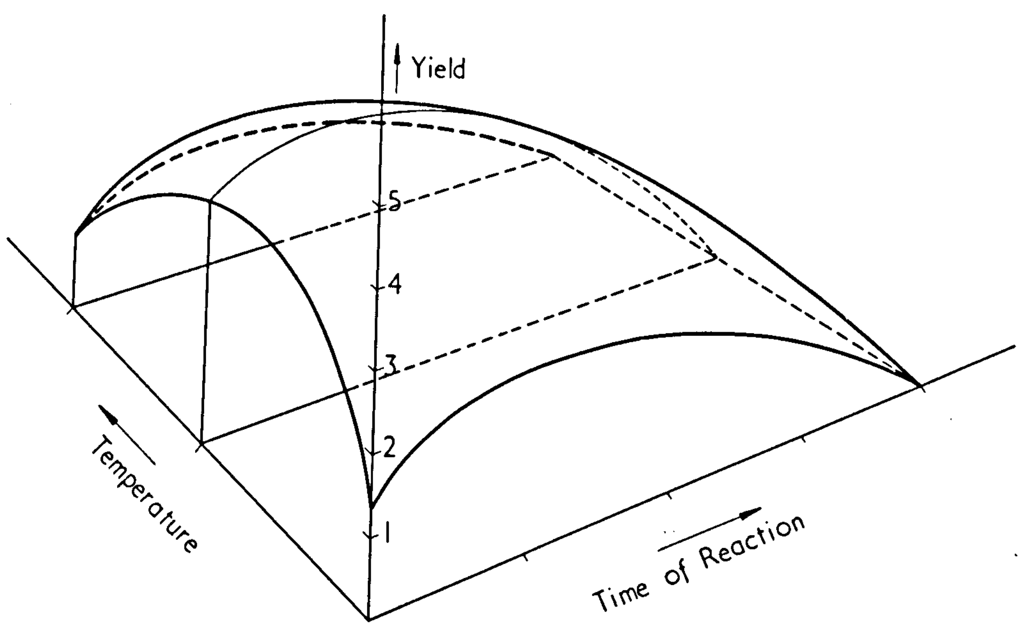

Fig.8. Yield Surface

exploring all possible combinations of values of the variables) to arrive at those values which bring the yield to its maximum.

There seems to be no reason why the same principles should not be applied to the optimisation of a control function. Any control function may be represented to any desired accuracy by a set of polynomial equations with a sufficient number of terms. There would be one such equation for each channel through which the controller was able to affect the process. The polynomial equations would be of the same form as Equation 6, with the addition of higher order terms:-

$$d = K + La + Mb + Nc + Pa^2 + Qb^2 + Rc^2 + Sab + Tbc + Vca + Va^2b + Wabc + \ldots\ldots \qquad (9)$$

where the optimisation procedure is applied to find the best values of K, L, M etc.

For simplicity, only the first order terms would be considered in a first approximation, and higher order terms would be introduced if the first approximation proved inadequate. So long as only the first order terms are considered, Equation 9 is identical with Equation 6, and the procedure for optimisation which can be derived from Box's theory is essentially that discussed in the previous section.

The application of Box's theory should make it possible to decide how, if at all, the variations in the parameters K, L, M etc. should be related to one another. One thing which is certain is that the different parameters should vary at the same time, rather than remaining constant while one at a time is varied. The realisation that factors should vary simultaneously is fundamental to the work of Box and is discussed by Fisher *(ref. 22)*.

Box arrives at procedures for conducting a set of experiments in the neighbourhood of a particular set of operating conditions, and for determining therefrom the way in which the conditions should be altered so as to approach the optimum. In *fig. 9* a particular set of operating conditions would be represented by a point P in the plane of the diagram. Suppose tests are made at each of two levels of each of the two variables. Then tests could be made under four different sets of operating conditions, which would be represented in the figure by the corners of a rectangle drawn round or near P. However, in order to determine the directions in which the operating conditions should be modified in order to approach the optimum, it is only strictly necessary to perform experiments at three of these points. Similarly, for three variables which are each tested at two levels, it is only necessary to conduct five out of the eight possible experiments. *(See ref. 17 p.507)*.

A learning machine seeking to optimise its control function by adjustment of a number of variables might advantageously be made to relate the experimental fluctuations in the variables. If the variables were three in number it would be made to test five out of the eight possible combinations

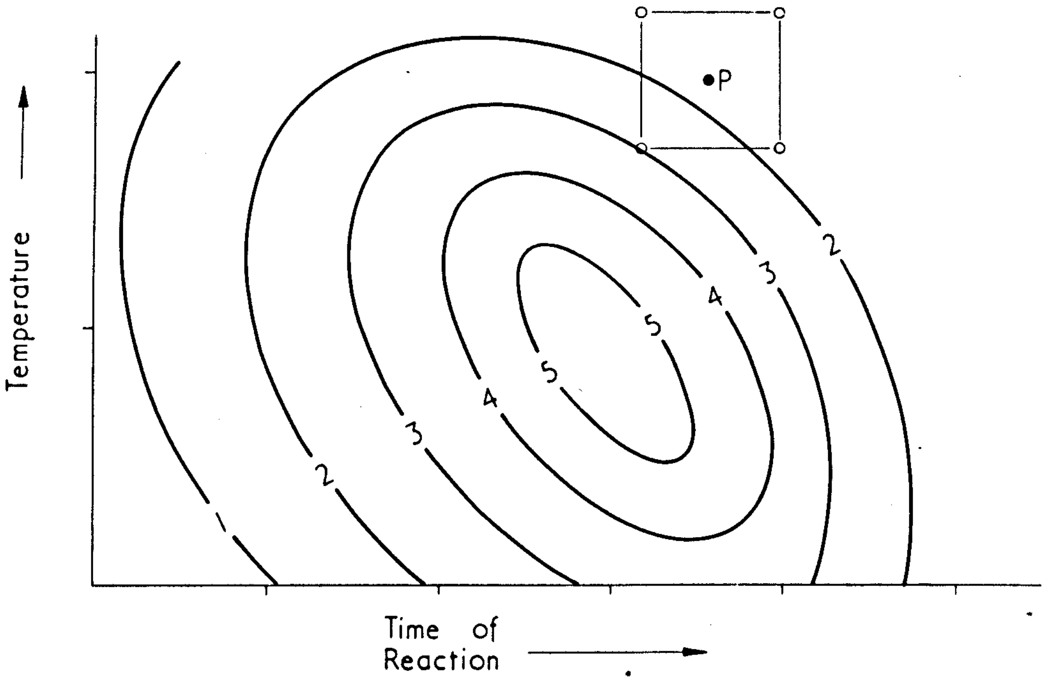

Fig. 9. Yield Contours

of signs of the fluctuations. This system would probably produce a more rapid approach to the optimum than completely random fluctuations would. This is a point which requires further investigation however. It is possible that a planned pattern of fluctuations is only advantageous when the errors of the individual observations of "hedony" are low; this is unlikely to be true in practical applications of learning machines to process control. Nevertheless it is clear that the statistical theory due to Box provides the basis for planning the experimental strategy of an industrial learning machine.

6.1 *Some other possibilities.*

A learning machine for process control could be made to operate by means other than by optimising control functions of the form of Equation 9. Other forms of empirical approximation could be used. It is also possible to devise learning procedures in which the aim is to represent the process as well as possible by mathematical expressions. From the mathematical model of the process it is straightforward to derive optimum control functions.

Kalman *(ref. 23)* has built a learning machine which controls a process by first deriving a mathematical model. The type of process he considers is one having one input channel and one output channel, but since it is a dynamic process, the controller must take account of signal levels over a range of time. Kalman represents the process by the equation

$$C_k + b_1 C_{k-1} + \ldots . b_n C_{k-n} = a_0 m_k + a_1 m_{k-1} + \ldots . . a_q m_{k-q} \ldots \qquad (10)$$

where the a_i and b_i are arbitrary constants and the b_0 has been set arbitrarily to unity. The m_i represent instantaneous values of the control effort applied by the controller to the process, at times kT, $(k-1)T$..... $(k-q)T$, where T is a fixed interval of time. The C_i represent the response of the process at times $kT, (k-1)T$.... $(k-n)T$.

Wilby and Woodcock *(ref. 24)* are building a self-optimising filter, which can adjust itself to simulate a process. This filter therefore lends itself to incorporation in a controller which, like that of Kalman, makes use of a mathematical model of the process. The type of controller proposed here does not incorporate a mathematical model of the process. The model is an intermediate step between the observations which the machine makes of the process and the control function it applies. In general, it is in the interests of simplicity to eliminate this intermediate step. There is one big advantage in having the mathematical model, however, since the learning involved in forming the model is still useful even if the goal of the learning machine is changed. In some applications of learning machines, therefore, machines incorporating a mathematical model of the process will be preferred.

7. FURTHER TRICKS WITH CORRELATION

7.1 *Amplitude of the fluctuations*

As already described, the values of the parameters K, L, M ..... are adjusted in value by making small fluctuations in their values, and correlating these with variations in the "hedony" or degree of goal-achievement, h. The obvious way of deciding the magnitude of the fluctuations is to start with very small ones and then to increase their magnitude until the value obtained for the correlation is statistically significant.

If the parameter is already at a value which gives a maximum of h, there may be no significant correlation even when the fluctuations are large. It is desirable that the machine does not continue these large fluctuations indefinitely, since they must affect the process adversely.

These large fluctuations can be avoided by computing two distinct correlations for each parameter. These are (a) the correlation of the parameter itself with hedony, as already mentioned, and (b) the correlation of the *modulus* of the fluctuations of the parameter with hedony. The magnitude of the fluctuations should increase until (a) becomes statistically significant or (b) becomes significant with negative sign.

7.2 *Self-reorganisation*

There are further correlations which may profitably be computed in the machine. Since the correlation between a parameter and the hedony appears as a continuous or "running" value, it is possible to compute a correlation between this correlation and some other variable. We had in Equation 6:-

$$d = K + La + Mb + Nc$$

Let r_{Lh} be the running value of the correlation between L and h when L fluctuates. Then if a significant correlation exists between r_{Lh} and a, the conclusion can be drawn by the machine that the equation for d could profitably include a term in a^2. Similarly, if there is a significant correlation between r_{Lh} and b, the equation could profitably include a term in ab, and so on.

With the addition of these terms, the equation becomes

$$d = K + La + Mb + Nc + Pa^2 + Qab \qquad (11)$$

If now there is a significant correlation between r_{Qh} and c, for instance, the machine may add a term in abc, and so on.

A learning machine may therefore incorporate self-reorganisation. It incorporates a fixed number of units, and may decide by the above procedure how these should be allocated to the computation of higher-order terms. It is also possible to devise criteria which the machine may apply in order to decide whether an already-allocated unit is serving a useful purpose. If it is not, it can be re-allocated. (One possible criterion to decide whether a unit is functioning usefully is to examine whether the fluctuations in its parameter, subject to the conditions discussed earlier, become large enough to involve changes in sign of the parameter in the course of the fluctuations. If they do, the unit is serving no useful purpose).

The development of the circuitry needed to allocate units to different jobs presents interesting problems. Circuits involving relays and uniselectors can be used in much the same way as they are used by telephone engineers in allocating telephone channels to subscribers. A type of static circuit which may be suitable has recently been described by Chapman and Freed *(ref. 25)* as a possible model for neuronal connections in the spinal cord.

7.3 *Composite goals*

In general, a process must be controlled so as to achieve a composite goal, or in other words to work towards a number of goals which must be balanced against one another in importance. Many human and animal activities similarly involve multiple goals; for example a man driving a car from *A* to *B* has the goal of avoiding an accident, but also has the goal of getting from *A* to *B* as quickly as possible. The two goals conflict to some extent and the driver's conduct is determined by some sort of estimate of their relative importance.

In an industrial application, the main goals will usually be (a) to maximise the quantity of the product, (b) to keep the quality within specification and (c) to minimise wear and tear of the equipment. It will not usually be difficult to devise a function of these goals which may appropriately be taken as the "hedony", since all industry has one main aim which is the maximisation of profits.

In this paper it has hitherto been assumed that the learning machine has no indication of the degree of goal-achievement except the indication of the value of "hedony". It has been assumed that the learning process depends entirely on correlations with this function. However, if no account is taken of the different components of the hedony, information which would appear to be significant is being ignored. It seems clear that the learning process would approach the optimum function more rapidly if the machine was kept informed, not only of its degree of goal-achievevent, but also of the precise way in which it has failed to achieve the goal.

It therefore appears that the machine should make use of correlations with the quantities which contribute to the hedony function, but the writer has not yet arrived at a scheme for doing this. This is one of the many aspects of the development of learning machines which require further consideration.

8. LEARNING BY WATCHING

A learning machine applied to an industrial process could not be coupled to the process while in a completely naive state. If its initial control actions were entirely random the consequences for the process might be disastrous. In many cases it would be possible to pre-set the controller so that it controls the process in a safe and reasonable fashion before any learning occurs. Then the learning process can operate to bring the control function nearer to the optimum.

As an alternative to pre-setting the controller manually, it might be preferable in some cases to couple the learning machine to the process while the process is under manual control. In this way the machine could, in effect, learn by watching the human operator. After a sufficient period of learning by watching the machine could be put in control.

A learning machine may be made to learn by watching in either of two ways. It may simply learn to imitate the control policy of the human operator. In this case the machine takes no account of the degree of goal-achievement of the process during the period of watching. The other possibility is that the machine observes the variations which occur in the operator's policy of control, and tries to correlate them with the variations in degree of goal-achievement. Then, when the machine is put in control its policy of control can be more consistently favourable than was the policy of the operator.

8.1. *Imitative learning*

The simpler form of learning by watching can be achieved by a straightforward adaptation of existing statistical theory. A linear approximation (of the form of Eq. 6) to the control policy used by an operator in controlling d as a function of a, b and c is given by the equation given in statistical texts for the regression of d on a, b and c. (see for example, Weatherburn, *ref. 26)*.

The equation is:-

$$\begin{vmatrix} \frac{d}{\sigma_d} & \frac{a}{\sigma_a} & \frac{b}{\sigma_c} & \frac{c}{\sigma_c} \\ r_{da} & 1 & r_{ba} & r_{ca} \\ r_{db} & r_{ab} & 1 & r_{cb} \\ r_{dc} & r_{ac} & r_{bc} & 1 \end{vmatrix} = 0 \qquad (12)$$

where σ_i is the standard deviation of i, and r_{ij} is the correlation coefficient of i and j.

Since, to control a dynamic process, the measured variables $(a, b, c$ in Eq. 12) must include past as well as present values, the order of determinant required in the regression equation will generally be inconveniently high. It is unlikely that a learning machine would incorporate the means of expanding the determinant, but it would certainly incorporate the means of computing the correlation coefficients and standard deviations. The conversion of the determinantal equation into the form of Eq. 6 would be done once and for all when the learning machine was changed from learning by watching to controlling. The computation involved in the conversion would probably be carried out manually or by a digital computer.

8.2 *Learning by watching with regard to goal-achievement*

For the more advanced form of learning by watching it is necessary that the regression equation be continuously computed and expanded into the form of Eqn. 6. Then the fluctuations of the coefficients in the resulting continuously-varying equation would be correlated with the fluctuations in goal-achievement. The mean values of these coefficients would also be computed. The values of coefficients which would be tried by the machine when it was put in control would be these mean values plus modifications. The signs of the modifications would be determined by the computed values of correlation between the goal-achievement and the fluctuations of the coefficients during the period of learning by watching.

9. SOME PRACTICAL DIFFICULTIES

General principles have been discussed for learning machines for industrial applications, and it is hoped that these will lead to practical forms. Readers who have followed the discussion this far will have realised, or at least suspected, that many practical difficulties have been omitted from the discussion. One of these is the difficulty of devising suitable functions to be taken as "running values" of correlation etc.

Equation 2 gives a suitable function to represent a running value of conditional probability. There is the difficulty here, however, that the time-constant of the averaging process (controlled by the value of r in Equation 2) must be arbitrarily chosen. The use of different values of r will affect the "temperament" of the machine.

The difficulty is greater when a more complicated function must be computed as a running value. In some cases it will be necessary to utilise running values of certain variables in order to compute the value of another variable. In this case a number of different time-constants are involved, and their ratios must be suitably chosen.

In computing a running value of correlation coefficient, the quantities which are multiplied must be fluctuations about mean values, and then the product is smoothed with a certain time-constant. The mean values are themselves running values obtained by smoothing the original signals with a certain time-constant. In what has been written here it has been assumed throughout that the correlation coefficients are normalised so as to lie between -1 and +1. The need for normalisation complicates the issue still further.

In fact, the computation of running values is too involved a subject to be discussed fully here. It will form the subject of a separate publication.

10 CONCLUSIONS

The writer hopes to apply the principles of learning by correlation to an industrial application. The general principles have been discussed here but there are many aspects which require further consideration. There are also many avenues of speculation which have been left unexplored.

Although reference has been made to several other workers, the present paper has not reviewed anything like all of the literature on learning machines. The emphasis has been on quantitative rather than logical-type machines because it appears that the quantitative type will find application more readily. However, it is very likely that logical-type machines will also find applications. Among the logical-type learning machines which have not been discussed are the computer programmes demonstrated by Oettinger *(ref. 27)* and the model of animal learning described by Deutsch *(ref. 28)*. Another type of computer programme with learning properties has recently been described by Friedberg. *(ref. 29)*.

Although little has been said about human and animal learning, it is felt that the principles discussed here are highly significant for psychologists. Their significance does not arise from any suggestion that the learning machines are models for animal learning unless in an extremely general way. The usefulness of the present discussion to psychology is likely to be in suggesting hypotheses and experiments rather than explanations.

I am indebted to the Director of the National Physical Laboratory for permission to publish this paper, to Dr. Uttley and to my other colleagues for valuable discussions relating to the subject matter. The responsibility for the opinions expressed, is, however, entirely my own.

REFERENCES

1. MacKAY, D. M. Towards an information-flow model of human behaviour. *Brit. J. Psychol.*, 1956, **47**, 30.
2. SLUCKIN, W. Minds and Machines. *Penguin, London* (1954).
3. THORPE, W. H. The concepts of learning and their relation to those of instinct. *Symposium of the Society for Experimental Biology*, 1950, no. 4, p.387.
   also Learning and Instinct in Animals. *Methuen, London,* (1956).
4. WIENER, N. Cybernetics, p.150 *Wiley, New York,* (1948).
5. SELFRIDGE, O. G. Pattern recognition and learning. Paper no.33 in Information Theory, ed. E. C. Cherry, *Butterworth, London,* (1956).
6. ASHBY, W. R. Design for a Brain. *Chapman and Hall, London,* (1952).
7. UTTLEY, A. M. The conditional probability of signals in the nervous system. *R.R.E. Memo, no. 1109,* 1955. Radar Research Establishment, Malvern, Worcs.
8. UTTLEY, A. M. A theory of the mechanism of learning based on the computation of conditional probabilities. *Proc. of the 1st International Congress of Cybernetics, Namur,* (1956).
9. UTTLEY, A. M. The design of conditional probability computers. *Information and Control.* In the press.

10. RUSSELL, G. Interim report on learning machines and adaptive control systems. *R.R.E. Memo. no. 1369.* 1957.
11. UTTLEY, A. M. The classification of signals in the nervous system. *E.E.G. Clin. Neurophysiol.,* 1954, **6**, 479.
12. MacKAY, D. M. The epistemological problem for automata. Automata Studies, p.235 ed. by C. E. Shannon and J. McCarthy, *Princeton,* (1955).
13. MacKAY, D. M. Mindlike behaviour in artefacts. *Brit. J. Philosophy of Science,* 1951, **2**, 105.
14. PITTMAN, G. F. A high-accuracy static time delay device using transistors. *A.I.E.E. Trans. Pt.1,* March 1955, p. 54.
see also Schaeffer, D. H. Magnetic core event counter for earth satellite memory. *Elect. Eng.,* 1958, **77**, 52.
15. LICKLIDER, J. C. R. A duplex theory of pitch perception. *Experientia,* 1951, **7**, 128.
see also SAYERS, B. M. and CHERRY, E. C. Mechanism of binaural fusion in the hearing of speech. *J.Acous. Soc. Amer.,* 1957, **29**, 973.
16. BOX, G. E. P. Evolutionary operation: a method for increasing industrial productivity. *Applied Statistics,* 1957, **4**, 81.
17. DAVIES, O. L., ed. Design and Analysis of Industrial Experiments. *Oliver and Boyd, Edinburgh,* (1956).
18. BOX, G. E. P. and WILSON, K. B. On the experimental attainment of optimum conditions. *J. Roy. Statistical Soc.,* 1951, **13**, series B, p.1.
19. BOX, G. E. P. The exploration and exploitation of response surfaces: some general considerations and examples. *Biometrics,* 1954, **10**, 16.
20. BOX, G. E. P. and YOULE, P. V. The exploration and exploitation of response surfaces: an example of the link between the fitted surface and the basic mechanism of the system. *Biometrics,* 1955, **11**, 287.
21. BOX, G. E. P. and COUTIE, G. A. Application of digital computers in the exploration of functional relationships. *Proc. I.E.E.* 1956, **103**, pt. B, supp. 1.
22. FISHER, R. A. The Design of Experiments. *Oliver and Boyd, Edinburgh,* 3rd. Edition, (1942) p.88.
23. KALMAN, R. E. Design of self-optimising control system. *Trans. A.S.M.E.,* (Feb. 1958,) p.468.
24. WILBY, W. P. L. and WOODCOCK, R. Self optimising filter. *Imperial College, London.* First progress report, July, 1957, second April, 1958.
25. CHAPMAN, B. L. M. and FREED, G. A. An electronic analogue of nervous activity. Unpublished paper.
26. WEATHERBURN, C. E. A First Course in Mathematical Statistics. p.247, *Cambridge University Press,* (1952).
27. OETTINGER, A. G. Programming a digital computer to learn. *Phil. Mag.,* 1952, **43**, 1243.
28. DEUTSCH, J. A. A new type of behaviour theory. *Brit. J. Psychol.,* 1953, **44**, 304.
29. FRIEDBERG, R. M. A learning machine. *I.B.M. Journal,* 1958, **2**, 2.

APPENDIX TO THE PAPER BY DR. A. M. ANDREW∅

One of the main objects of the paper is to point out the need for some form of interpolation in any learning machine applied to process control. Two forms of interpolation are discussed; one under the heading "Interpolation" in section 4.3, and one in terms of polynomial functions in part 5. Without some form of interpolation the learning machine cannot exploit the continuity of its environment.

The principles discussed in the latter part of the paper lead to a number of interesting possibilities not mentioned in the paper. One of these is the possibility of making machines which will "learn to learn". The design of a learning machine involves the choice of values for a number of parameters, including time-constants of smoothing for the running values of statistical quantities, and threshold levels at which these quantities produce a change in the control policy of the machine. A person designing the machine would need to choose these values rather arbitrarily, since there is no simple way of determining the values which will produce most rapid learning. In fact, the design of a learning machine is precisely the kind of intractible mathematical problem which it is hoped to by-pass by the use of learning machines. Eventually, therefore, it is likely that machines will be devised in which the learning process itself will be subject to adaptive modifications.

Two of the references require further comment. In referring to Wilby and Woodcock *(ref. 24)*, the writer should have realised that these workers are under the direction of Professor Gabor, and are putting into practical form the self-optimising filter proposed by Prof. Gabor *(ref. 1)*.

A much abbreviated form of the paper by Chapman and Freed *(ref. 25)* has now been published *(ref. 2)*.

REFERENCES

1. GABOR, D. Communication Theory and Cybernetics. *Trans. I.R.E.*, 1954, **CT-1**, No. 4.
2. CHAPMAN, B. L. M. and FREED, G. A. Random channel selectors. *Automation Progress*, 1958, **3**, 430.

∅ Added after the Symposium.

DISCUSSION ON THE PAPER BY DR. A. M. ANDREW

MR. C. STRACHEY: I find myself in some difficulty at the prospect of having to cram into five minutes all the comments I want to make on Dr. Andrew's paper. First of all, I think it is fair to say that there are two or perhaps three types of learning programme (or machine) which can be envisaged. There is the sort of self-organising type, such as the perceptron, which you hope is ultimately going to organise itself into a suitable configuration. Then there is the type which might be called the 'thinking' machine, which is sophisticated to start with and has some of its concepts already built in, as with the geometrical problem-solving machine. I find both these very interesting indeed, but neither of them at the moment show signs of having any practical application. There is a third type of learning machine, which is really not so much a learning machine as an optimum seeking machine, which is specialised, in that you decide beforehand what sort of thing you want it to learn in considerable detail. Machines of this sort have probably less claim to be called learning machines, but more claim to be called useful. They have been simulated on digital machines for a number of years, and are generally regarded as being uninteresting from the point of view of programming.

I think Dr. Andrew has confused the issue considerably by trying to apply to a practical problem a machine of the self-organising perceptron type. I do not think this is an appropriate thing to do. The first two-thirds of the paper refers to logical type conditional probability computers, and finally comes to the conclusion, with which I shall agree, that they are not much good for the type of thing he is proposing for them; the tables of the size of storage required in any practical problem certainly makes it difficult to imagine constructing a machine of this sort.

I have some further evidence which I should like to put before you to reinforce this point of view; it is not concerned with the amount of equipment but with the way in which it functions. The last third of the paper is concerned with a totally different proposal, which I think might probably stand more chance of success. But it seems to me there is nothing in this part of the paper which is other than purely speculative. It seems to me a great pity that, when you are talking about learning machines, you should start to describe them as applied to complicated physical problems such as controlling a chemical works before you have made some practical investigation, or simulated investigation, on digital computers which you can make so very easily for these systems. I think in any practical control problem you have to make use of all the specialised information you can get hold of, though I'm afraid I have not time to expand that at the moment.

I just want to say that, after reading this paper, I was stimulated - perhaps I should say pricked - into writing a programme for a digital machine which simulates on a slightly larger scale, the machine described in figure 7. This is supposed to be a learning machine of the same type as the conditional probability computer. The machine I envisaged has two decimal digit inputs and two decimal digit outputs: you can set it to add, subtract or multiply or to give the least significant digit of the sum or the product. What you ask it to do depends on the indication of success which is given by a separate part of the routine, which is called the teacher. The way it worked was such that if it got a bad answer the probability of that one was depressed, and if it got a right answer it went on getting it right. Now the remarkable thing about this particular routine is that it behaves considerably worse than you would expect; it behaves worse than at random. At first it was only looking for the least significant digit in the sum, and before it got all the answers correct there were 649 wrong ones. On the average it had tried six and a half times for each successful answer, which is worse than if you went straight through, say, from 1 to 9. This is not unreasonable because obviously sometimes you try the same wrong numbers by chance several times.

DR. L. M. SPETNER: In this paper Dr. Andrew has indicated that it seemed quite hopeless to use the idea of logical design or digital techniques in his machine because of the large number of boxes that would be required to handle each one of his functions. The situation might, however, be little more optimistic than this, in that one might take advantage of a self-organizing type of device, such as the perceptron, in doing this job. Here the self-organization might consist, not of reinforcing the values of the association cells, but perhaps adjusting the thresholds. Adjustment of the thresholds seems to be a reasonable thing to do when one is trying to effect a logical network. The reason for this is that one may consider a threshold device with many binary inputs as a generalized logical network. For example, a threshold device with two inputs that can have the value of 0 or 1 and has a threshold of 1 is an either-or device, and so on. So that if one has N inputs into such a unit then there are N different thresholds that may be used. This allows the device to be any one of the N different symetrical logical functions. There is of course a sticky problem as to how one tells whether he should raise or lower any particular threshold to effect some given response.

This might be done by nutating the threshold during the computation of the responses, and from this nutation determine which way to change the threshold in order to improve the response.

I hope that these comments will stimulate further thought along these lines.

DR. A. M. ANDREW (in reply): I disagree with Mr. Strachey in his assumption that the third type of learning machine he mentions - the optimum-seeking machine - must necessarily be such that the sort of thing it is to learn is predetermined in considerable detail. Section 7.2 of the paper is intended to indicate how the principles of these machines can be extended to incorporate self-reorganisation. The use of sub-goals which are subject to adaptive modification, as mentioned in Section 4.2, introduces still further flexibility of operation.

I apologise for not adequately reviewing previous work in self-optimising systems. A large amount has been published, including a review of the field by Aseltine, Mancini and Sarture (*ref. 1*), and details of the Westinghouse OPCON by Burt and Van Nice (*ref. 2*). The work of Draper and Li (*ref. 3*). is particularly noteworthy.

It is difficult to comment on Mr. Strachey's experiment in programming without having further details. In any case, the problem to which he has applied the computer is rather different from the process-control type of application which I have been considering.

I agree that computer simulation of learning machines can be extremely valuable. One of my colleagues is already using a digital computer in this way, and I am starting to prepare a programme simulating a learning machine which applies the control function shown as Equation 6 of the paper. The construction of such a programme becomes a rather more complicated matter than Mr. Strachey suggests if it is designed to make a worthwhile contribution to the derivation of design principles for learning machines.

The kind of logical network suggested by Dr. Spetner does not, so far as I can see, allow interpolation any more readily than do other logical-type devices. Without this feature the machine must take a long time to learn to perform any task involving continuous variables, even if its complexity can be kept within limits of feasibility.

REFERENCES

1. ASELTINE, J. A., MANCINI, A. R., and SARTURE, C. W. A Survey of Adaptive Control Systems. *I.R.E. Trans,* 1958, vol. PGAC-6, 102.
2. BURT, D. A. and VAN NICE, R. I. Optimizing Control Systems for the Process Industries. *Technical Publication, Westinghouse Electric Corporation,* Feb. 1959.
3. DRAPER, C. S. and LI, Y. T. Principles of Optimalizing Control Systems and an Application to the Internal Combustion Engine. *A.S.M.E. Publication,* Sept. 1951.