SELF-ORGANISING LEARNING CONTROL AND ITS APPLICATIONS TO MUSCLE RELAXANT ANAESTHESIA

D A Linkens and S B Hasnain

Department of Control Engineering
University of Sheffield
Mappin Street
Sheffield S1 3JD

RESEARCH REPORT NO 383   January 1990

SELF-ORGANISING LEARNING CONTROL AND ITS APPLICATION TO MUSCLE RELAXANT ANAESTHESIA

Linkens, D.A & Hasnain, S.B

Department of Control Engineering, University Of Sheffield

1 INTRODUCTION

Controlling systems with interacting response variables and interacting actuators has been a formidable task requiring extensive modelling to identify measure plant characteristics. These attempts often result in less then an ideal controller. With self-organising techniques the uncertainty of plant behaviour need not be an obstacle to fast, efficient and stable control. In fact they do not require a definitive course of action based on a given set of conditions. Self-organising control is characterised by the fact that autonomous on-line modification of the control law is accomplished by the use of logic which is independent of a priori knowledge of the functional relationship between the actuator and the plant inputs and their outputs. Inherent in this characteristic is a freedom from a need for explicit mathematical representation of control loop elements and their interaction as a system with the environment. Self-organising control can ultimately lead to increased system reliability, since a large class of controller components failures and actuators malfunctions can be placed in the category of a changing plant against which adaptation is to be achieved.

Forms of self-adaptive or self-tuning control have been studied extensively for industrial processes over recent years (examples texts being Billings & Harris, 1981, Astrom & Wittenmark, 1984). In these cases a mathematical model of the process is required, such that parameter estimation may be used to determine the necessary controlled parameters in an on-line manner. Relatively little work has been done on control systems which do not require a detailed model structure, and in this paper we describe an approach called the self-organising controller. The first practical application of the Self-Organising Controller(SOC) was reported by Barron [1965], who developed a laboratory prototype called Probability State Variable controller (PSV). Barron's work grew out of the work by

R.J.Lee [1959] and Ostgard [1962]. With the interest of U.S Department of Defence, successful flight testing of an elementary PSV controller was performed [Barron 1965]. Later several aerospace applications of PSV controller were also developed [Barron 1966; Barron 1969(a); U.S, State Dept 1970].

In the past, attention has focused on aerospace and aircraft applications of SOC. Here we have applied SOC to a medical application where it is used to control muscle relaxant anaesthesia in operating theatres. The model for the anaesthetic process has been developed and verified by earlier researchers, a brief review is provided in section 2. The learning aspect of self-organisation is discussed in section 3. The functional elements of the learning system and the concept of Probability State Variable is described in section 4. Section 5 covers the Performance Assessment logic module and its mechanisation. The Probability State Variable logic module is described in section 6. The modified version of SOC called Advanced Self-Organising Controller for multiple actuator control and its inherent parallelism is presented in section 7. Results of the simulations performed with SOC on sequential and parallel machines applied to the anaesthetic model are given in section 8.

2. MUSCLE RELAXANT PROCESS MODEL

Muscle relaxant drugs are used by anaesthetists who, based on their own experience, determine the adequate dose in order to obtain a predefined degree of paralysis. However, anaesthetists sometimes fail to maintain a steady level of relaxation resulting often in fluctuating and possibly large consumption of drugs by the patients. A safe and fast method for designing closed-loop infusion systems is to replace the patient with a mathematical model.

The pharmacology of muscle relaxant drugs comprises two important parts, pharmacokinetics and pharmacodynamics. Pharmacokinetics concerns the absorption, distribution and excretion of drugs, whereas pharmacodynamics is a term employed in drug pharmacology to describe the relationship between drug concentration and the therapeutic effects of these drugs.

2.1 PHARMACOKINETICS

For the drug Pancuronium the body is considered to consist of two compartments between which reversible transfer of drugs can occur. The drug is injected into compartment 1 which represents the plasma volume and the perfused tissues such as heart, lung, lever etc., from which it is excreted into urine. The drug is exchanged between the 1st compartment and the connecting peripheral compartment. The plasma concentration of drug versus time is given by the bi-exponential equation.

$$c(t) = A1e^{-\alpha 1t} + A2e^{-\alpha 2t} \tag{1}$$

where;

A1 and A2 are complex functions

$A1e^{-\alpha 1t}$ reflects the distribution phase

$A2e^{-\alpha 2t}$ reflects the elimination phase.

2.2 PHARMACODYNAMICS

It is known that the interaction between the injected drug and the components of the body induce a certain therapeutic effect. The steady state plasma concentration of the drug is proportional to the amount of drug at the site of action. The plasma concentration of drug versus effect relationship can be described mathematically by a Hill equation[Whiting and Kelman 1980] of the form;

$$E = \frac{E_{\max} \cdot C^{\alpha}}{C^{\alpha} + C^{\alpha}_{50}} \tag{2}$$

where;

E represents the drug effect

$E_{\max}$ the maximum drug effect

C the drug concentration

C_{50} The drug concentration corresponding to a 50% effect.

Using previous pharmacological identification studies[Linkens et. al. 1982(a)], the drug Pancuronium can be modelled as a two-time-constant linear dynamic system with a dead-time in series with a non-linear part comprising a dead space and a limiting device or a Hill equation as illustrated in Fig 1.

3. SELF-ORGANISING AND LEARNING

Although the ultimate objective in the development of Self-organising techniques may be to duplicate the full range of human intelligence capabilities, current emphasis and practical attempts are restricted to learning techniques. Even in this limited aspect of artificial intelligence a further division is necessary. Thus, a distinction must be made between the process of learning and the long-term retention and reuse of results from the past learning activities.

Learning is synonymous with self-organisation, i.e, the learning goal is taken to have been achieved when the elements of the learning system have organised themselves to produce an output confirming to the specified goal. A self-organising control system may contain one or more learning systems, and these may be arranged to serve a variety of purposes in performing the control function. Thus, a self-organising control system can be configured so as to learn the control strategy, or it can employ an explicit control law, using learning to alter the values of the parameters within this law.

Self-organising control systems can be viewed as a sub-class of self-adaptive control systems; however the latter term, by common usage, has become associated with particular types of adaptive systems, i.e, to various model reference systems, high-gain techniques and error switching techniques evolved in the course of time. Nevertheless the purpose of self-organisation is to provide better adaptation, the distinguishing characteristic of self-organising control system being in the manner in which the adaptation is accomplished. Thus a self-organising adaptive control system can be defined as one which utilises learning to carry out adaptation processes.

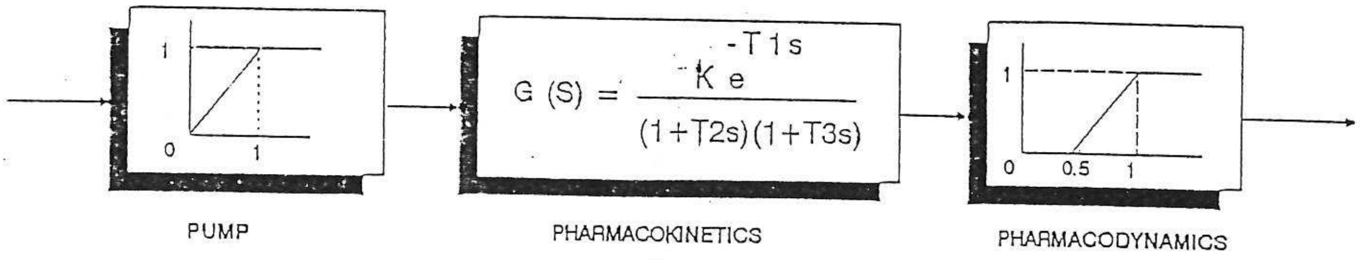

FIG 1: PANCURONIUM BROMIDE DRUG MUSCLE RELAXANT MODEL

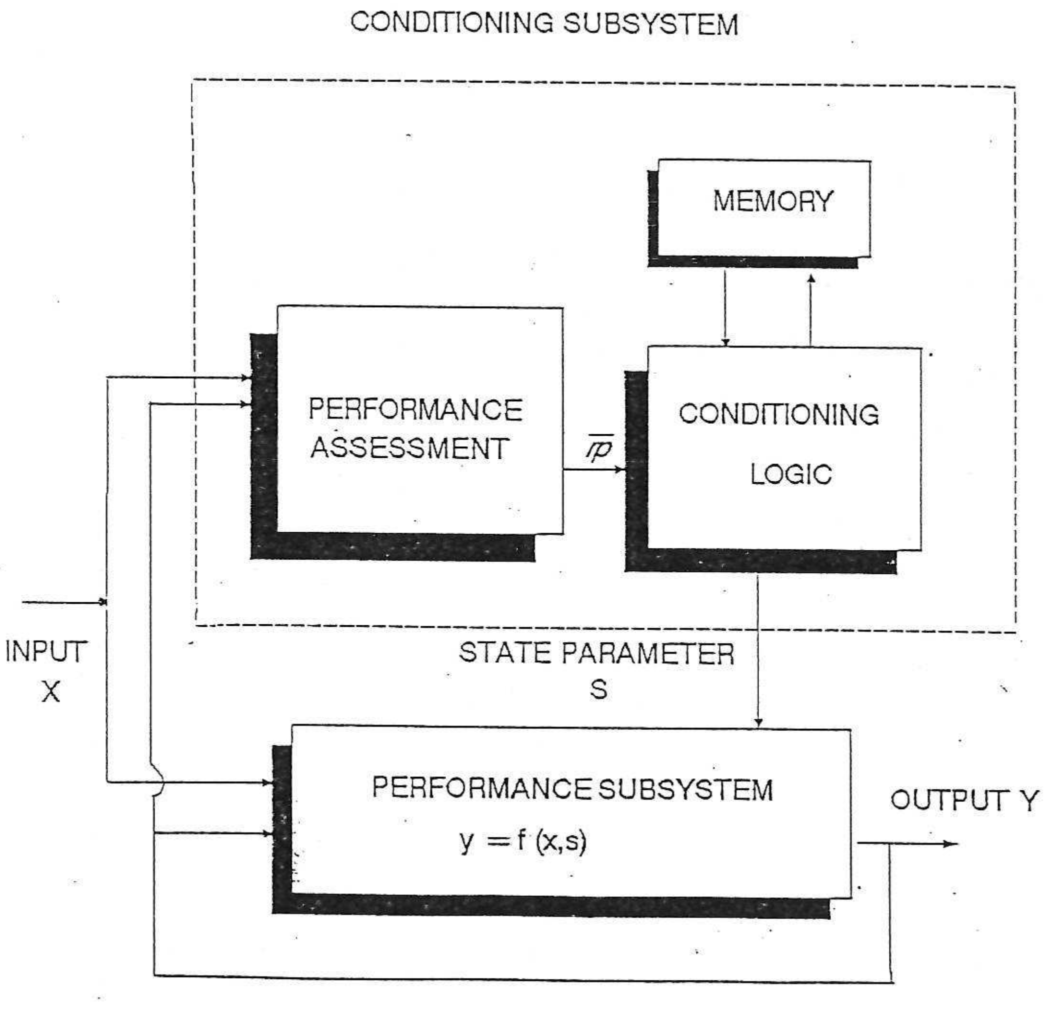

FIG 2: FUNCTIONAL DIAGRAM OF LEARNING SYSTEM

4. FUNCTIONAL ELEMENTS OF LEARNING SYSTEMS

A learning system performs continuous or periodic assessments of its own behaviour, and by relating these assessments to its past experimentation(experience), alters(adapts) its characteristics so as to increase the likelihood that its future behaviour will be brought into or remain in accordance with its criterion of desirable behaviour. A learning system therefore contains the following functional entities in addition to those elements usually found in conventional systems:

1. PERFORMANCE ASSESSMENT LOGIC: A means for assessment of the system behaviour in terms of criteria of desirable behaviour. In general the performance assessment requires processing of patterns present in the controller input/output information and comparison of these patterns with an appropriate standard. However, the comparison need not be explicit.
2. MEMORY: A means for storing information concerning past experiments. The memory can take the form of explicit arithmetic data or of implicit probability distributions.
3. CONDITIONING LOGIC: A means for computing and effecting experiments, i.e, alterations of the system characteristics. The conditioning logic implements the strategy of learning.

These functional entities taken in combination, constitute what may be called the conditioning subsystem [Barron 1968(b)]. The conditioning subsystem generates changes to parameters within the performance subsystem. Fig 2 illustrates the relationships between the conditioning subsystem and performance subsystem in a learning system. We can write two equations which express the key properties of learning systems:

$$y = f(x,s) \tag{3}$$

$$s_{k+1} = g(x_k, s_k) \tag{4}$$

where

x = system input vector

y= system output vector

s = system internal state vector

subscript "k" refers to the kth instant of time

subscript "k+1" refers to the (k + 1)st instant of time

A mechanisation of equation 3 is termed a performance subsystem as shown in Fig 2. A Conditioning subsystem is the mechanisation of equation 4 and is often viewed as the form of generalised network of trainable devices. A conditioning subsystem will generally require measures of x and y to generate changes in the performance subsystem state. However to qualify as a learning system process, the conditioning must not require a priori knowledge of either the form or numeric characteristics of $f(x,s)$. Furthermore this independence of a priori information must apply both to the operation and the design of the performance assessment and conditioning logics.

In the SOC applications, the Performance Assessment (PA) logic computes three discrete performance values (denoted by $\overline{rp}$ in Fig 2), +1 for forward (or positive reinforcement), -1 for punishment (or negative reinforcement), and 0 for zero reinforcement. The conditioning logic mechanises the strategy of learning. Strategies of learning usually embody a form of parameter search, often linked to hill climbing. However to be of practical utility in learning applications for dynamic processes the hill climbing must be performed within a time varying environment. The basic requirements are,

(i). High speed of search.

(ii). Provision for probable degradation of information by noise.

(iii). Simplicity of mechanisation despite multiple parameters.

Since conventional hill-climbing techniques (gradient methods of steepest descent) do not satisfactorily meet all the above requirements, new strategies have been evolved. One of these strategies relates to a small number of parameters(in the order of ten) in a rapidly time-varying environment, while the other strategy is intended for a large number of parameters(in the order of several hundred) in a relatively static environment. The latter strategy, termed Random State Variable strategy (RSV) has proved very useful in parameter identification processes. The former strategy Probability State Variable (PSV) is here a part of the conditioning logic and fundamental to SOC systems.

Generally speaking, the PSV is found to provide convergence times sufficiently small to permit the compensation of dynamic environments and time-varying plants without major restrictions. This is partly because, with prediction, the periodic performance assessment can be made over intervals of time which are a small fraction of the transient response time. To make full advantage of high learning speed, the performance criterion should therefore be designed to shape the transient response in an optimum manner.

5. PERFORMANCE ASSESSMENT LOGIC MODULE

The goal of the Self-Organising Controller (SOC) unit can be expressed mathematically in terms of the minimisation of an integral performance index of the general form.

$$I = \sum_{k=0}^{k=kf} \int_{t_k}^{t_k + \Delta t} F \, dt \tag{5}$$

where $F \geq 0$; the discrete times, t_k, are clocking points at which the controller state changes may occur; and Δt is the interval between two successive clock occurrences.

Assuming the controlled plant to be continuous in the time domain and describable in terms of an Nth order characteristic equation, equation 5 can be approximated as follows,

$$I = \sum_{k=0}^{k=kf} \left[F_{t_k} \Delta t + \left[\frac{d}{dt} F \right]_{t_k} \frac{\Delta t^2}{2} + \ldots\ldots + \left[\frac{d^N}{dt^N} F \right]_{t_k} \frac{\Delta t^{N+1}}{(N+1)^!} \right] \tag{6}$$

At each point t_k, F and all its derivatives below $(\frac{d^N}{dt^N}F)$ are predetermined (they are system state variables). $(\frac{d^N}{dt^N}F)$ is the lowest-ordered derivative which may be altered by the SOC acting at a point.

It follows that the goal (min I) produces a necessary condition,

$$\left[\frac{d^N}{dt^N}F\right]_{t_k} < 0 \quad (k=1,2,.....,kf) \tag{7}$$

which system actuator elements can always produce in constraint free cases. Furthermore, given sets of alternative values for $(\frac{d^N}{dt^N}F)$ at each point t_k, the time optimal-control policy is that which always chooses the most negative value of this derivative. In practice it is sometimes difficult to realise this latter (sufficient) condition because its use requires a priori knowledge of plant polarity.

For the special case of N=2, equation 7 leads to a requirement on $(\frac{d^N}{dt^N}F)$. A form of the integrand F is F = $| e_p |$, where e_p is the predicted error function. The differential order of e_p is of importance in determining the shape of transient responses. In general, to define this shape completely, one would have to deal with

$$e_p = e_{t_k} + \left[\frac{d}{dt}e\right]_{t_k} T + ... + \left[\frac{d^{N-1}}{dt^{N-1}}e\right]_{t_k} \frac{T^{N-1}}{(N-1)!} \tag{8}$$

where T is a positive constant. Usually, $T \gg \Delta t$, because this produces a well damped response of the system during transients.

For a system to have a stable equilibrium state, some scalar function, V, of its state variables must be positive-definite while $\dot{V}$ must be negative-definite.

The trivial solution, $V = 0$, is then asymptotically stable in the sense that the state vector will converge toward the trivial solution as $t \rightarrow \infty$. For example, if N = 2 the state variables are e and $\dot{e}$, and $V = f(e, \dot{e})$. The trivial solution, and hence the equilibrium state for which asymptotic stability is sought, then corresponds to e and $\dot{e}$ becoming zero simultaneously. A geometric interpretation of the above would consist of a plot of the surface defined by $V = f(e, \dot{e})$. Because of the conditions imposed on V and $\dot{V}$, this surface is in the form of a cup or cone which is always above the $e - \dot{e}$ plane, touching it only at the origin which, in this case, represents the desired equilibrium state $e = \dot{e} = 0$.

In the synthesis of a self-organising control system for the case N = 2, one approach might be to select V as a positive-definite function of e and $\dot{e}$ and mechanise $\dot{V}$ in the performance assessment unit. If the value function were generated on the basis of sign $\dot{V}$ (Lyapunov stability theory), one might expect that the system would exhibit stable behaviour. In practice, however, it is found that the use of $\dot{V}$ does not go far enough (except for the case in which N =1), because $\dot{V}$ says nothing about the desired quality of control, i.e, no requirements are imposed on the manner in which the equilibrium state is to be approached. An improvement is obtained by basing the performance assessment on $\ddot{V}$ (for N = 2), requiring that $\ddot{V}$ be negative. The value function is then positive for $\ddot{V} < 0$ and negative for $\ddot{V} > 0$. This last condition achieves two objectives: First, if $\dot{V}$ is positive, the learning process is directed to make $\dot{V}$ negative so as to assure stable convergence. Second, because V may be viewed as a measure of the "distance" from the desired state, the requirement $\ddot{V} < 0$ leads to optimising the rate with which this distance is reduced, i.e, for N = 2 the system strives to accelerate convergence toward the equilibrium state.

To achieve accelerated motion to the phase plane origin, the function V for N = 2 may be chosen to be

$$V = | e_p | = | e + T\dot{e} | \tag{9}$$

equation 9 does not comply with the general requirements for a Lyapunov function, since it is not positive-definite, i.e, $e + T\dot{e} = 0$ is a line in the phase plane passing through the origin. This, however, is often a desirable form, since anywhere on the line the solution of $e + T\dot{e} = 0$ yields,

$$e = e(0)\ exp(-t/T) \tag{10}$$

where $e(0)$ is the initial error and t denotes time. Thus, once on the line given by $e + T\dot{e} = 0$, system error convergence tends to take place exponentially and T may be viewed as nominal time constant for the terminal region of the response.

In summary the PA criterion may be based on the second derivative of V as defined by equation 9 and since only sign information is sought in simpler self-organising processes, this yields,

$$\overline{rp} = sign\ \ddot{V} = sign\ e_p\ sign\ \ddot{e}_p \tag{11}$$

where

$$e_p = e + T\dot{e} \tag{12}$$

$$V = |\ V_P\ | \tag{13}$$

The criterion of equation 9 will call for accelerated convergence to the $e = -T\dot{e}$ line and constrained(exponential) convergence along this line to the origin. This is illustrated in Fig 3, where the surface defined by equation 9 now takes the form of a trough touching the phase plane along the $e = -T\dot{e}$ line. Equation 7 can now be employed to derive two types of performance Assessment (PA) criteria for the case N = 2.

5.1 PERFORMANCE ASSESSMENT CRITERION

For the case N = 2 and taking F = $|\ e_p\ |$, equation 7 requires that

$$\left[\frac{d^2}{dt^2}\ |\ e_p\ | < 0\right] \tag{14}$$

where from equation 8

$$e_p = e + T\dot{e} \tag{15}$$

Introducing a binary reinforcement (reward or punish)quantity denoted by $\overline{rp}$ equation 14 yields,

$$\overline{rp} = -sgn \frac{d^2}{dt^2} \mid e_p \mid \tag{16}$$

where $\overline{rp}$ = +1 is defined as reward and $\overline{rp}$ = -1 as punish. Equation 16 is the basic form of the performance assessment criterion.

5.2 MECHANISATION OF PERFORMANCE ASSESSMENT CRITERION

The essential purpose of performance assessment is to perform a continuous assessment of self-organising control system performance as a function of system error signal on channel 1 and to generate V signal on channel 2 (Fig 4). The prediction error signal e_p on channel 3 is calculated by a predictor function, and may be expressed as $e_p = e(t) + T\dot{e}$, where $e(t)$ is the instantaneous system error signal on channel 1 and T is the prediction interval(constant). Since the overall goal of the system is to reduce $e(t)$ to zero as rapidly as possible without overshoot(which would result in the oscillatory convergence of the error signal to zero), it is desirable to generate a V signal on channel 2 which produces maximum acceleration until the predicted error changes sign and then an exponential convergence to zero error. A signal of the form,

$$sgn\ V = -\ sgn\ e_p\ .\ sgn\ \ddot{e}_p \tag{17}$$

produces this result .

The Performance Assessment stage output may be expressed as the Boolean function.

$$V = reward = sgn\ e_p\ .\ \overline{sgn\ \ddot{e}_p} + \overline{sgn\ e_p}\ .\ sgn\ \ddot{e}_p \tag{18}$$

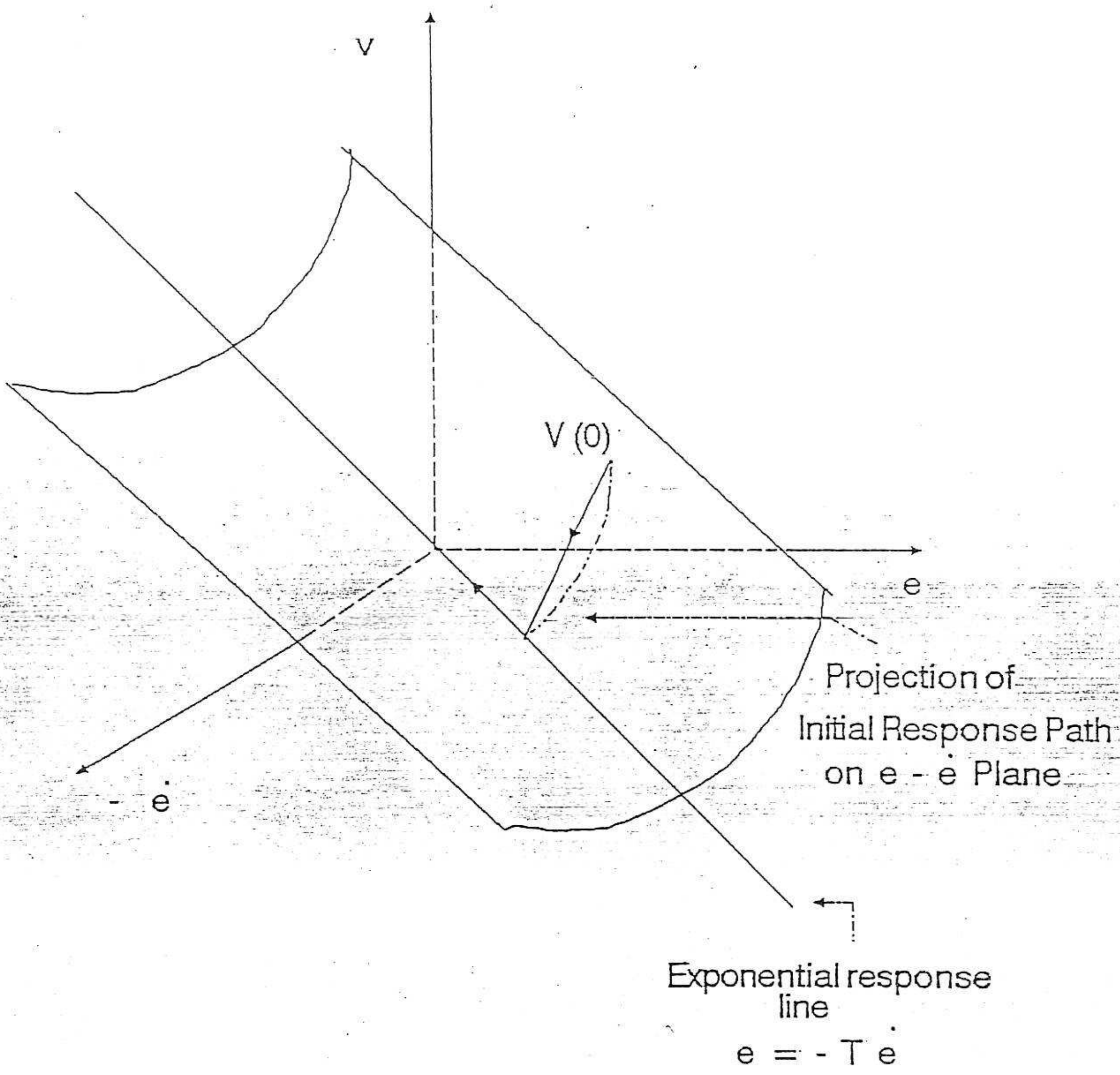

FIG 3: STATE SPACE REPRESENTATION OF SOC

when the output is logical one, and

$$V = punish = sgn\ e_p\ .\ sgn\ \ddot{e}_p + \overline{sgn\ e_p}\ .\ \overline{sgn\ \ddot{e}_p} \tag{19}$$

when the output is logical zero.

6. PROBABILITY STATE VARIABLE MODULE

The experimental process in a Self-organising system must exhibit rapid convergence to appropriate time-varying parameter and/or signal levels. To achieve this rapid convergence, it is desirable to use extremely fast PA sampling and conditioning logic experiment generation. The general functions performed by the Probability State Variable (PSV) conditioning logic section of the SOC are,

1. To make experimental changes in its own output state
2. To retain a record, or memory of recent output state changes
3. To associate the contents of its memory with a performance assessment signal, $\overline{rp}$ (for reward punishment).

Consideration of the functions listed above leads to a sub-division of the PSV logic into three functional sections. The first of these sections is a source of experimentation having controllable statistics, the second is a means for storing the results of previous experiments and the third is the means for logically combining the memory contents with the $\overline{rp}$ signal so as to obtain a valid correlation between cause and effect.

With an incremental PSV strategy, the output state variable here denoted $k(t)$, is periodically incremented one step either up or down from its prior existing level. The step size, Δk is predetermined, and the time interval between state changes termed Δt is fixed. The direction of increments of $k(t)$ is probabilistic, and is determined by the instantaneous state of a statistical source, which generates a random binary sequence having a duty cycle, or probability of being in one state, which is continuously controllable. The control of the statistics of the

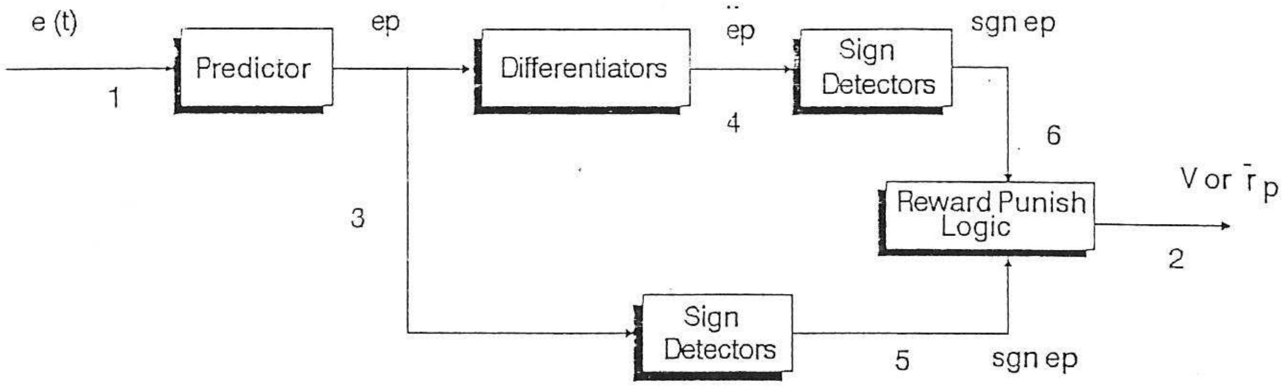

FIG 4: PERFORMANCE ASSESSMENT UNIT

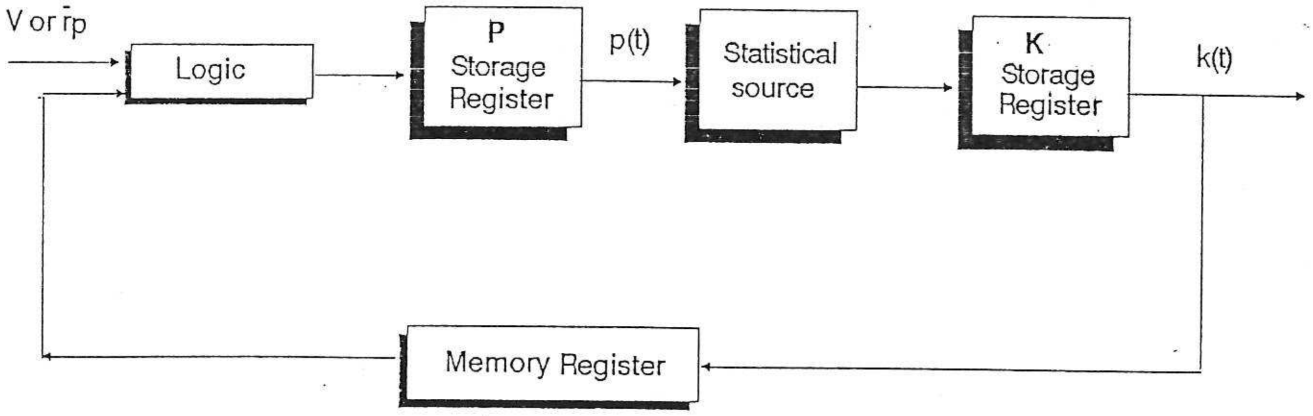

FIG 5: BASIC DIAGRAM PSV CONDITIONAL LOGIC

source is exerted by a variable, termed $p(t)$, which is integral to the PSV logic. This variable is also incremental and like $k(t)$, can be incremented one step per Δt, although unlike $k(t)$, $p(t)$ need not be incremented every Δt. The direction of incrementation of $p(t)$ is not probabilistic however, but is determined by the memory state (sign Δk) and the instantaneous value of $\overline{rp}$. Thus the combination of cause (as reflected in the memory state) and effect, ($\overline{rp}$) acts, by incrementing $p(t)$, to change the probability associated with a particular direction of change of $k(t)$, and in particular, acts to increase the probability that $k(t)$ will change in a direction that will produce a positive ($\overline{rp}$). A diagram showing the interrelationship of these basic functions is shown in Fig 5.

The variable $p(t)$ is permitted to assume a fixed number of discrete values, and which may, but need not, step one increment up or down at time intervals of Δt. Associated with each value of $p(t)$ is a specific probability that the statistical source output random binary sequence will be zero or one. A value of $p(t)$ at the midpoint of its range corresponds to a probability of 50% that the source output will be a one or zero. An excursion of $p(t)$ away from its midpoint will increase the probability of a one, while an excursion in the other direction will increase the probability of a zero.

The remaining functions within the PSV conditioning logic are the memory and the correlation network. The latter consists of gating, by means of which the memory contents are associated with $\overline{rp}$ information received from the PA unit. The approach taken for this part of the synthesis was to proceed on the basis that the incremental performance assessment has validity, i.e, the individual effect of a recent Δk whether positive or negative has been correctly discerned by the PA in its determination of $\overline{rp}$. It is necessary therefore only to store information as to the direction of each $k(t)$ increment, and then to associate this memory information with the resulting $\overline{rp}$ so as to determine the desirability of increasing or decreasing the probability of continuing in the particular Δk direction.

6.1 STATISTICAL SOURCE

The statistical source is a signal generator producing an output which is a random sequence of zeros and ones. The unique feature of this statistical source is that its duty cycle, or the probability of its being in one state or the other, is continuously controllable by means of an analog voltage, $p(t)$. When this probability control voltage is at its nominal centre, the output status has a 50% probability of being in the either state. Changing the probability control voltage to its extreme positive or negative value biases the probability to about 90% in favour of one or zero, depending upon the polarity of the probability control voltage.

7. ADVANCED SELF-ORGANISING CONTROLLER

Based on the SOC theory presented earlier, Barron proposed a modified SOC [Barron 1967(a); Barron 1968(a)]. The conditioning logic unit was replaced by a more effective Activation Logic Unit(ALU). The block diagram of a single input multiple output SOC with Activation logic is presented in Fig 6. The Performance Assessment unit is similar to the one described earlier. The PSV logic module is now a subsystem of the ALU.

7.1 ACTIVATION LOGIC UNIT

The activation logic subsystem consists of a non-linear amplifier, the correlation logic unit (PSV) and an output multiplier. Feld'Baum [1965] states " control actions must bear a dual character. They must be investigators to a known degree, but also directors to a known degree". Feld'Baum refers to control in which such a dual character is present as "dual control". Examples of dual control are automatic search systems, and in particular automatic optimising systems. In usual systems of this type it is often easy to separate the investigating or testing part of the action from the directing part of the action either on account of the difference in their frequency ranges or because they alternate in time. But in the general case such division is not compulsory, the same action can have a dual character and be partially perceptive and partially directive.

In dual control systems a contradiction arises between the two aspects of the control action mentioned above. In fact successful control is only possible

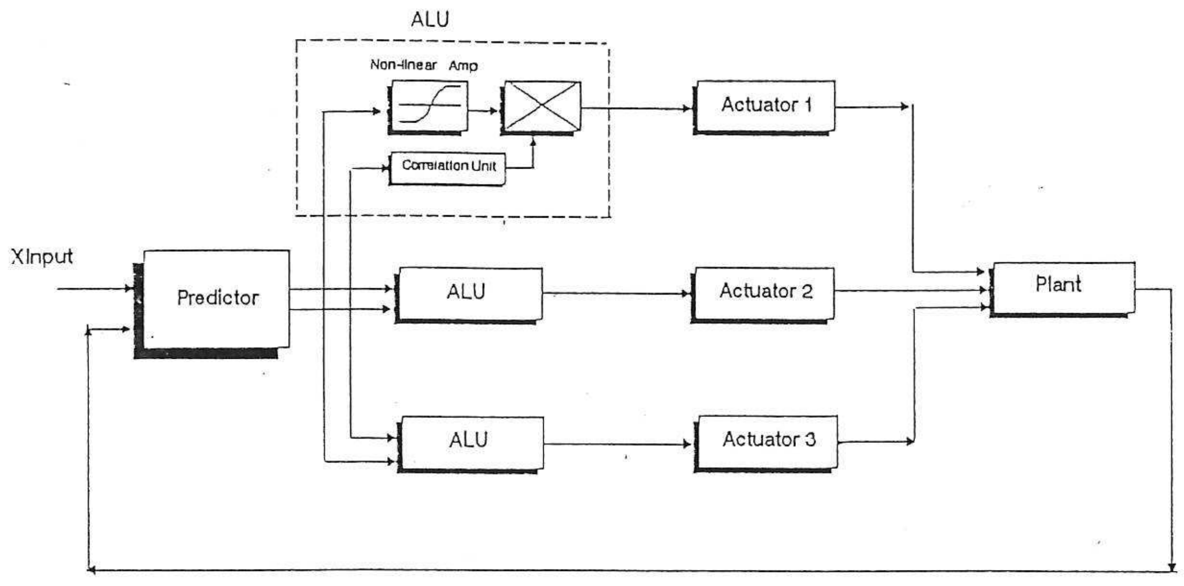

FIG 6: SOC FOR SINGLE INPUT 3 OUTPUT SYSTEM

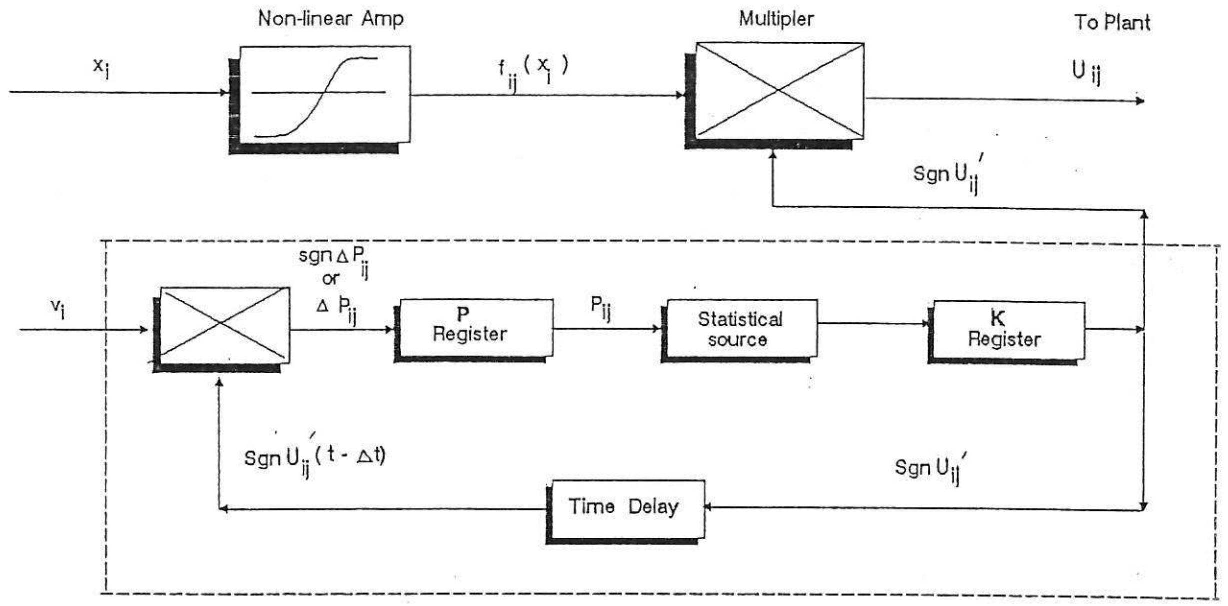

FIG 7: ACTUATION LOGIC DIAGRAM

with simultaneous actions on the object. A delayed action makes the quality of the control process worse. But we can control successfully only when the properties of the object are known accurately. Meanwhile the study of the object requires time. Too rapid a controller will perform unjustified control actions, which will not be sustained properly by the information obtained as a result of the study of the object. Too cautious a controller will wait for an unnecessarily long time, accumulating information, and will not be able to direct the object in time to the required conditions.

Viewed in this framework, the non-linear amplifier in SOC Activation logic is driven by the x_i signals to produce the working part of the control action. The conditioning logic units are then responsible for the investigating and testing part of the action, although these units can sometimes act to improve upon outputs of the non-linear amplifiers (in which the conditioning logic units become "Partially perceptive and partially directive" as described above). Finally the output multiplier of the activation logic subsystems may be viewed as the principle means of blending dual characteristics of the SOC.

7.2 NON-LINEAR AMPLIFIER

In design of time optimal control systems, the theory calls for control at all times, viz,

$$u_{ij} = | u_{ij} | \max sgn\, x_i\, sgn\, u'_{ij} \qquad (20)$$

This suggests use of threshold detectors to obtain $sgn\, x_i$. In practice however, a somewhat more complex element is generally used to compute a function $f_{ij}(x_i)$ which is then employed as follows,

$$u_{ij} = f_{ij}\,(x_i)\; sgn\, u'_{ij} \qquad (21)$$

the one essential requirement is

$$sgn\, f_{ij}\,(x_i) = sgn\, x_i \qquad (22)$$

The optimal controller solution is meaningful only when x_i is large, because the total time of the transient process is increased only slightly by use of non-optimal functions close to the switching surfaces.

A common form of this amplification function is the soft limiter as shown in Fig 7. The soft limiter provides control for large x_i values, but for small x_i values the linear region of the amplifier characteristics acts to reduce oscillations about the switching surface.

7.3 CORRELATION LOGIC UNIT

The correlation logic unit is in fact the PSV module described in section 5. It is used to identify the respective elements of the polarity matrix. At the input of the appropriate logic unit (Fig 7) the product of V_i or $sgn\ V_i(t)$ and $sgn\ u'_{ij}\ (t - \Delta t)$ is obtained, with $sgn\ u'_{ij}$ delayed one sample interval, Δt, so that the cause and effect relationship between $sgn\ u'_{ij}$ trials and the resulting $sgn\ V_i$ or V_i values may be established.

7.4 SOC FOR MULTIPLE GOAL MULTIPLE ACTUATOR CONTROL

Plants having multiple-goals and multiple-actuators often exhibit an inter-dependence among goals. Thus, the operation of one of the actuators for control of one of the goals will result in a significant change in the remaining plant variables thereby requiring that the system be capable of control despite the inter-dependence of the several goals and actuators in the plant. High speed SOC provides simultaneous, multiple-goal multiple-actuator control in which the instantaneous influence of each actuator or error signal is identified and a self-organising controller compensates for changing the polarities of actuator, direct and cross-coupled effects [Barron 1967(b)].

A schematic diagram of the entire system using SOC is shown in Fig 8 where multiple command inputs X_{c1}... X_{ck} from an external source are all simultaneously fed to a summing point with measured response variables X_{mi}..X_{mk} which are provided by the sensors, there normally being one sensor for testing the response of each of the variables or goals of the plant. The summing point

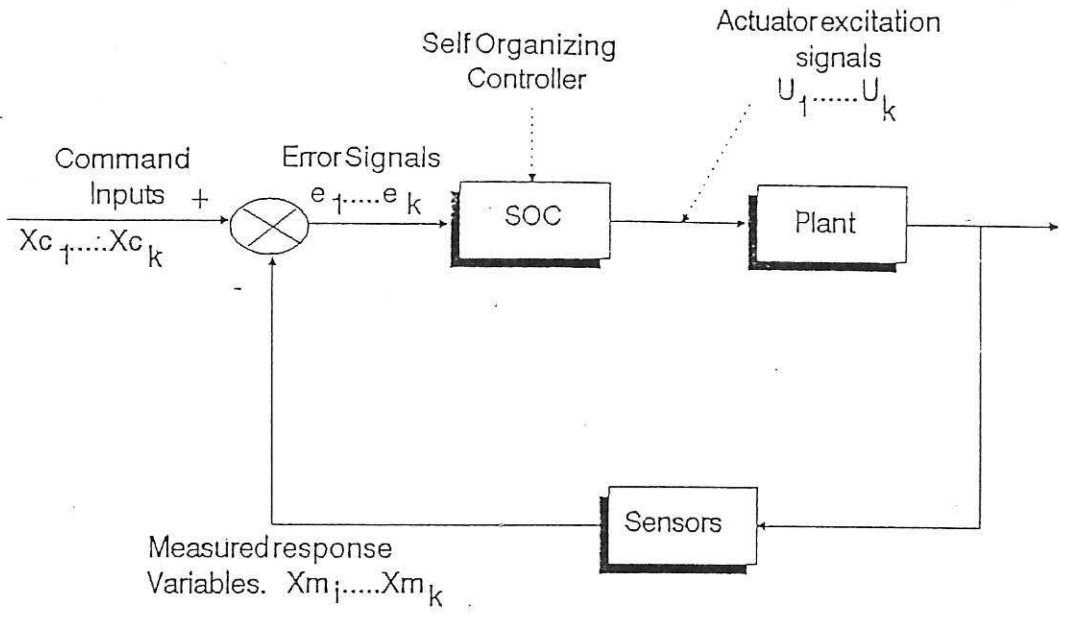
Self Organizing Controller
Actuator excitation signals $U_1 \ldots\ldots U_k$
Command Inputs +
$Xc_1 \ldots Xc_k$
Error Signals
$e_1 \ldots\ldots e_k$
SOC
Plant
-
Sensors
Measured response Variables. $Xm_i \ldots\ldots Xm_k$

FIG 8: SCHEMATIC DIAGRAM OF SOC FOR MULTIPLE INPUT OUTPUT SYSTEM

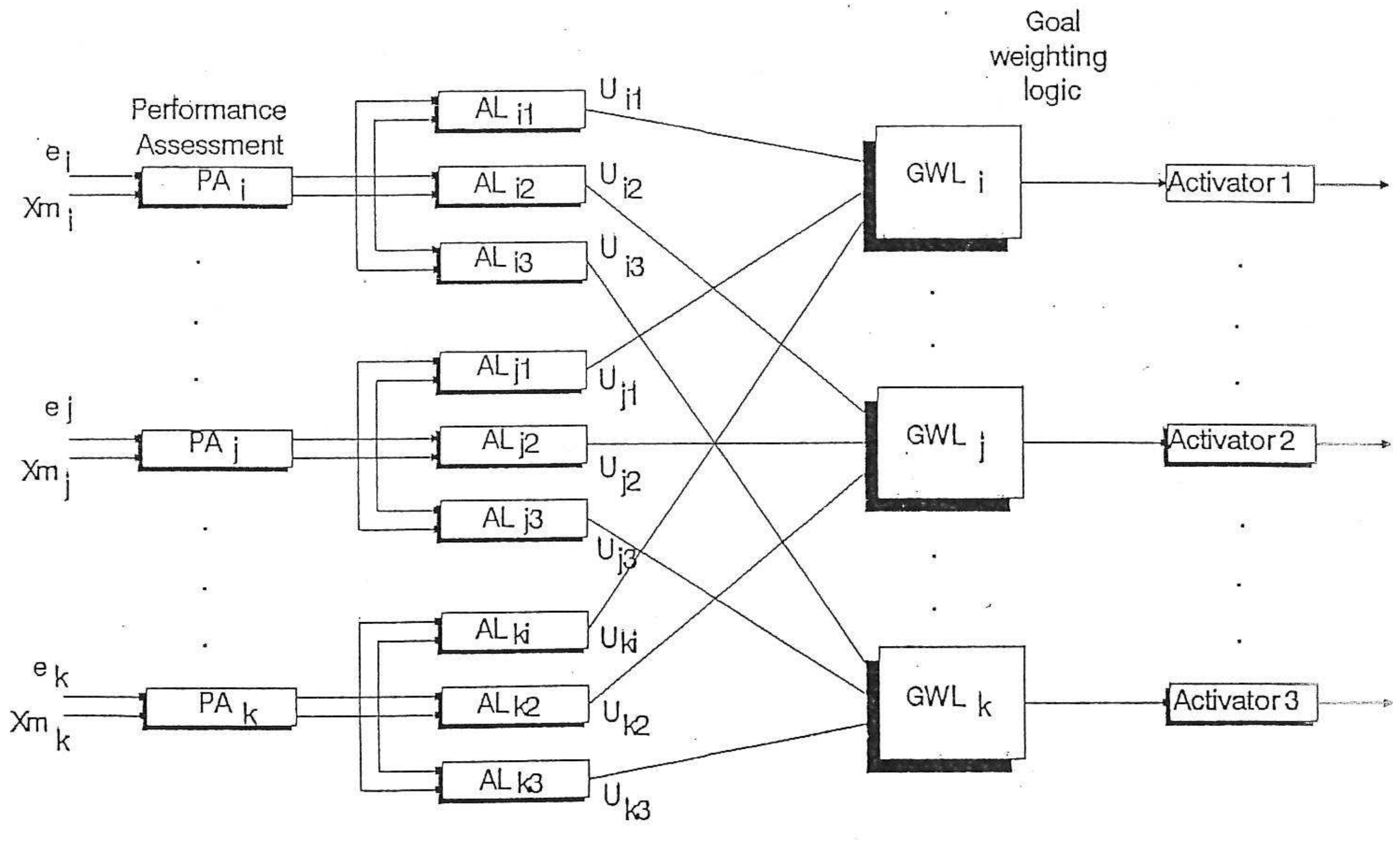

FIG 9: MULTIPLE ACTUATOR SOC

provides a plurality of error signals $e_1...e_k$ to a SOC system which provides the actuator excitation signals $U_1...U_k$ based upon the evaluation of the system performance.

A SOC for a multiple-actuator system is shown in Fig 9. The controller includes a combination of performance assessment units $PA_1...PA_k$, there being one PA unit for each goal of the system and all working in parallel. All PA units synchronise to evaluate the performance relating to the variable with which each is associated. The output from each PA unit is fed into a combination of logic circuits $AL_{ii}..AL_{ji}..AL_{ki}$, each activation logic circuit being composed of a coupling unit, a clock, and a PSV unit. For each PA unit the number of activation logic units is equal to the number of actuators. All these activation logic units work in parallel and the output signals $U_i...U_k$ are fed to the goal weighting logic (GWL) circuits $GWL_1..GWL_k$, there being one GWL unit for each actuator. The GWL circuits provides actuator excitation signals $U_i...U_k$ in response to signals applied, and thereby alter plant operation.

8. SIMULATIONS AND RESULTS

The SOC controller described in the preceding sections was applied to control the process used for drug administration for muscle relaxation in operating theatres, as described in section 2. To compare the timing details on various processors the simulations were performed in three phases, each time using a faster machine.

Phase1.

Using a simulation package "PSI" on a mini computer (PERKIN ELMER) 3140

Phase2.

Using Fortran to simulate the self-organising controller and the process on a SUN Unix system.

Phase3.

Using OCCAM to simulate self-organising controller and the process on a TRANSPUTER.

In the following sections each step is described in detail

8.1 Phase 1

Interactive Simulation Package (PSI), is a block-oriented interactive simulation program designed to study the behaviour of dynamic, continuous systems. The structure of the simulation model is described by means of a block diagram. Forty three different block types are available including integrators, time delays, linear and non-linear elements, logical units and some block types to simulate discrete systems such as zero-hold, a discrete PID controller and unit delay. The program is controlled interactively by use of a command language. Using this language a system can be defined and simulated with output directed to the graphical terminal. During simulations, output of up to four blocks can be stored for latter study.

The SOC was programmed using combinations of different blocks available, e.g, shift registers used in the PSV module were programmed using a combination of AND, OR gates and INVERTOR blocks. In all, 84 blocks were used for the complete system simulation including the SOC and the process. The transfer function of muscle relaxant model is given by

$$G(s) = \frac{Ke^{-T_1 s}}{(1 + T_2 s)(1 + T_3 s)} \tag{23}$$

A sampling interval of 1 minute was used with a total experimental length of 200 minutes for each simulation. The procedure was repeated for a range of parameters in equation 23. Here results are presented for either a constant or a varying set point in Fig 10 and 11 respectively, where the time constants are T_2 = 1 min and T_3 = 10 min. The results obtained were satisfactory in terms of speed and rise time. The overshoot and settling time for a variation of parameters in the controlled process demonstrates the controllability of SOC.

8.2 PHASE 2

The SUN Unix-based workstation is a faster machine compared to the Perkin Elmer mini-computer. The SOC algorithm and the muscle relaxant anaesthetic model were coded in Fortran. The time constants used in equation 23 were similar to those used in phase 1.

T_1 = 1 Minute

T_2 = 1 Minute

T_3 = 10 Minutes

The result for a constant set point is presented in Fig 12, again showing a good response

8.3 PHASE 3

The complex arrangement of SOC with the large amount of logic, presents a problem for sequential computing and simulation. The main objective was to implement the self-organising controller on a transputer and investigate the efficiency induced due to parallelism. The SOC algorithm and the anaesthetic model of Pancuronium were coded in Occam. As the SOC algorithm is purely sequential for a single input/output processes, pipeline approach of parallelism was adopted. In a pipeline, successive instructions are executed in overlapped fashion. thus reducing the computation time. Here the pipeline consisted of 4 processes, namely the SOC algorithm which formed 3 processes, and the simulation of the controlled muscle relaxant model which formed the 4th process.

The results for a constant set point, with the same time constants as used in phase2, is presented in Fig 13. The timing details for transputer T414 (without a floating point unit) and transputer T800 (with a floating point unit) along with the timing detail of sequential machines are presented in Table 1. The time recorded is for a model simulation time of 200 minutes with a sampling interval of 1 minute.

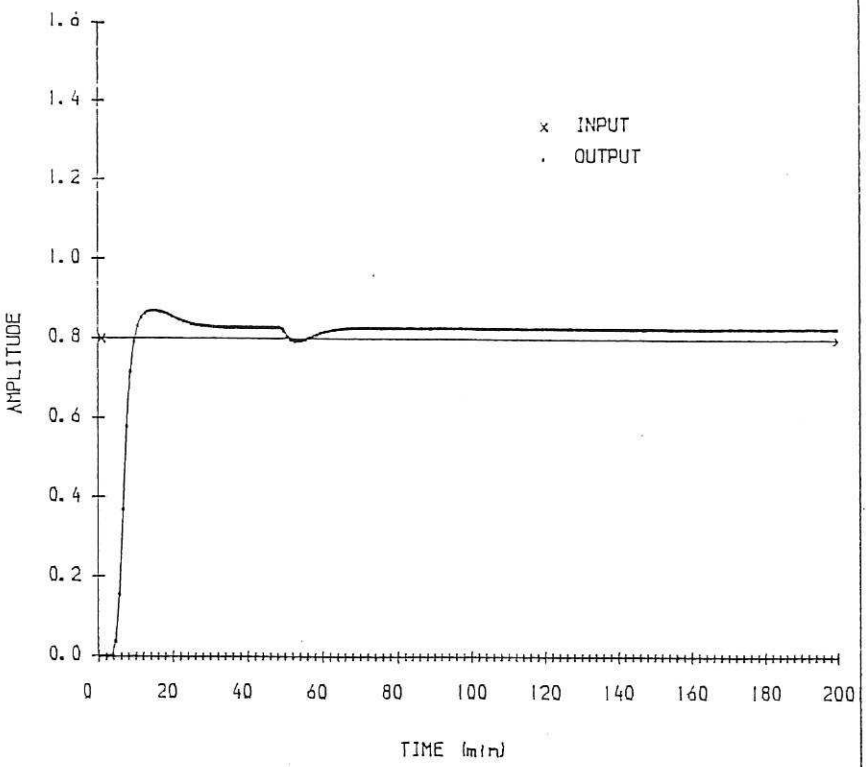

FIG 10: SOC ON MINI COMPUTER (PSI SIMULATION) FOR PANCURANIUM DRUG, TIME CONSTANTS OF 1,10

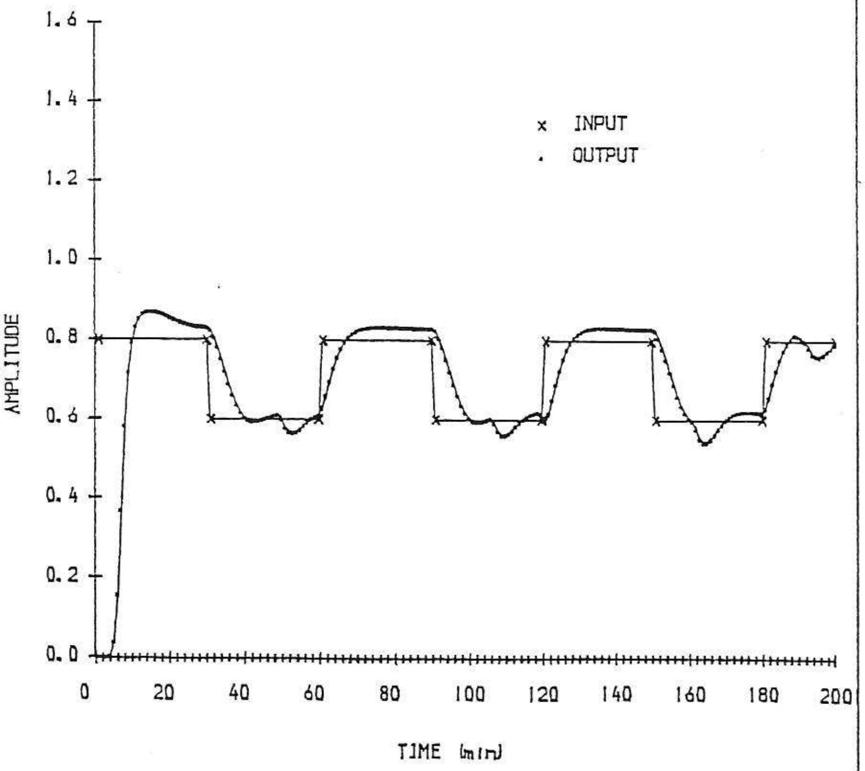

FIG 11: SOC ON MINI COMPUTER (PSI SIMULATION) FOR PANCURANIUM DRUG, TIME CONSTANTS OF 1,10

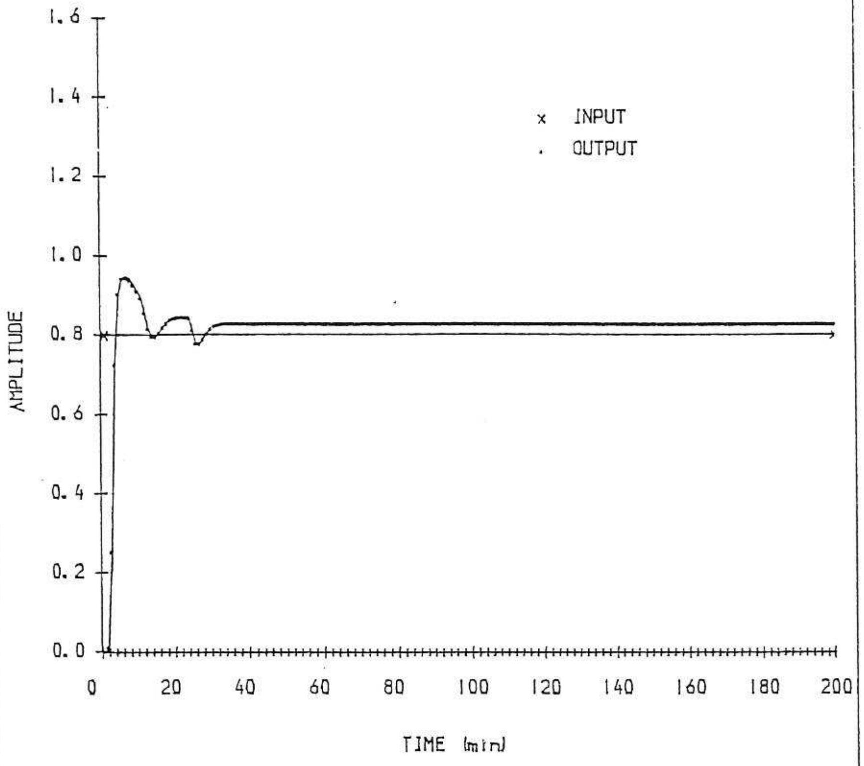

FIG 12: SOC ON SUN UNIX SYSTEM (FORTRAN SIMULATION) FOR PANCURANIUM DRUG, TIME CONSTANTS OF 1,10

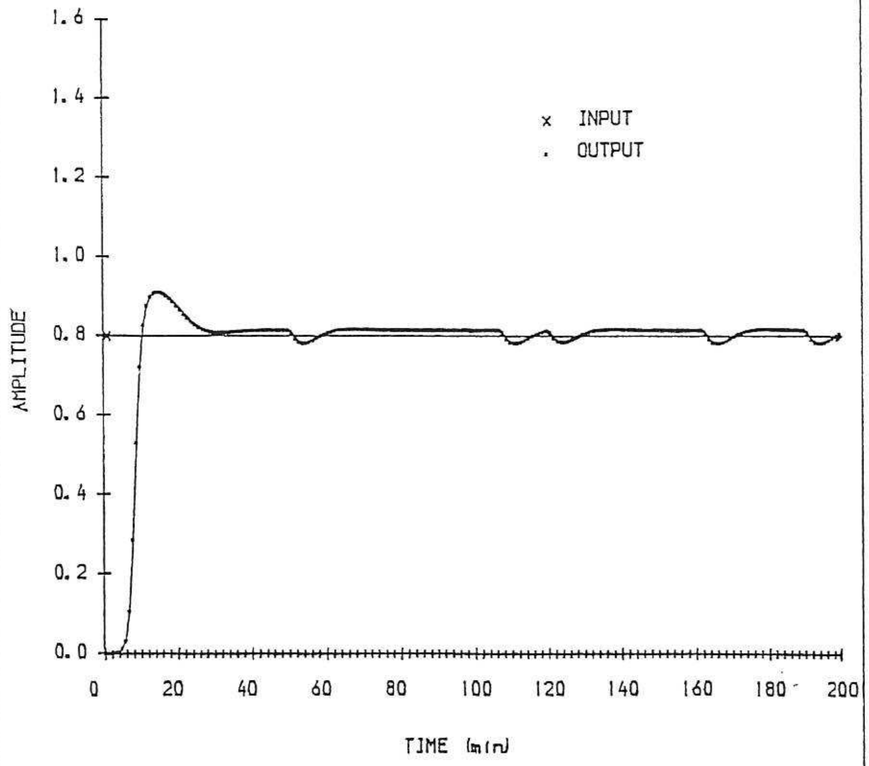

FIG 13: SOC ON TRANSPUTER (OCCAM SIMULATION) FOR PANCURANIUM DRUG, TIME CONSTANTS OF 1,10

Table 1 :TIMING DETAILS	
Processor	Time taken by SOC
Perkin Elmer (mini computer)	3 Min 10.20 sec
SUN Unix System	8.03 sec
T414 Transputer	4.42 sec
T800 Transputer	0.874 sec

The parallel arrangement inside the SOC, particularly for multiple-goal multiple-actuator control presented in section 7 can be realised via transputers, thereby enhancing the power and speed of control. SOC shown in Fig 9 can be realised on transputers by different combinations. Fig 14 presents one combination for a 3 input 3 output system. The parallelism can be introduced at 4 levels. (1) Calculating P.A signal. (2) Forming AL units (3) Combining GWL units and (4) Producing actuator outputs. The host transputer transfers data to the relevant PA modules implemented on transputers A_1, A_2 and A_3. Outputs from the first layer transputers are fed into AL units implemented on a combinations of B_{ij} transputers, in the second layer. The number of AL units for each PA unit is equal to the number of actuators which in this case is three. The output from the second layer is synchronised to go back to the first layer transputers, which also contains the code for the respective GWL and the actuator. The actuator excitation signal U_i generated by the respective GWL unit is fed back to the process simulated on the host transputer.

9. CONCLUSIONS

The Self-Organising Controller described in this chapter is an evolutionary form of adaptive controller in which guided random search techniques using methods of fast assessment of search results, are used to achieve flexibility and speed of adaptation. Basically the SOC combines statistical decision theory (to determine the true instantaneous plant performance), prediction theory (to

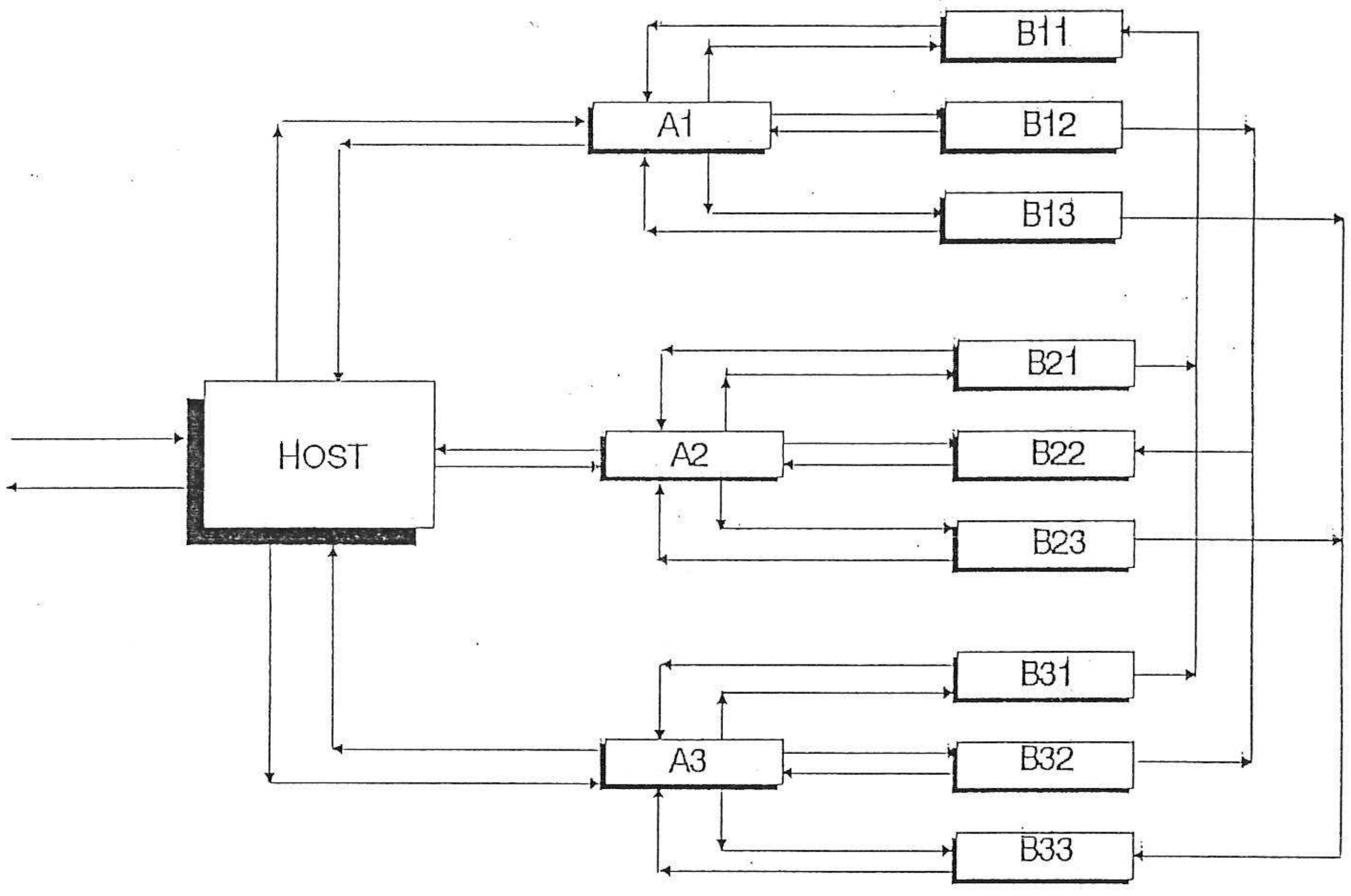

FIG 14: TRANSPUTER ARRANGEMENT FOR MULTI INPUT OUTPUT SOC

determine the performance trend), and rapid trial generation (to ascertain what must be done to improve the performance trends). The key to the SOC lies in the modification of internal paths by on-line changes in controller Probability State Variable PSV and using registers for short term memory intervals.

The uniform response characteristics produced by SOC over different combinations of time constants induced in the muscle relaxant model have been presented. The SOC simulations on a transputer illustrates the effectiveness of this implementation (Table 1). The parallel presentation provides guide lines for future work on multi-input multi-output SOC. As shown in Fig 14 multi-input/output SOC can be realised in a simple and effective way, as the number of inputs and outputs increases. Depending on the system, the effective parallelism also increases accordingly, thus providing a highly reliable, efficient and faster system. In future work it is intended to investigate SOC for multivariable biomedical systems, one example being for the simultaneous drug-infusion control of unconscientiousness and muscle-relaxation using a dual regime such as isoflurane and atracurium. Another possible application is that of blood pressure management either in intensive care or operating theatre. In this area, both single variable and multivariable on-line drug infusion schemes have been investigated. Thus, using Sodium Nitroprusside (SNP) Sheppard & co-workers have performed extensive control studies which have elicited detailed mathematical models (Slate etal, 1979), and applied both simple control strategies (Sheppard etal 1979) and adaptive control techniques (Slate, 1980). Other algorithms such as pole placement and generalised Minimum Variance have been investigated by Mansoor and Linkens (1989) and Millard etal (1986) respectively. In the multivariable case, the simultaneous control of Mean Arterial Pressure (MAP) and Cardiac Output (CO) via SNP and dopamine has been investigated by Voss etal (1987) and McInnis and Deng (1985) using simple difference equation models. More recently, Mansoor and Linkens (1989) have studied multivariable pole-placement self-tuning using a more realistic dynamic mathematical model derived via physiological considerations.

Another strand of control design for systems which do not have a known mathematical model is that based on fuzzy logic, Zadeh (1965). It is initially studied in forms of industrial applications, an early example being that of a steam engine/boiler combination (Mamdani and Assilian, 1975). Only recently has fuzzy control been considered for biomedical purposes, which is surprising considering the difficulty that there is in obtaining mathematical models for many physiological processes. Thus fuzzy logic control has been investigated for muscle relaxant administration (Linkens and Mahfouf, 1988) via extensive simulation. Also, Ying etal (1988) used a fuzzy logic "inference shell" to produce a controller for arterial blood pressure, and Vishnoi and Gingrich (1987) report an application for gaseous anaesthetic agent delivery. The concept of self-organisation can also be attached to fuzzy logic, the first consideration being that of Procyk (1977). Further studies on self-organising fuzzy logic controller (SOFLC) by Yamazaki and Mamdani (1982) were followed by Daley and Gill (1984) who applied it to attitude control of a spacecraft. SOFLC has also been investigated for aircraft control (Larkin, 1985) and for the control of autonomous guided vehicles (Harris and Moore, 1989). In biomedicine, it has been applied to muscle relaxant anaesthesia by Linkens and Hasnain (1989), showing it to be robust to noise contamination and model parameter uncertainity (Hasnain, 1989).

Hence, the SOC structure of Barron described in this paper takes its place alongside numerous other approaches, against which it should be compared in the future for suitability in biomedicine. Certainly, its basic motivation for control of imprecisely known systems makes it attractive for life science applications. Further investigations into its multivariable implementation mentioned above and its comparision with other forms of self- adaptive system are planned for future.

References

ASTROM, K.J AND PRITTENMARK, B, (1984). ''Computer controlled systems: Theory and Design,'' *Prentice Hall.*

BARRON, R.L, (1965). ''Self-organizing Space craft Attitude Control,'' *Final technical report, Air Force Flight Dynamics Lab, AFFDL-TR-65-141.*

BARRON, R.L, (1966). ''Self-organizing Control of Aircraft Pitchrate and Normal Acceleration,'' *Final technical report, Air Force Flight Dynamics Lab, AFFDL-TR-66-91.*

BARRON, R.L, (1967(a)). ''Analysis and Synthesis of Self-organising Control Systems,'' *Final technical report, Air Force Avionics Lab, AFFAL-TR-67-93.*

BARRON, R.L, (1967(b)). ''Self-Organising Control Systems for providing Multiple Goal, Multiple-Actuator Control,'' *United States Patent Office No: 3519998.*

BARRON, R.L, (1968(b)). ''Analysis and Synthesis of Self-organising Control Systems II,'' *Final technical report, Air Force Avionics Lab, AFFAL-TR-68-238.*

BARRON, R.L, (1968(b)). ''Self-Organising and Learning Control Systems,'' *Cybernetics Problems in Bionics.*

BARRON, R.L, (1969(a)). ''Self-Organising Control of Advanced Turbine Engines,'' *Final technical report, Air Force Propulsion Lab, AFAPL-TR-69-73.*

BILLINGS, S.A AND C.J, HARRIS., (1981). ''Self-tuning and Adaptive Control: Theory and Applications,'' *Peregrinues, London.*

DALEY, S AND GILL, K.F, (1986). ''A design study of a self-organising fuzzy logic controller,'' *Proc Inst Mech Engrs*, vol. 200, pp. 59-69.

FELD'BAUM, A.A, (1965). ''Optimal Control Systems,'' *Academic press NewYork.*

HARRIS, C.J AND MOORE, C.G, (1989). ''Intelligent identification and control for autonomous guided vehicles using adaptive fuzzy based algorithms,'' *Department of Aeronautics & Astronautics, Southampton University.*