Monograph:
Marriott, S. and Harrison, R.F. (1995) A Self-Organising State Space Decoder for Reinforcement Learning. Research Report. ACSE Research Report 582 . Department of Automatic Control and Systems Engineering

A Self-Organising State Space Decoder for Reinforcement Learning

Shaun Marriott and Robert F. Harrison

Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street
Sheffield, S1 3JD, U.K.

Research Report No. 582

June 1995

A Self-Organising State Space Decoder for Reinforcement Learning

Shaun Marriott and Robert F. Harrison

Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street
Sheffield, S1 3JD, U.K.

Abstract

A novel self-organising architecture, loosely based upon a particular implementation of adaptive resonance theory is proposed here as an alternative to the fixed state space decoder in the seminal implementation of reinforcement learning of Barto, Sutton and Anderson. A well known non-linear control problem is considered and the results are compared to those of the original study. The objective is to illustrate the possibility of neurocontrollers that adaptively partition state space through experience without the need for a priori knowledge. Input / output pattern pairs, desired state space regions and the network size / topology are not known in advance. Results show that, although learning is not smooth, the novel reinforcement learning implementation introduced here is successful and develops an effective control mapping. The self-organising properties of the new decoder allow the neurocontroller to retain previously learned information and adapt to newly encountered states throughout its operation, on-line. The new decoder increases its information capacity as necessary. The adaptive search element and the adaptive critic element of the original study are retained.

Keywords- Adaptive resonance theory, fuzzy ART, reinforcement learning, incremental clustering, Euclidean clustering, neurocontroller, self-organising systems.

1. Introduction

1.1 The incremental Paradigm and ART

Artificial neural networks which learn incrementally by adding new nodes or processing elements during operation have been used to approximate mappings (Platt, 1991; Kandirkamanathan and Niranjan, 1992). This technique obviates many of the problems associated with fixed network structures such as that of ascertaining the optimum network size / configuration (Fujita, 1992), deciding upon a connection topology and providing sufficient information capacity (complexity) for adequate representation of the problem domain.

Incremental learning is especially useful in situations where information is gathered and used on-line. In many situations, it is not enough simply to train a neural network on a

given collection of data and leave it to operate without further adjustment through experience. What if conditions arise which have not yet been encountered by a trained network? Does new information necessitate retraining? What happens to the existing body of information represented by the network if new information is incorporated? Some fixed network structures suffer the double problems of requiring off-line retraining to deal with new conditions and *catastrophic forgetting* where an established mapping is replaced by a new one (Sharkey and Sharkey, 1994).

One class of neural network architectures especially suited to increasing learning capacity through experience is that based upon adaptive resonance theory (ART) (Grossberg, 1980; Carpenter and Grossberg, 1987a, 1987b, 1989; Carpenter *et al,* 1991a, 1991b, 1992). ART networks have the capability of dynamically allocating nodes as required during processing, without the need for retraining, to incorporate novel information; this property provides a natural basis for on-line adaptive learning. This paper presents a novel network, based loosely upon some of the principles of ART, which acts as a self-organising state space decoder to provide an internal representation of state space.

1.2 Control Systems and the Problem of Delay

An area of application for the incremental paradigm is the dynamic partitioning of state space for control and related problems. Very often, it is difficult to establish more than crude qualitative information about state space trajectories on all but the simplest of analytical systems. Ascertaining an accurate model of system dynamics and contriving an objective or cost function signifying desired behaviour, is usually the preferred route in optimal control problems. Most adaptive methods are indirect and use an estimated system model to recompute controls at each step (Sutton *et al*, 1992). Even if adequate knowledge is available, the *a priori* integration of this knowledge into the network structure can severely limit the autonomy and flexibility of the network. Autonomous learning systems need to be able to extract and organise information during experience in their particular data rich environment, increasing their information capacity as necessary.

Consideration of autonomous, self-organising systems reveals another, related, aspect. The world exhibits obscure structure to any observer and convenient labels, indicating the spatiotemporal significance of, and relationships between, objects or events are simply not available *a priori* (Edelman, 1989). It is very clearly ordered but we often do not understand the causal relationships between objects and events. For example, living organisms make sense of the world through experience and evaluation of behavioural consequences. They structure experiences autonomously and develop conceptual schemata with which to classify perceptual stimuli as a basis for future behavioural responses. Where desired responses are available for neural network training, the initial learning problem has been solved autonomously by a human operator who has organised and correlated relevant information to provide training data for the associative network. This is especially true in control applications where desirable control actions have to be specified and presented to a given neural network along with the conditions which necessitate such actions. To increase neural network autonomy, the processes of information extraction and organisation must be incorporated into the architecture to allow more intelligent use of "raw" data.

How data is used by a neural network depends upon the learning method used. Within the field of artificial neural systems, we can distinguish three broad classes of learning methods *viz.*

- *supervised learning*, in which input-output pattern pairs (consisting of a stimulus and desired response) are applied repeatedly to a neural network until a specified degree of learning is achieved,

- *unsupervised learning,* in which input patterns alone are supplied to a neural network, without regard to any pre-established categories, so that the neural network might learn to discern intrinsic features within a set of input data, and

- *reinforcement learning*, which allows neural network operation within information-poor environments which often provide little feedback; available feedback is usually in the form of a scalar evaluation of network performance following a series of control actions.

For the supervised case, the provision of input-output pattern pairs by an external teacher is an artificial process which relies upon several underlying presuppositions for its operation as a training method. One such being that of a temporal connection between input-output pairs; the nature of the assumed temporal connection forms the basis for a model of the state transition dynamics of the system being controlled. In other words, assumptions regarding the relative timing between stimuli and responses in a system determines the form of the system model; if those assumptions are wrong, or cannot be accommodated by the neural network architecture being used, then control is unlikely to be successful. An adequate representation of state transition dynamics is a prerequisite for successful control as these dynamics determine system responses to stimuli through state transitions that depend on both the present state and the current input. As well as a possible change in the current response, there might be a transition to a new state. These dynamics characterise a system and must be represented in some way by a candidate neurocontroller.

A sizeable proportion of neural network theory is based upon *associationism* which has its historical roots in psychology (James, 1892). Learning laws which associate pre-synaptic and post-synaptic outputs (Hebb, 1949) often assume little or no time delay between correlated signals. Networks based upon these learning laws, and variants of them, function as simple pattern associators which strengthen connections between frequently associated patterns and which weaken others.

The effects of an input on state transitions are not usually limited to instantaneous changes unless memoryless systems are considered; a more accurate assessment of real world systems is that state transitions are influenced by inputs as a function of the time interval between a particular input and a given state transition. This temporal effect reduces the validity of simplistic stimulus-response pairing of input and output to some extent. Problems which involve delayed feedback to a learning system can be reduced to simple pattern association tasks but require a problem to be solved beforehand by a human teacher in order to specify the optimal actions which should be taken by the learning system (Myers, 1992). One way to take delay into account is to present delayed inputs as part of the pattern pair. However, this requires assumptions about

the system model; for example, what is the minimum delay time that can be assumed for a good model? Incorporating time delay information into the training data set increases the dimensionality of the input space.

Both the difficulty of obtaining relevant input-output pairs and the issue of the temporal connection between inputs and state transitions (and, therefore, outputs) are addressed by the paradigm of reinforcement learning (Barto *et al*, 1983; Sutton, 1988, 1992; Barto, 1992; Sutton *et al*, 1992; Daynan & Hinton, 1993); this paradigm will be considered in section 2 below.

1.3 An overview

Following Michie and Chambers (1968) and Barto *et al* (1983) the cart-pole system problem is used to exemplify some of the characteristics that distinguish neural networks as autonomous learning systems from other available data processing methods. The characteristics of autonomy and adaptability are among the most important. As a test problem, The cart-pole system provides an example of a highly non-linear system involving the characterisation of complex state-space trajectories. Standard solution methods require assumptions about the form of the control force function and an objective function (Anderson, 1989; Hocking, 1991). Furthermore, such techniques rarely generalise and, thus, require an *a priori* analysis of each dynamical system encountered. Like Barto *et al* (1983), we assume that the available feedback is of much lower quality than is required for both standard control techniques and for supervised learning techniques. Furthermore, we believe that similar assumptions can be made about the state space partitioning problem where any autonomous system will have limited information about the structure of state space in advance of experience. Indeed, merely specifying a fixed partitioning *a priori* makes assumptions about the granularity of the resultant control mapping and constrains the available adaptive procedures within a pre-specified temporospatial structure.

Standard state space methods can be used to obtain a linear model as an approximation to a non-linear system and to design a closed-loop feedback controller (via pole placement) to control the system within a limited region of state space (e.g. Friedland, 1987). This control method requires an *a priori* model of the dynamical system, obtained by using the simplification of linearisation to render the problem amenable to linear techniques. more sophisticated approaches using feedback linearisation, for example, may extend the neighbourhood of effective control but are still highly dependent on highly accurate *a priori* models. For many desirable control purposes, however, such models may not be available or may contain too many analytical simplifications which render the proposed control system incapable of following the complex dynamics of the real system under consideration.

The intention here is not to simply develop another controller for a particular non-linear control problem; it is to explore some of the issues for which the cart-pole problem provides a convenient example and to indicate the possibilities of developing flexible, general purpose controllers capable of adapting to a given dynamical system with a minimum of *a priori* information.

Section 2 of this paper explores reinforcement learning as a paradigm and goes into more detail about a particular implementation; this implementation and its modifications form the core of the present work.

In section 3, The state space decoder, a sub-module of the reinforcement learning implementation, is singled out as important and provides a focus for modifications leading to a novel approach to the problem of state space characterisation. The novel approach is discussed and illustrated in the simulations of section 4. This self-organising approach, consisting of an adaptive state space decoder coupled with an implementation of reinforcement learning, constitutes a controller that can be applied to a non-linear control system almost immediately without the need for extensive analysis and design. Also, a further refinement to the new approach, which resulted from experience gained through the simulations, is proposed and tested. Finally, section 5 presents the conclusions drawn from the combination of ART-based methods and reinforcement learning, considers problem areas and puts forward suggestions for further work and future directions.

2. Reinforcement learning

2.1. Introduction.

Reinforcement learning (Barto *et al*, 1983; Sutton, 1988, 1992; Barto, 1992; Sutton *et al*, 1992; Daynan & Hinton, 1993) arose out of earlier work based upon classical (stimulus-response) conditioning (Pavlov, 1928; Hebb, 1949; Hull, 1943, 1951, 1952; Hilgard and Bower, 1966; Sutton & Barto, 1981, 1990; Barto & Sutton, 1982; Klopf, 1986, 1988). In its simplest formulation it consists of using a single scalar variable to represent the punishment / reward status of an artificial neural system with respect to the environment in which it is operating. This signal is fed back to the learning system by a critic which rewards favourable system responses and punishes undesirable ones. Earlier work (Michie and Chambers, 1968) was entirely failure driven. The system considered here is the seminal implementation of Barto, Sutton and Anderson (BSA) (Barto *et al*, 1983) which consists of an associative search element (ASE) and an adaptive critic element (ACE); the latter being responsible for interpreting the success / failure status of the ASE sub-system. In the original form, the ASE and ACE are both implemented using a single adaptive element but this is not a necessary condition.

As mentioned in the introduction, learning laws proposed in the form of modified Hebbian synapses (Hebb, 1949) are inadequate for significant number of purposes. Consequences of actions do not always follow immediately from the environment and varying degrees of temporal association have to be taken into account since delays are often present (Myers, 1992).

In addition to the problem of delayed feedback, there is also the problem of *credit assignment* which involves the distribution of credit or blame between contributing nodes. For reinforcement learning, the scalar feedback signal assesses the performance of the network as a whole (Barto *et al*, 1983) and, thus, reflects the importance of the ensemble of actions in eliciting a given environmental (system) response rather than the single input and the subsequent near-instantaneous response implied by learning laws based upon simple associationism.

While neural network systems utilising the unsupervised learning method require neither explicit pattern pairs nor evaluative feedback *per se* to operate effectively, they are only able to organise input patterns by means of clustering methods and have no intrinsic means for adjusting control actions on the basis of environmental responses. External learning mechanisms have to be incorporated into candidate self-organising controllers based upon such clustering networks. These external mechanisms can, for example, involve the use of stimulus-response pattern pairs, a cost-function or scalar evaluative feedback; the latter approach being taken here.

A self-organising controller, based upon a Kohonen topology conserving network, was developed to learn the control actions of a teacher in supervised learning mode (Ritter *et al*, 1992). A variant akin to reinforcement learning, using only a reward signal based upon a specified cost function, has also been developed (Ritter *et al*, 1992). In both cases Kohonen's original learning algorithm (Kohonen, 1989) has been extended to incorporate an output value for each node of the network lattice. In the supervised case, the cart-pole problem has been solved by a teacher external to the network which

acts as a look-up table following training. Although this method obviates the need for re-calculation of output values, the requirement of an external teacher limits the autonomy of the network.

The variant removes the requirement for an external teacher and computes desired outputs on the basis of a generalised reward signal derived from a system specific cost function. The network no longer has access to desired control outputs and forms a continuous mapping between state space and control output space, with the control outputs being determined via a stochastic search process. During the search, the stored output value for a particular lattice node is allowed to converge to a desirable control action.

Both the supervised and the variant topology conserving controllers have a planar network lattice structure which is fixed *ab initio*. This places a restriction on the information capacity of the network through the determination of the state space resolution by the size of the lattice. In other words, with too few nodes the control hypersurface will be coarsely defined. Too many nodes may reduce the parsimony of the network depending upon the size of the local update region with respect to the granularity of state space coverage. However, with heuristic determination of network size, the problem may not be of practical importance.

2.2 Two tasks

In this paper we explore a hybrid approach which combines the flexibility of self-organisation with the adaptability of reinforcement learning and its suitability to information-poor environments. The control problem is resolved into two decoupled tasks *viz.*

- the *self-organisation task*, which involves the autonomous categorisation of input information to provide a basis for subsequent control actions, and

- the *control action learning task*, which involves the evaluation and correction of elicited control actions associated with individual states or distinct regions of state space represented internally by the neurocontroller.

The hybrid approach is possible because the two tasks are decoupled and can be solved independently. For example, solution of the control action learning task only requires the assumption that individual regions of state space are represented such that they provide a unique 'key' to associated control actions which are stored separately from the internal representation of state space. The assignment of credit or blame and the updating of individual control actions is not linked inextricably to the method of state space partitioning provided that there is a one-to-one correspondence between the evaluated region of state space and its associated control action.

Decoupling the tasks also allows the possibility of other related neurocontroller methods. Indeed, as stated here, the first task is too restrictive and can be replaced by the more general

- *decoder task*, which involves the decoding of state space into a representation with which individual control actions can be associated.

Note that we have used the term 'decoder' as opposed to the term 'encoder'; this usage is consistent with the BSA formulation of reinforcement learning and reflects the analogy between the BSA state space decoder and a computer memory address decoder which decodes input addresses to allow access to physical memory locations (Barto *et al*, 1983). The state space decoder task is to treat the state as a decoder "address" pointing to an associated control action stored within the neurocontroller.

The statement of the decoder task says nothing about its nature; for example, it can have a fixed structure and act as an indexing system for a control action look up table (Mitchie and Chambers, 1968) or it can be self-organising as is the decoder presented here. Also, both discrete-valued and continuous-valued decoders are possible; the latter consisting of a smooth mapping between a continuous input space and a continuous output space.

The original BSA approach can be decoupled into the decoder task and the control action learning task. Here the decoder indexes a fixed state space representation which is in the form of a look-up table; as learning proceeds, the look-up table is filled in (Barto *et al*, 1983). It is precisely because of the decoupling between the two tasks that other types of state space decoder are possible whilst retaining the original BSA implementation of reinforcement learning.

Fuzzy-logic-based neurocontrollers allow decoupling between the two tasks by treating the decoder task as one of determining *rule antecedents* and the control action learning task as one of determining associated *rule consequents* (e.g. Berenji and Khedkar, 1992; Jang, 1992,1993; Jang and Sun, 1995).

The generalised approximate-reasoning-based intelligent control (GARIC) architecture of Berenji and Khedkar (1993) uses an action selection network (ASN) and an action evaluation network (AEN) which are analogous to the ASE and ACE respectively. Here, although the decoding task is independent of the control action learning task, the state space decoding is not carried out by a separate decoder system but is subsumed within the operations of the ASN. States are decoded into the constituent *terms* of *linguistic variables* where each linguistic variable consists of a set of terms. One term is selected from each individual linguistic variable; the selected term represents the "value" of the linguistic variable, e.g. the term "near zero" might be selected from the set of terms comprising the linguistic variable "velocity". The resultant collection of selected terms, consisting of a single term from each set, comprises a rule antecedent. Note that individual states are not represented by individual rules in this case; a single state can fulfil the antecedent conditions of more than one rule and thereby trigger multiple rules. Rule antecedents can be viewed as characterising distinct regions of dynamical space which overlap in places where multiple rules are involved. Control actions are associated with input states through fuzzy encoding as rule consequents. GARIC uses reinforcement learning to tune the fuzzy rule base so that it is able to represent the desired control mapping. The fuzzy encoding of both the input terms and the output terms is tuned adaptively. A fixed number of rules is used to solve the cart-pole problem.

The adaptive-network-based fuzzy inference system (ANFIS) of Jang (1992,1993; Jang and Sun, 1995) also tunes the fuzzy encoding of the state space input patterns to form the rule antecedent terms but treats the rule consequents as real functions which are linear-in-the-parameters. These functions determine the control output when ANFIS is used to control the cart-pole problem, again, using a fixed number of rules. Learning consist of two stages: a forward pass to update the consequent parameters by recursive least squares estimation, and a backward pass to update the antecedent parameters by gradient descent down the error energy surface determined by the input fuzzification. Thus, the two tasks are decoupled and separate learning phases are used.

Although decoupling between the input and output tasks is less clear when using a continuous decoder, the continuous decoder approach is exemplified by the neurocontroller system of Anderson (1989) which uses reinforcement learning to update the control actions. It consists of two networks, the evaluation network and the action network, which are both two layer networks. Two layers, consisting of non-linear computing elements, are required to solve the cart-pole problem because both the desired evaluation function and the state space partitioning are non-linear. The evaluation network and the action network are analogous to the ACE and the ASE respectively. Both networks receive continuous-valued state inputs.

2.3. The Cart-Pole Problem

The starting point for the BSA formulation of reinforcement learning (Barto *et al*, 1983) is the 'boxes' adaptive problem solving system of Michie and Chambers (1968). The ASE adaptive element alone can emulate the performance of the boxes algorithm. As the boxes learning system forms the basis for the evolution of the present work, it will be described briefly here.

The problem posed by Michie and Chambers, to illustrate the boxes adaptive learning system, is the cart-pole problem which consists of a cart constrained to move along a one dimensional track, with a pole attached to it. This is illustrated in Figure 1. The movement of the pole is constrained within the vertical plane and is represented by the state variables θ and $\dot{\theta}$ signifying the angle of the pole from the vertical and the angular speed of the pole respectively. The movement of the cart is controlled by an impulse force (bang-bang control) in either direction and is represented by the state variables x and $\dot{x}$ which signify the distance from the origin (centre) of the track and the speed of the cart respectively. Thus, there are four state variables representing the whole motion of the cart-pole system. System parameters are given in appendix A which also specifies the physical system and computer simulation details.

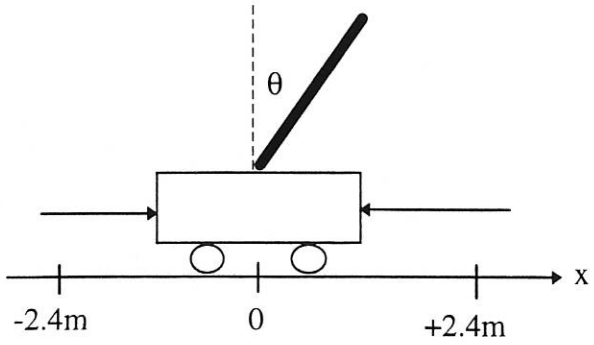

Figure 1. The cart-pole system. Motion is constrained within the vertical plane. See the body of the text for details.

Information from the physical system simulation is minimal and does not provide stimulus-response pairs consisting of inputs and desired outputs to be associated. Only the state vector and a coarse failure signal, reflecting the cart-pole status, are supplied to the control system. If the pole falls or the cart hits the track boundaries then a failure signal is sent to the controller and the cart-pole system is reset to its initial conditions to begin a new trial.

Under these conditions, the credit assignment problem becomes apparent; there are difficulties concerning the assignment of credit (blame) to individual control actions which, taken together, comprise the state space trajectory which leads to failure and thus the final failure signal (Barto *et al*, 1983). The boxes system (Michie and Chambers, 1968) partitions state space into 225 non-overlapping regions by quantising the state variables; note that this partitioning is fixed *ab initio* in both the boxes and the ASE / ACE systems. Each individual region is independent and is said to contain a local "demon" (Selfridge, 1959) which has to choose a control action of ±N Newtons whenever the state space trajectory enters the local region.

A global demon has overall control; its task is to decode the state vector, assign its trajectory to individual regions and distribute the failure signal to the local demons. Left/right force decisions are taken on the basis of the utility of these decisions calculated from past failure signals weighted by the time interval from box entry to failure for a given run. Thus, the expected lifetimes of a left or right decision determine the box output at any particular time and the temporally weighted effect of failure on the system is fed back to compute new left/right decision expected lifetimes. The full formulation of the boxes system is found in Michie and Chambers, (1968).

2.3. The ASE/ACE Implementation

2.3.1 The Associative Search Element (ASE)

The BSA implementation (Barto *et al*, 1983) uses the following quantisation of state space:

i) x: $-2.4m \le x < -0.8m \le x \le +0.8m < x \le +2.4m$,

ii) θ: $-12° \le \theta < -6° \le \theta < -1° \le \theta < +1 \le \theta < +6° \le \theta \le +12°$,

iii) $\dot{x}$: $\dot{x} < -0.5m/s \le \dot{x} \le +0.5m/s < \dot{x}$

iv) $\dot{\theta}$: $\dot{\theta} < -50°/s \leq \dot{\theta} \leq +50°/s < \dot{\theta}$

This collection of intervals results in a state space partition of 162 distinct regions. A decoder system (see Figure 2) assigns a unique output line to each state space region. This set of decoder outputs forms the unit input vector to the single ASE processing element. During processing, a state vector enters the decoder which switches on the appropriate input line to the ASE which subsequently issues a control action depending upon the current system state.

To avoid confusion between the original ASE /ACE notation and original ART notation, the ASE / ACE notation has been modified and consequently differs from that used in the original paper of Barto *et al*, (1983).

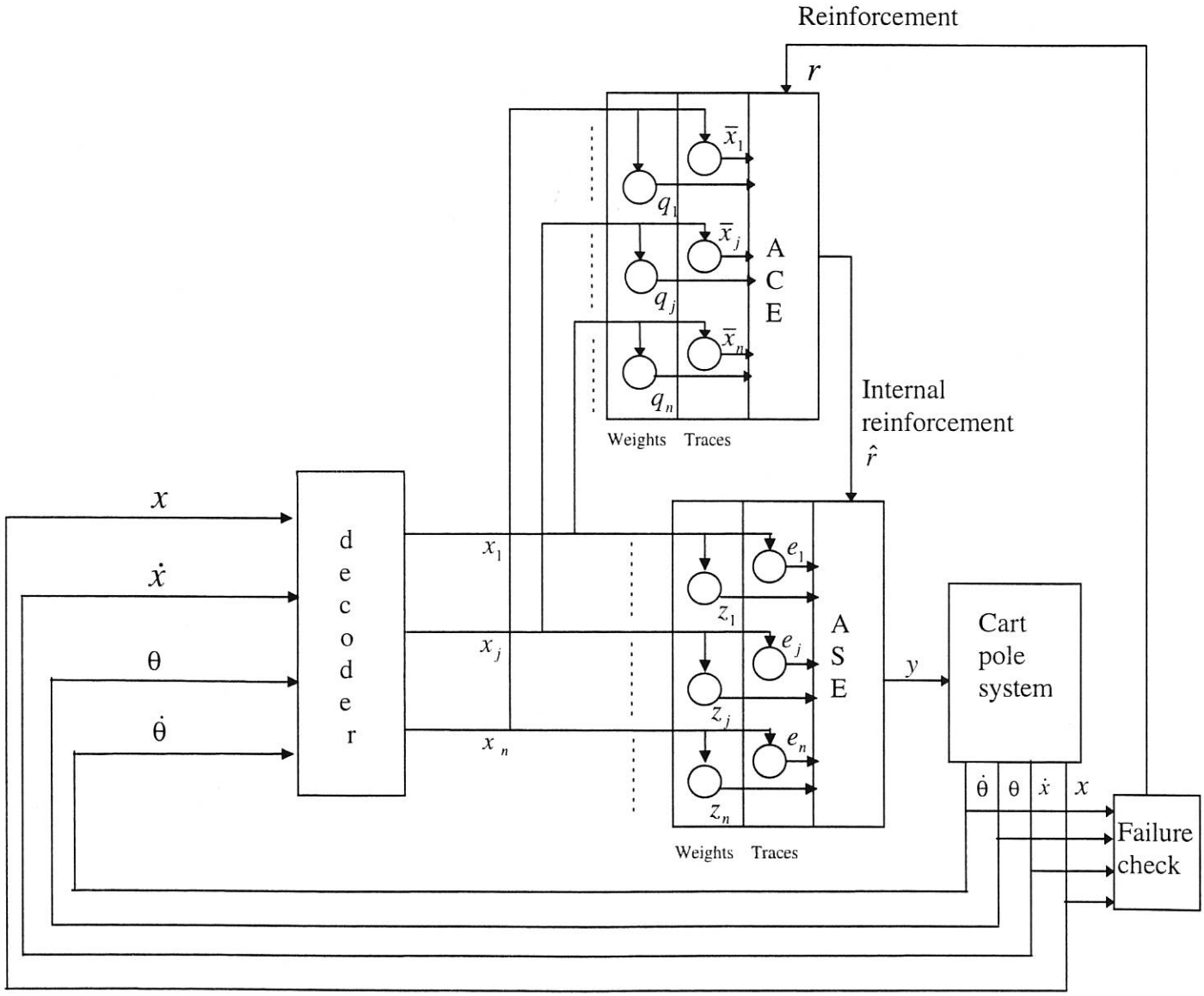

Figure 2. A neurocontroller based upon reinforcement learning. Both the original associative search element (ASE) and the adaptive critic element (ACE) of Barto *et al* (1983) have been retained. The independence of the decoder from the ASE / ACE subsystems makes it a focus for possible modifications (After Barto *et al.*, 1983).

The ASE control output is computed by

$$y(t) = f\left[\sum_{i=1}^{n} z_i(t)x_i(t) + \varepsilon(t)\right] \qquad (1)$$

where $y(t)$ is the output at time t, $z_i(t)$ is the scalar weight value of the i^{th} ASE input line at time t, $x_i(t)$ is the activation of the i^{th} ASE input line, $\varepsilon(t) \sim N(0,1)$ is Gaussian noise derived from a zero mean source with unit variance
and

$$f(x) = \begin{cases} +1 & for\ x \geq 0 \\ -1 & for\ x < 0 \end{cases}$$

gives the activation function of the ASE element which signifies the right and left control actions respectively.

The BSA implementation uses a standard basis of 162 unit vectors of 162 entries; when the i^{th} input line is active, the basis vector signifying the ASE input vector consists of all zero entries except for a "one" at the i^{th} entry. The decoder is a sub-system of the whole control system which lends itself to useful modification. This allows the properties of the controller to be modified whilst retaining the functionality of the ASE and ACE sub-units. Various methods of state space partitioning become possible (e.g. Lin and Kim, 1991) including self-organisation through experience as considered in this paper. Thus, the *a priori* partitioning of state space, as given in the original formulation, is a sufficient but not necessary condition for using the ASE / ACE system.

From equation (1) it can be seen that, at a given time, τ

$$y(\tau) = f\left[z_k(\tau) + \varepsilon(\tau)\right] \quad k \in \{1,\ldots,162\}$$

where k is the index of the input line. The weight, $w_i(\tau)$ signifies the direction in which the control force is applied at time, τ depending on the result when added to $\varepsilon(\tau)$ the random perturbation also at time, τ.

The ASE weight evolution equation, for the i^{th} input line is given by

$$z_i(t+1) = z_i(t) + a\hat{r}(t)e_i(t), \qquad (2)$$

where $\hat{r}(t)$ is the real valued reinforcement at time t, $e_i(t)$ is the eligibility at time t of input pathway i and a is the positive rate of change constant for z_i which determines the magnitude of change in z_i with respect to the reinforcement signal. The term 'reinforcement' has already been mentioned and, for the ASE unit operating alone, is given the value of 0 throughout a trial until failure occurs when it becomes equal to -1.

Eligibility is derived from the work of Klopf (Klopf, 1986, 1988) and represents the temporal weighting of the reinforcement signal in the derivation of the weight change. In a series of modifications to the Hebbian model (Hebb, 1949), Klopf suggests that, "instead of correlating approximately simultaneous pre- and post-synaptic signal levels, earlier pre-synaptic signal levels should be correlated with later post-synaptic signal levels." (Klopf, 1988). Klopf considers *changes* in levels to be more important but here we are concerned with the signal levels and delay effects. This is consistent with a solution of the credit assignment problem which requires temporally adjusted weight updates for distributing credit or blame to state space partitions traversed by an evolving state space trajectory. The eligibility update equation is given by:

$$e_i(t+1) = \delta e_i(t) + (1-\delta)\, y(t) x_i(t) \qquad (3)$$

where δ, $0 \le \delta < 1$ is a constant determining the eligibility trace decay rate.

This linear difference equation gives an exponentially decaying eligibility trace which maximally contributes to weight updates when the given input line is activated recently with respect to the reinforcement signal. Without stimulation via conjunction of pre- and post-synaptic activity reflected in equation (3), the eligibility trace passively decays. This is Hebbian learning (Hebb, 1949) with passive decay. The inclusion of the term *y(t)* ensures that information regarding the direction of the force is included in the weighting which reflects the expected lifetime and desirability of a particular control force. Consequently, actions which were made relatively long ago, with respect to eventual failure, merit little change to their expected lifetimes and, thus, exert little influence on the outcome.

2.3.2. The Adaptive Critic Element (ACE)

The ACE is similar in structure to the ASE (see Figure 2) and computes an expected or predicted reinforcement signal given the current state vector and external reinforcement from the system; the predicted reinforcement is continuous unlike the external reinforcement signal and allows learning throughout a trial. Thus, the combined ASE / ACE system is not a purely failure driven system. The prediction of expected reinforcement is given by

$$p(t) = \sum_{i}^{n} q_i(t) x_i(t) \qquad (4)$$

where $q_i(t)$ is the weight for the i^{th} input line and $x_i(t)$ is the input signal for that line as before.

The learning rule is given by

$$q_i(t+1) = q_i(t) + b\hat{r}(t)\bar{x}_i(t) \qquad (5)$$

where b, $b > 0$ is a constant which determines the rate of change of learning in q_i, $\hat{r}(t)$ is the predicted reinforcement and $\bar{x}_i(t)$ is a trace of the activity of the input variable x_i.

This trace, unlike the eligibility trace, does not take into account the control action chosen by the system for the region of state space. It is given by:

$$\bar{x}_i(t+1) = \lambda \bar{x}_i(t) + (1-\lambda)x_i(t) \qquad (6)$$

Where λ, $0 \leq \lambda < 1$ is a rate of change constant. Although similar in form to the eligibility trace, it provides a record of the activity of the input line x_i alone during the trial to determine whether or not the particular input line contributes to the prediction. With the present protocol of selecting a single input line, equation (4) becomes $p(\tau) = q_k(\tau)$ at time τ where the weight q_i reflects the prediction of failure for a given control action elicited by entering the region of state space coded for by input line k.

A distributed version of equation (4) might also be used where multiple input lines, $x_i(t)$ are activated to varying degrees, in the range zero to one, and thus weight the prediction contributions to give a final prediction of reinforcement; this possibility is mentioned in Barto *et al* (1983).

The predicted reinforcement is given by

$$\hat{r}(t) = r(t) + \gamma p(t) - p(t-1) \qquad (7)$$

where r(t) is the external reinforcement, $r(t) \in \{0,-1\}$, and γ, $0 < \gamma \leq 1$, is a discounting factor.

The discounting factor is required to prevent the reinforcement from becoming self-sustaining. To see this, consider $\gamma = 1$ and $p(t) = p(t-1)$ at some time, t. If failure has not yet occurred, equation (7) gives

$$\begin{aligned}\hat{r}(t) &= 0 + p(t) - p(t-1) \\ &= 0\end{aligned}$$

Now, from equation (5),

$$q_i(t+1) = q_i(t) + b\hat{r}(t)\bar{x}_i(t) = q_i(t) \quad : \hat{r}(t) = 0$$

for some node, i, so that

$$p(t+1) = q_i(t+1) = q_i(t) = p(t)$$

if node i is chosen again. Thus, the prediction for a particular node becomes self-sustaining.

When r(t)=0, (failure has not yet occurred) a smaller prediction of failure, p(t)>p(t-1), signifying a transition from a region of higher expected failure to a region of lower expected failure, constitutes a positive reinforcement.

When r(t)=-1 (failure), p(t)=0 (no present prediction) and equation (7) becomes

$$\hat{r}(\tau) = -1 - p(\tau - 1)$$

Thus, the degree of prediction of failure is taken into account and fully predicted failure is not penalised.

2.4. The Decoder Subsystem

In the original BSA implementation, the decoder is specified by using a fixed mapping between the partitioned state space and the input lines. Another decoder scheme (Lin and Kim, 1991) uses the cerebellar model articulation controller (CMAC) of Albus (Albus, 1975a, 1975b, 1979; Tolle and Ersu, 1992) with a fixed number of memory locations and an efficient mapping which maps only states which are used, to locations in the CMAC controller. The distribution of state space information across the locations leads to a degree of overlap and, consequently, some ability to generalise about regions of state space not yet traversed. The large state space is mapped to smaller storage space using the state variables as an address key (Lin and Kim, 1991) for the decoder. This compression avoids allocation of storage for large regions of state space which are not used.

The decoder provides a sub-unit replete with possibilities for modification. Decoder modules can be designed which implement various mappings between state space and the ASE / ACE controller sub-systems. If the decoder co-domain consists of independent input lines as in the original BSA implementation then the possibility of increasing network size by exploring state space presents itself. The addition of new input lines, representing newly traversed areas of state space, will not conflict with the previously established input lines to the ASE / ACE and their corresponding weight and trace values. Although the input lines are independent, the state space regions represented by these lines may overlap and temporarily disrupt the mapping; this phenomenon is considered in section 4.

3 An ART-based decoder

From equations (1) and (4), for some k,

$$x_i(t) = \begin{cases} 1 & for \quad i = k \\ 0 & for \quad \forall i \neq k, \end{cases}$$

means that y(t) and p(t) depend upon one input line only. This decoupling of the x_i allows the addition of new input lines without disruption of the established output and prediction values, for the existing lines, which would occur if more than one input line contributed to the calculation. Thus, decoders which dynamically partition the state space, using whatever method, can be easily linked to the ASE / ACE sub-systems provided that the coded state space regions have unique input lines. This method is highly dependent on experience and is flexible in that new regions of state space encountered by different initial conditions or disturbances can be accounted for by allocating new storage areas (nodes) which contain the traces and expected lifetime / prediction values for the newly encountered state space region.

A distributed representation of *y(t)* and *p(t)* of equations (1) and(4) respectively is possible if input line conflicts are avoided by allocating input line activity according to the degree of node membership (e.g. Zhang and Grant, 1992). Here, the single activated input line convention of Barto *et al*, (1983) is adopted for compatibility between the original ASE / ACE formulation and winner-takes-all dynamics.

3.2 Fuzzy ART

One of the ART family of networks, fuzzy ART (Carpenter *et al*, 1991) presents itself as a candidate self-organising state space decoder through its ability to perform unsupervised clustering. A brief description of the fuzzy ART architecture and operation is included here to provide a basis for the explanation of the proposed modifications.

3.2.1 Basic Architecture

Each fuzzy ART module consists of three fields, or layers, of nodes: an input field, a matching field and a choice field. A schematic outline of a fuzzy ART module is shown in Figure 3. The input field, F_0 stores the current input vector and transmits it to the matching field, F_1 which also receives top-down input from the choice field F_2; this latter field representing the active category assignment of the input data.

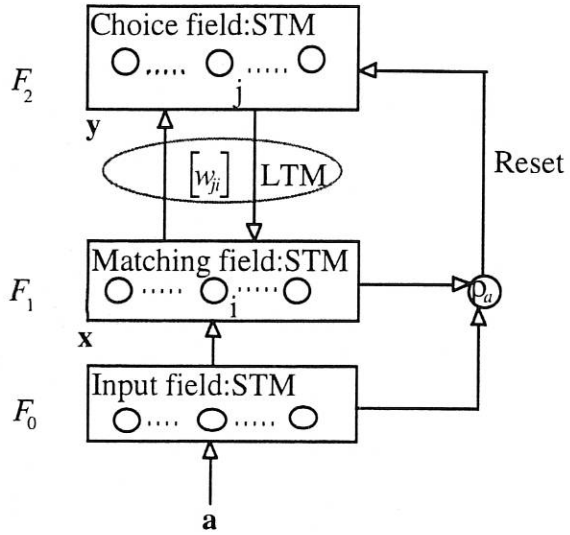

Figure 3. A Fuzzy ART module. Fuzzy ART dynamics form a basis for a self-organising state space decoder system (after Carpenter *et al*, 1991).

The F_0 activity vector is denoted by $\mathbf{I} = (I_1,\ldots,I_M)$, $I_i \in [0,1]$, a subset of the real line, , $\forall i = 1,\ldots,M$. The F_1 and F_2 activity vectors are denoted by $\mathbf{x} = (x_1,\ldots x_M)$ and, $\mathbf{y} = (y,\ldots,y_N)$ respectively. Each F_2 node represents a class or category of inputs grouped together around an exemplar or prototype generated during the self-organising activity of the fuzzy ART module. Furthermore, each F_2 category node, j has its own set of adaptive weights stored in the form of a vector $\mathbf{w}_j = (w_{j1}, w_{j2},\ldots, w_{jM})$, $\forall j = 1,\ldots,N$.

These weights represent the long term memory (LTM) traces which evolve during network operation. The initial weight vector values are given by:
$w_{ji}(0) = 1$, $\forall j = 1,\ldots,N$, $\forall i = 1,\ldots,M$.

With no categories being allocated to F_2 nodes at this stage, the nodes are said to be *uncommitted.* Once a category node is chosen to represent a category it then becomes *committed.*

3.2.2 Choice field activity

The choice field (F_2) nodes operate with winner-takes-all dynamics modelled by the F_2 output function (choice function)

$$T_j(\mathbf{I}) = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \quad \forall \mathbf{I} \in [0,1]^M, \tag{8}$$

where $\mathbf{I}$ is the given input vector, $\mathbf{w}_j$ is the j^{th} F_2 node weight vector, α is the choice parameter where $\alpha \cong 0$, $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min_i(p_i, q_i)$, is the fuzzy AND operator (Zadeh, 1965), and the L^1 norm $|.|$ is defined by $|\mathbf{p}| = \sum_{i=1}^{M} |p_i|$.

The overall F_2 winner, node J, is selected by $T_J = \max_j \{T_j : j = 1,\ldots,N\}$ to represent a category choice for a given input vector $\mathbf{I}$.

The choice parameter, α breaks the deadlock between competing nodes when $\mathbf{w}_j$ and $\mathbf{w}_k$ are both *fuzzy subsets* of $\mathbf{I}$, by selecting the node j such that $|\mathbf{w}_j| > |\mathbf{w}_k|$.
This is because $T_j(\mathbf{I})$ is monotonically increasing so that, $|\mathbf{I} \wedge \mathbf{w}_j| = |\mathbf{w}_j|$ giving

$T_j(\mathbf{I}) = \dfrac{|\mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}$. Thus for $|\mathbf{w}_j| > |\mathbf{w}_k|$, $T_j(\mathbf{I}) > T_k(\mathbf{I})$.

In the case that $T_j = T_k$ for some $j,k \le N$, such that $T_j, T_k > T_l \;\; \forall l \neq j,k$
the node with the lowest index is chosen.

3.2.3 Matching field activity

The F_1 layer activity is governed both by bottom-up F_0 layer and top-down F_2 layer activity according to

$$\mathbf{x} = \begin{cases} \mathbf{I} & \text{if } F_2 \text{ is inactive} \\ \mathbf{I} \wedge \mathbf{w}_J & \text{if the } J^{th} F_2 \text{ node is active.} \end{cases}$$

If node J is active, $\mathbf{w}_J$ represents an expected pattern or template fed down from F_2; this template is combined with the input vector present across F_0 to produce a resultant vector. The ratio of the magnitude of the resultant vector to the magnitude of the input vector gives the degree of match. This ratio, or match function, is denoted by $\dfrac{|\mathbf{I} \wedge \mathbf{w}|}{|\mathbf{I}|}$ and must fulfil the criterion

$$\frac{|\mathbf{I} \wedge \mathbf{w}|}{|\mathbf{I}|} \ge \rho, \qquad (9).$$

to ensure that the input vector belongs to the chosen category. This state is known as *resonance* and allows learning to occur in the relevant section of the LTM weight matrix. The parameter ρ is the *vigilance* parameter, where $\rho \in [0,1]$.
The situation where $\dfrac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} < \rho$, known as *mismatch,* causes the system to reset and inhibits the winning node ($T_J = 0$) which is, thus, unable to re-enter the competition from which a new winner is selected. The cycle continues with multiple re-presentations of the input vector until the input is either assigned to an existing category or becomes the exemplar for a newly created category.

3.2.4 Learning

Following a successful search, LTM changes are made according to

$$\mathbf{w}_J^{(new)} = \beta(\mathbf{I} \wedge \mathbf{w}_J^{(old)}) + (1-\beta)\mathbf{w}_J^{(old)} \qquad (10)$$

for the winning F_2 node, J. These changes correspond to the notion of learning.

The learning rate parameter, β, with $0 \leq \beta \leq 1$ ensures that the new weight vector $\mathbf{w}_J$ is a convex combination of the resultant vector across F_1 and the F_2 layer expectation template. For $\beta = 1$, known as *Fast-Commit-Fast-Recode (FCFR)*, F_1 resultant vectors directly replace the present category exemplars.

An option, *Fast-Commit-Slow-Recode(FCSR)*, allows for initial fast learning prior to the convex combination learning rule of equation (10) by setting $\beta = 1$ for uncommitted nodes only. Thus, $\mathbf{w}_J^{(new)} = \mathbf{I}$ initially.

3.2.5 Complement coding

According to Carpenter *et al,* (1991a, 1991b, 1992) normalisation of the input vectors is required to prevent category proliferation. In Carpenter *et al,* (1991) it is proved geometrically that, without complement coding, the monotonically decreasing weight components would eventually result in many categories clustering near to the origin with others being created to replace them. For example, on the real line, when all categories to the left of an input value are inhibited, the first category to the right will be selected as any categories further to the right will result in a smaller activation value for the function $T(I)$ (Marriott and Harrison, 1994). Furthermore, the condition of equation (9) is always fulfilled as $I < w_J$ gives $\dfrac{|\mathbf{I} \wedge \mathbf{w}_J|}{|\mathbf{I}|} = \dfrac{I}{I} = 1 \geq \rho$

Normalisation is represented by $|\mathbf{I}| \equiv \gamma, \quad \forall \mathbf{I} \in [0,1]^M$ for some $\gamma > 0$. To achieve this for arbitrary $\mathbf{I} \in [0,1]^M$ take $\mathbf{I} = (\mathbf{a}, \mathbf{a}^c) \in [0,1]^{2M}$ where $\mathbf{a} \in [0,1]^M$ is the original input and $\mathbf{a}^c = \mathbf{1} - \mathbf{a}$ where $\mathbf{1} = (1,1,\ldots,1)$, and, $|\mathbf{1}| = M$.
Thus, the new F_0 layer input vector , $\mathbf{I}$ is complement coded and of dimension 2M with $|\mathbf{I}| = \gamma = M, \quad \forall \mathbf{I} \in [0,1]^{2M}$.

3.3 EUCART

To solve a non-linear control problem using a neurocontroller, a method of representing state space is required which allows the association of control actions with distinct state space regions; the regions may overlap but they must be distinctly identifiable so that unique outputs may be assigned to them. A convenient set of methods of representing state space that is compatible with the incremental learning paradigm involves Euclidean clustering (e.g. Kohonen, 1989). Individual states are assigned to a cluster and, in some cases, new clusters may be added as required. Euclidean clustering methods provide a convenient way of assigning cluster membership by comparing the distance between an input vector and various categories stored by the system. Category assignment based upon the Euclidean distance between inputs and category centres, or prototypes, results in a partitioning of state space as shown in Figure 4 if winner-takes-all dynamics are used. Inputs are assigned to the category represented by the nearest category centre; this results in a unique assignment for each input unless two or more category centres are equidistant from an input vector, which is unlikely. Category centres may be represented by nodes within the neurocontroller. Following competition between category nodes, either an overall

winner or a selection of active nodes may be chosen to compute a control output. In the latter case, a method of weighting the nodal contributions is required; the weighting is usually a function of the category node activation.

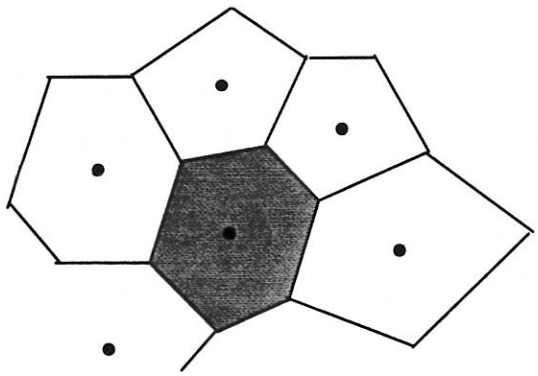

Figure 4. Using Euclidean clustering with winner-takes-all dynamics results in a state space partitioning that consists of irregular convex regions. The regions are comprise intersecting hyperplanes that represent the decision surface between neighbouring category centres.

Here, a novel Euclidean clustering method, inspired by some of the attractive properties of fuzzy ART (Carpenter *et al.*, 1991), is presented that overcomes the problem of *category drift* (Moore, 1989) and allows incremental learning of state space information without supervision. This method is compatible with the BSA formulation of reinforcement learning and is implemented as a state space decoder that replaces the fixed structure of Barto *et al*, (1983).

3.3.1 Fuzzy and Euclidean clustering

The fuzzy ART system has many desirable properties of which a subset can be abstracted for the purpose of designing a state space decoder. Framing this subset in Euclidean terms serves as a springboard to further developments in decoding schemes. As will be discussed in this section, emulating one particular aspect of fuzzy ART operation provides a first attempt at a solution to the problem of Euclidean category drift (Moore, 1989). Other properties framed in Euclidean terms lose some of the characteristics which make fuzzy ART particularly good at unsupervised learning. However, the Euclidean network presented here is not designed to function as a classifier in the sense that the fuzzy ART system is and further development would produce a closer functional relationship between the fuzzy and Euclidean clustering schemes if required. The object is not simply to have a Euclidean form of fuzzy ART, if that is at all possible; fundamental differences in Euclidean and fuzzy metrics restrict operational correspondences in networks to functional analogies.

3.3.2 The EUCART System

EUCART is a Euclidean self-organising state-space decoder based loosely on Fuzzy ART (Carpenter and Grossberg, 1991), hence the name, from EUClidean ART. Its purpose is autonomously to structure state space so that the ASE / ACE sub-units may associate control actions with individual state space regions through reinforcement learning. The main property of fuzzy ART incorporated into EUCART is the *category*

growth property which prevents category templates or prototypes from wandering (Figure 5). The category growth property allows the new category to incorporate the region of state space encompassed previously by the category by expanding outwards towards the input vector up to a maximum possible extent. The existing members of the cluster will always remain within the category.

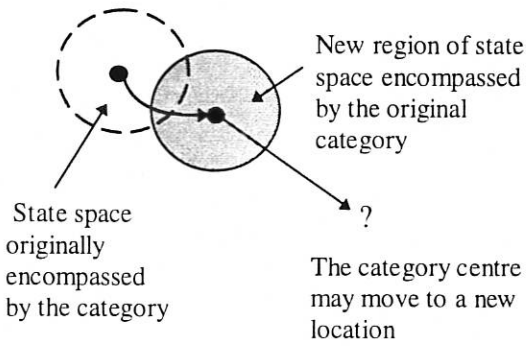

Figure 5. A schematic illustration of the phenomenon of category wandering. As a given category centre is updated by a stream of inputs assigned to the category that it represents, the location of the category centre moves and encompasses a new region of state space. Consequently states that belonged to a particular category may not belong to it any longer.

Category drift may cause degradation of performance in control applications where information is lost through the movement of categories. For example, it is possible that some states have control actions associated with them when assigned to a particular category, but lose these associations when the category moves. Consequently, either no control actions are available for such "displaced" states, or , different control actions are assigned following the category movement. For winner-takes-all dynamics, gains in using the category growth property may be offset by the plasticity of category assignment where states are reassigned to nearer category centres during learning. Using the category growth property prevents the case where states lose any associated output altogether by ensuring that states always remain members of the category extent of one or more category nodes once they come within the category boundaries; where there is multiple membership, an overall winner may be chosen. The problem of category dissociation is shown in Figure 6.

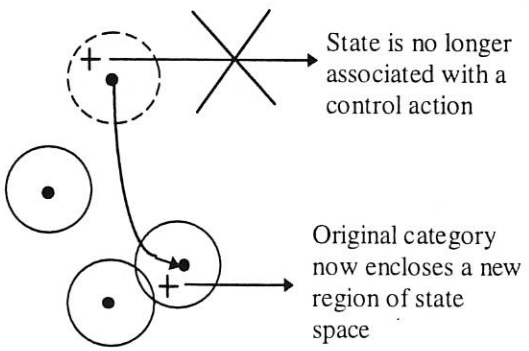

Figure 6. Category wandering can result in dissociation of states from control actions that are learned responses to these states. Control information is lost or disrupted depending upon the degree of category displacement.

Where a distributed representation is used, the phenomenon of category wandering presents more of a problem because once a state "informs" a control output associated with a category node, even if that particular category node is not the overall winner next time, it still contributes proportionately to the output as states may belong to one or more categories. Thus, when states are dissociated from categories that they have informed (i.e. they have modified associated category information) when these states or regions of state space are reactivated, information relating to the control output is lost. This is illustrated in Figure 7.

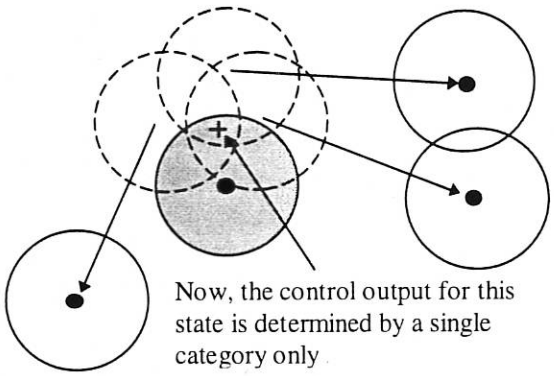

Figure 7. The effect of category wandering when a distributed representation is used. Categories that previously contributed to the control output no longer have any effect. The category growth property ensures that, for a distributed representation, once a region of state space contributes to a category and its associated control action, it will always continue to do so.

The EUCART system retains fields F_0, F_1 and F_2 but differs in the dynamic functioning of the latter two, especially in the case of the matching field, F_1. Unlike Fuzzy ART, EUCART does not use complement coding. The input is given by $\mathbf{I} = (I_1,\ldots,I_M)$,. a subset of the real line, , $\forall i = 1,\ldots,M$.

Note that the dropping of complement coding removes the unit normalisation restriction. That is, inputs do not have to remain within the *unit* hypercube. However, EUCART inputs have to be within a fixed input space (hypercuboid) of specified dimensions. In this paper we confine the inputs within the unit hypercube, $[0,1]^M$ for convenience.

Each F_2 node represents a class or category of inputs and operates in winner-takes-all mode as before.

Associated with each F_2 node j is a set of adaptive weights
$\mathbf{w}_j = (w_{j1}, w_{j2},\ldots,w_{j2M}), \quad \forall j = 1,\ldots,N.$
These weights store the network LTM traces in a form allowing emulation of a fuzzy ART property which prevents category wandering. Two vectors of length M are stored as a single vector representing the minimum and maximum extents of category growth in elements 1 to M and M+1 to 2M respectively. Before discussing this mechanism it instructive to look at both Euclidean and fuzzy categorisation in more detail.

3.3.3 Category drift: a fuzzy ART approach

we have seen that Euclidean clustering suffers from a phenomenon known as 'category drift' which results from the updating and subsequent movement of category centres in the classification system representation of input space. In some cases, categories drift quite dramatically and even re-occupy previous class centre positions. Monotonic changes can help to rectify this problem but can introduce problems of a different type (Moore, 1989). Fuzzy ART gets around this problem by using complement coding. Complement coding allows a category to grow by incorporating previously enclosed space within the new category extent. The category growth property of fuzzy ART ensures that categories do not drift and occupy different areas of state space.

The weight vector of a fuzzy ART category is given by

$$\mathbf{w}_j = (\mathbf{u}_j, \mathbf{v}_j^c) \qquad (11)$$

where $\mathbf{u}_j$ and $\mathbf{v}_j^c$ are non-complement coded and complement coded respectively. Note that $\mathbf{v}_j^c$ is not necessarily the complement coded form of $\mathbf{u}_j$. When the new category is first created, $\mathbf{v}_j^c$ is the complement coded form of $\mathbf{u}_j$ but, during operation, sometimes only $\mathbf{u}_j$ is replaced by the new input following the fuzzy AND operation and at other times, $\mathbf{v}_j^c$ is replaced depending upon the new input vector.

As illustrated in Figure 8, $\mathbf{u}_j$ and $\mathbf{v}_j^c$, for a two dimensional system, represent the extent of the current category if the size of the rectangle R_j is defined as

$$|R_j| = |\mathbf{v}_j - \mathbf{u}_j| \qquad (12)$$

If a new input $\mathbf{a}$ is used to update the J^{th} winning category of extent $|R_J|$, the updating operation, signified by $\oplus$, gives

$$|R_J \oplus \mathbf{a}| = |(\mathbf{a} \vee \mathbf{v}_J) - (\mathbf{a} \wedge \mathbf{u}_J)|. \qquad (13)$$

This comes from applying the FCFR learning equation (β=1.0), using input $\mathbf{a}$, to the weight vector $\mathbf{w}_J$.

If the new category $|R_J \oplus \mathbf{a}|$ is too large then fuzzy ART resets and searches for a new category. It can be shown (Carpenter *et al*, 1991) that, for an M dimensional input space,

$$|R_J \oplus \mathbf{a}| \leq M(1-\rho) \qquad (14)$$

Thus, the growth of a particular category is limited by the vigilance parameter as would be expected.

An important point to note is that the weight values do not signify the centre of a category in the normal sense of clustering procedures; they signify the 'extent' values which allow the growth of a category to include previously encompassed values of weight space. Categories do not move but grow until a specified upper limit of category size is reached.

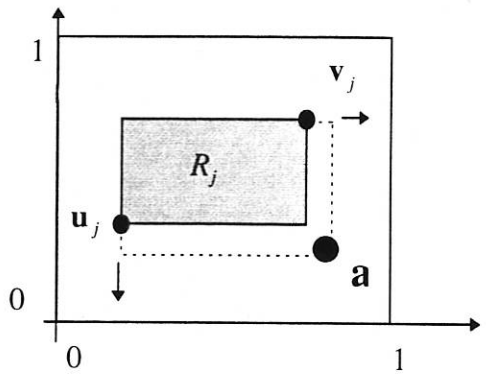

Figure 8. Fuzzy ART clustering and category growth illustrating the category growth property. (After Carpenter and Grossberg (1991)). Categories are represented by 'extent markers' and not centres; the latter are not meaningful when using the L^1 norm which is more suited to fuzzy operations than is the Euclidean norm.

EUCART weight values are given by $\mathbf{w}_j^E = (\mathbf{u}_j^E, \mathbf{v}_j^E)$, where $\mathbf{u}_j^E$ and $\mathbf{v}_j^E$ are the minimum and maximum category extent markers with components given by $u_{jk}^E(t+1) = \min(u_{jk}^E(t), a_k)$, and $v_j^k(t+1) = \max(v_j^k(t), a_k)$ respectively. These 'poles' moving in 'opposite' directions in M-dimensional hyperspace delimit hyperspace categories whose extent is given by

$$e_j = \left\| \mathbf{v}_j^E - \mathbf{u}_j^E \right\| \qquad (15),$$

where $\| . \|$ is the Euclidean or L^2 norm (Euclidean distance between two vectors); see Figure 9. Analogously to equation (14) the category growth criterion for normalised Euclidean space becomes

$$e_j \leq \frac{\sqrt{M}}{2}(1 - \rho_E) \qquad (16)$$

where $\sqrt{M}$ is the maximum possible Euclidean distance between points in $[0,1]^M$ and ρ_E is the vigilance parameter of EUCART. Analogous to fuzzy ART, for high vigilance, i.e. $\rho_E \to 1$, the resulting categories are very small and for low vigilance, i.e. $\rho_E \to 0$ they are very large.

The centre of a category is given by

$$\mathbf{c}_j = \frac{1}{2}(\mathbf{u}_j^E + \mathbf{v}_j^E) \qquad (17)$$

and, as with fuzzy ART, does not reflect the centre of mass (centroid)of the category which may be desired for certain applications. The centre of mass of a category may be computed incrementally during learning as required. All input vectors that contribute to a category will continue to belong to that category throughout the learning process.

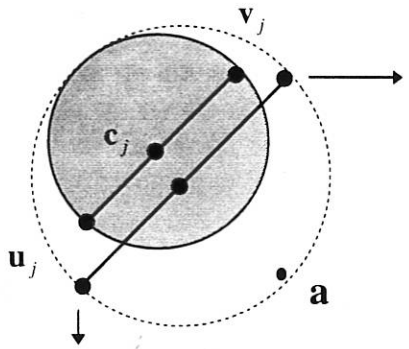

Figure 9. EUCART clustering using the category extent markers analogous to those of Fuzzy ART. When a new input extends the category boundaries, the category centre will also change. However, the subset of input space encompassed by the previous category remains within the new category.

An input, $\mathbf{I}$ is said to be a member of the i^{th} category if

$$\|\mathbf{c}_i - \mathbf{I}\| \leq \frac{\sqrt{M}}{2}(1-\rho_E) \qquad (18)$$

which forms the EUCART match criterion. This form of match criterion, unlike equation (9) of fuzzy ART does not take into account the absolute magnitude of the input vector. In the context of state space partitioning this is not particularly important as the main focus of interest is upon absolute distances from a state space exemplar. In pattern recognition tasks, however, the absolute magnitude must be taken into account so that patterns are matched according to the degree of correlation between them. For example, an input may be closer to a signal category in terms of absolute magnitude but have a much smaller correlation in terms of vector direction, i.e. dot product. Thus, a category with a smaller exemplar vector magnitude but nearer in terms of angle may well be the desired category. This important point is reflected in fuzzy ART by the dual choice and matching functions and search mechanism. Note that choosing a winner in terms of distance alone is not equivalent to finding the largest net input by using the dot product, unless the exemplar weights are normalised.

3.3.4 The fuzzy choice and matching functions revisited

The fuzzy ART choice function of equation (8) approximates to

$$T_j(\mathbf{I}) \approx \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{w}_j|} \qquad \text{with } \alpha \to 0.$$

This measures the extent to which $\mathbf{w}_j$ is a fuzzy subset of $\mathbf{I}$ (Zadeh, 1965; Kosko, 1992).

Where $\mathbf{w}_j$ is a fuzzy subset of $\mathbf{I}$, $\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{w}_j|} = 1$ and equation (8) becomes

$$T_j(\mathbf{I}) = T_j(\mathbf{w}_j) = \frac{|\mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \qquad (19)$$

with $T_j(\mathbf{I}) \approx 1$.

$|\mathbf{w}_j|$ can be maximised up to a maximum value at $\mathbf{w}_j = \mathbf{I}$ to give the highest choice function for fuzzy subsets of $\mathbf{I}$ through the monotonic increasing property (M.I.P.) of equation (19); this property is proved informally in appendix B. The fuzzy ART match function of equation (9) has a value of

$$\frac{|\mathbf{I} \wedge \mathbf{w}_j|}{|\mathbf{I}|} = \frac{|\mathbf{w}_j|}{|\mathbf{I}|} \qquad (20)$$

for $\mathbf{w}_j$ a fuzzy subset of $\mathbf{I}$.

So, for this special case, given the set of fuzzy subsets of $\mathbf{I}$, denoted by $\Omega_{\mathbf{I}}$,

for $\mathbf{w}_j, \mathbf{w}_k \in \Omega_{\mathbf{I}}$

$T_k(\mathbf{I}) \geq T_j(\mathbf{I}), \quad \Rightarrow \quad |\mathbf{w}_k| \geq |\mathbf{w}_j|, \quad \forall j \neq k$

by the M. I. P.

Now, for $|\mathbf{w}_k| = \max_j \{|\mathbf{w}_j|\}$, if reset occurs, no more matches can be found since, from equation (20)

$|\mathbf{w}_j| \leq |\mathbf{w}_k| \Rightarrow \frac{|\mathbf{w}_j|}{|\mathbf{I}|} \leq \frac{|\mathbf{w}_k|}{|\mathbf{I}|}$, and,

$$\frac{|\mathbf{w}_k|}{|\mathbf{I}|} < \rho \Rightarrow \frac{|\mathbf{w}_j|}{|\mathbf{I}|} < \rho, \quad \forall \mathbf{w}_j \in \Omega_{\mathbf{I}} \text{ for some } \mathbf{w}_k \in \Omega_{\mathbf{I}}.$$

Thus, no further search for fuzzy subsets is required. Other searches will follow for cases when $\mathbf{w}_j \notin \Omega_{\mathbf{I}}$.

If EUCART is given the choice function

$T_j^E(\mathbf{I}) = 1 - \dfrac{\|\mathbf{c}_j - \mathbf{I}\|}{\sqrt{M}}$, where $\mathbf{c}_j = \dfrac{1}{2}\left(\mathbf{u}_j^E + \mathbf{v}_j^E\right)$ as before, for a normalised space, $[0,1]^M$ then $T_j^E(\mathbf{I}) \in [0,1]$ and a match criterion based purely upon absolute distance

$$\|\mathbf{c}_j - \mathbf{I}\| \le \frac{\sqrt{M}}{2}(1 - \rho_E)$$

removes the necessity of a further search if the criterion is not fulfilled. In other words,

$$T_k^E(\mathbf{I}) \ge T_j^E(\mathbf{I}) \quad \Rightarrow \quad \|\mathbf{c}_j - \mathbf{I}\| \ge \|\mathbf{c}_k - \mathbf{I}\|, \quad \forall j \ne k$$

and
failing the match criterion gives

$$\|\mathbf{c}_j - \mathbf{I}\| \ge \|\mathbf{c}_k - \mathbf{I}\| > \frac{\sqrt{M}}{2}(1 - \rho_E)$$

which implies that

$$\|\mathbf{c}_j - \mathbf{I}\| > \frac{\sqrt{M}}{2}(1 - \rho_E), \quad \forall j$$

Thus, the search for a new winner is not required as no better match, in the sense of Euclidean distance, can be found. The simplified choice and match functions of EUCART do not take the magnitude of the input vector into account and so remove the need for a more complex search pattern. The more complex search is required to find a new input with roughly the same relative spatial pattern regardless of the absolute magnitude. These simplified dynamics suffice in the present control context because stored patterns reflect state values and correlation of an input with its canonical or exemplar state is based purely upon absolute distances. As mentioned above, more sophisticated pattern clustering, using a Euclidean metric, requires some form of angle or dot product measure to assess input correlation with stored exemplars to improve clustering properties. This correlation measure allows matching independently of signal magnitude, For example, in the clustering of visual data, it is desirable that patterns can be clustered within a sample space of varying background illumination, with respect to relative reflectance patterns.

3.3.5 EUCART Learning

Learning in EUCART is analogous to learning in fuzzy ART and LTM changes are made according to

$$\mathbf{u}_J^{(Enew)} = \beta_E(\mathbf{I} \wedge \mathbf{u}_J^{E(old)}) + (1-\beta_E)\mathbf{u}_J^{E(old)} \qquad (21a)$$

and,

$$\mathbf{v}_J^{E(new)} = \beta_E(\mathbf{I} \wedge \mathbf{v}_J^{E(old)}) + (1-\beta_E)\mathbf{v}_J^{E(old)} \qquad (21b)$$

where $\wedge$ and $\vee$ are the fuzzy AND and OR operators respectively (Zadeh, 1965). Equations (21a) and (21b) are used to find the new min and max category extent markers respectively provided that the category growth criterion of equation (16) is not violated by the updated extent markers; if this criterion is violated then the update is not carried out. The *Fast-Commit-Fast-Recode* and *Fast-Commit-Slow-Recode* options are retained in EUCART.

4. EUCART and the Cart-Pole Problem: some results

Section 4 presents simulation results showing the application of EUCART to the solution of a non-linear control problem using reinforcement learning. First, EUCART is applied in the form discussed in section 3. Second, a modified version of EUCART, EUCART with nearest neighbour priming, is proposed in an attempt to improve performance further. The results are compared to those of the original BSA implementation and some properties of the EUCART-based neurocontroller are discussed.

4.1 The EUCART Decoder Implementation

This section describes the simulations carried out using the EUCART state space decoder in place of the fixed state space decoder of Barto *et al.* (1983). The first objective was to see if the idea of decoupling the state space representation task and the control action learning task was tenable. If so, the original BSA reinforcement learning implementation could be retained and variant neurocontroller architectures could be developed through modifications to the decoder. The second objective was to develop a decoder which did not require *a priori* state space structuring and could organise state space information autonomously through experience. The achievement of the second objective is a sufficient condition for achievement of the first and indicates the possibility of other decoder architectures.

4.1.1 Simulations and Discussion

Simulations, following the method of Barto *et al* (1983) comprising 10 runs of 600 trials each, were carried out. As in the BSA implementation, the state vector was reset to $x = \dot{x} = \theta = \dot{\theta} = 0$ after each trial. The simulation conditions and parameters were similar to those in the BSA implementation except for a few minor changes necessitated by the new approach. First, runs were not terminated when the trial of a particular run first reached the ceiling of 500,000 time steps of 0.02 seconds (approximately 2.8 hours of simulated time). Learning was still occurring in some cases and the system had to reach the ceiling value a large number of times consecutively to indicate convergence. Second, the learning parameter, a was set to 1,000 in the BSA implementation to establish control actions quickly. In the present implementation, because the state space partitioning is not fixed, learning needs to remain plastic to prevent premature establishment of control actions. Hence a was set to 0.8.

The parameters used in the EUCART decoder were a=0.00001, b=0.9 and ρ=0.8. The FCSR option was used.

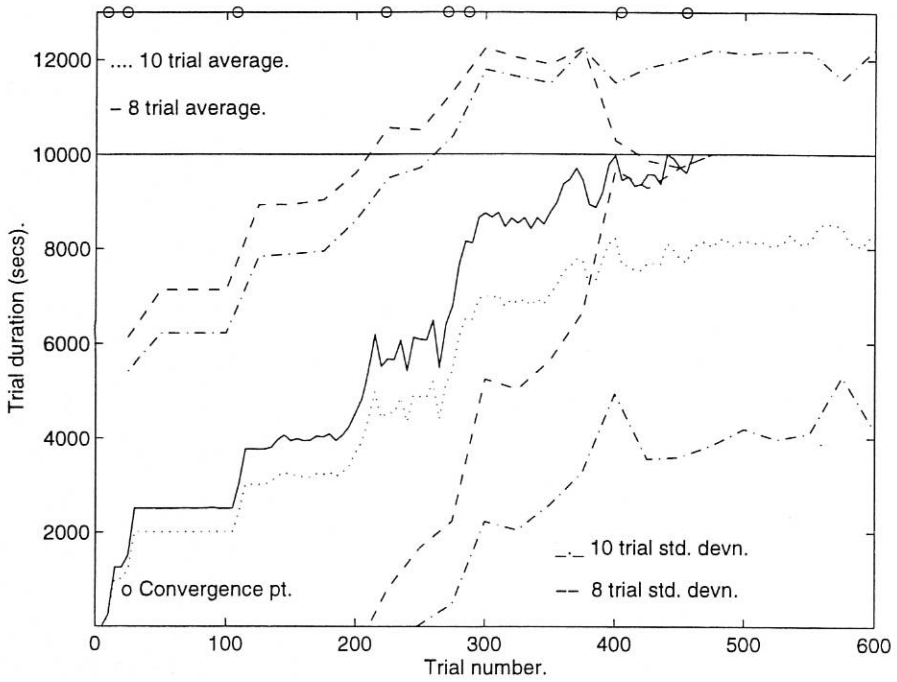

Figure 10. Simulation results showing the performance of the ASE / ACE system with the EUCART state space decoder. The trials were averaged over ten or eight runs as described in the text.

Figure 10 shows the results of 10 runs and a subset of 8 runs. The subset was required for clarity as 8 of the 10 runs converged to the ceiling value of 10,000 seconds (500,000 time steps) within the 600 trial limit; The remaining two runs converged at about 1500 trials and 1200 trials respectively. The solid curve shows the average of the 8 runs which converged during the trial limit. The dotted curve shows the average with the remaining two runs added to the ensembles for each trial. As with the BSA study, a single point is plotted to indicate the average of each bin of 5 consecutive trial (ensemble) averages. The remaining curves show 1 standard deviation either side of the respective means, i.e. the dashed curve is associated with the solid curve and the chained curve is associated with the dotted curve. These are calculated at 25 trial intervals on the original ensemble values (not on the five trial bins). Although the sample size is small, standard deviation is used to indicate spread, since maximum and minimum values are dominated by the trial which converges first. The circles at the top of the graph indicate at which trial the members of the 8 run subset converged.

In the original BSA implementation the simulation results show that convergence towards a solution of the cart-pole problem occurs mostly within 100 trials. The present implementation requires more trials than this on the whole, but does eventually solve the problem. Here, learning is incremental, and is required to be more plastic; consequently, learning is slower to allow for adjustments in the state space representation. With rapid learning of control actions, changes in state space representation for a particular node centre and its immediate vicinity would not be followed by concomitant changes in the control actions to the required extent. Thus, the control action would not be representative of the modified state node and its current sphere of influence; it would represent, instead, the established control based upon a premature partitioning of state space.

The utility of the EUCART lies in the generality of the resulting approach when coupled with reinforcement learning. No *a priori* partitioning of state space is required unlike "boxes" (Mitchie and Chambers, 1968) or the original BSA implementation (Barto *et al.* 1983). In principle, the new approach could be applied to other dynamical systems with little modification without the requirement for a alternative fixed state space partition specific to the new system. This indicates the possibility of general purpose autonomous neurocontrollers.

Note the wide variation of average trial duration for each ensemble of 8 or 10 trials. The stochastic nature of the control output for a new node results in widely varying state space trajectories while the control mapping is being established during each run. Within a single ensemble of trials, one run may have established a control mapping very quickly within a limited region of state space while another may still have low trial durations through initial control outputs pushing the state space trajectory further away from a desired region and causing the creation of many naive nodes requiring training.

Figure 11 shows the average increase in the number of EUCART nodes for both the full set of 10 runs and for the 8 trial subset. Both the 8 run averages and 8 run maxima reflect convergence to a final set of desirable control actions. The 10 run averages and 10 run maxima indicate that adequate state space coverage has not yet been achieved for the remaining two runs; adequate coverage in the present context means that a control mapping has been established which maintains control for a cart-pole system starting with a given set of initial conditions. Whether coverage is adequate given a different set of initial conditions is another matter.

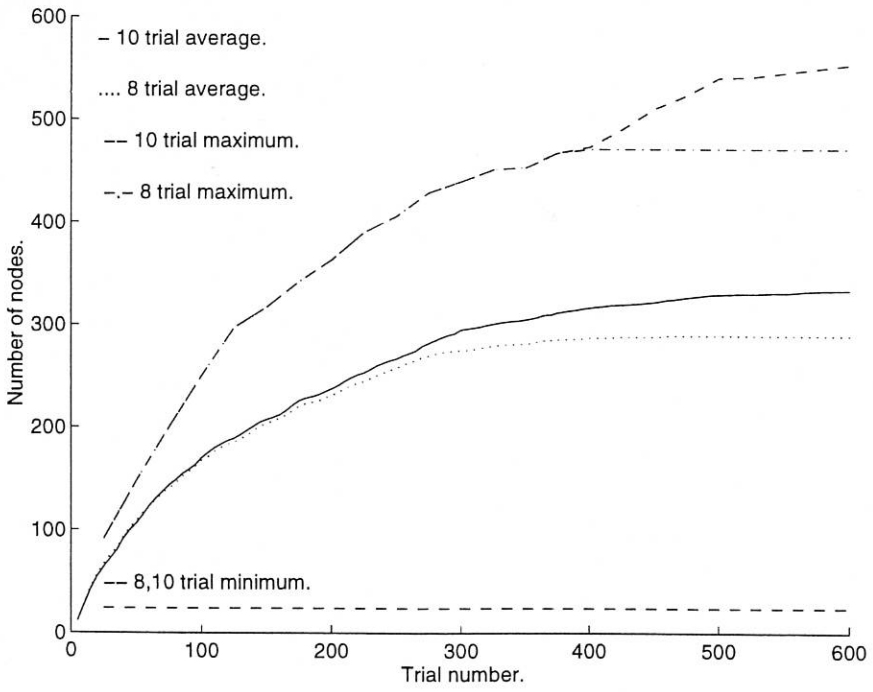

Figure 11 Simulation results showing the increase in the number of EUCART nodes representing individual state space regions and their associated control actions.

As expected for the neurocontroller performance, the trend is towards greater trial durations as the trial number increases. However, the increases in trial duration are not monotonic. This is because the addition of a new EUCART node introduces an initial arbitrary control action. This sometimes pushes the state space trajectory into previously unencountered regions of state space or a region where the control actions are not properly established. The neurocontroller is then likely to fail if the well-established state space regions are not re-entered quickly. Also, new nodes are sometimes added to cover "gaps" in state space and their influence replaces some well-established state space regions with naive coverage because the regions are now associated with a new node (i.e. the new node centre is now nearer to states previously encompassed by other nodes).

Figure 12 illustrates the situation schematically. The dark border shows the decision boundary within which states belong to the new node. The degree of disruption caused by a new (and naive) node depends upon the extent of overlap. The extent of overlap, in turn, is a function of the Euclidean distance between the category centres. If this distance, for a particular node with respect to its nearest neighbour, exceeds twice the maximum possible category radius, then no overlap will occur until a new node is added which violates the minimum distance condition. The dependence of overlap on inter-category distance is exploited in the modified EUCART decoder implementation discussed in section 4.2.

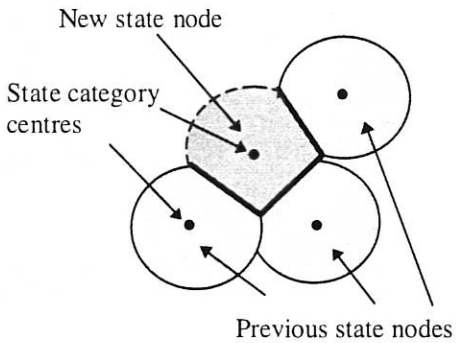

Figure 12. A schematic illustration of the problem of overlap associated with adding new nodes. The decision region for the new node now includes space previously covered by the original EUCART nodes. Disruption to established control actions within these regions is possible until the new node has learnt to represent a desirable control action.

As the results show, performance is eventually recovered when the new nodes learn to represent desirable control actions. The undesirable disruption is reminiscent of the *stability-plasticity dilemma* (Carpenter and Grossberg, 1987a) which states that adaptive systems must balance the requirement for stable learning of information against the requirement of plasticity and adaptation to novel phenomena.

With fixed non overlapping decoders, the neurocontroller is a pre-established look-up table which is filled during learning. Although there is no overlapping and hence no disruption of learning between state space regions, *a priori* assumptions are made by an operator which restrict the autonomy of a learning system; such assumptions include a prespecified state space granularity and a prespecified distribution of state space categories. The disruption effect is a consequence of the autonomy of a

EUCART-based neurocontroller; attempts to improve neurocontroller performance must include a reduction of this disruption effect without reducing the level of autonomy.

Figure 13 shows the results of a typical run. Again, the results are plotted as an average of bins of 5 consecutive values and not 5 averages of 8 or 10 runs. This time, 5 consecutive trial values are used. The graph indicates some correlation between increases in node numbers and disruption of trial duration. This is readily apparent at around trial 400 with the small increase in the number of EUCART nodes occurring simultaneously with a drop in the trial duration before recovery and final convergence. From inspection of Figure 13, it is apparent that the trend is towards increasing trial durations until the ceiling of 10,000 seconds is reached. The effect of transient disruptions caused by the addition of new nodes is more pronounced when winner-takes-all dynamics are used because a trained category node is replaced outright by a naive node which, henceforth, wins the competition in a given region until, possibly, replaced by a new node. Over time, this naive node will be trained and reflect the control mapping correctly within a given region of state space.

4.1.2 EUCART, Incremental Clustering and Stability

The last point in section 4.1.1. raises the question of stability. The incremental clustering algorithm of EUCART gradually builds a *cover* over regions of state space; whilst the cover is being built, transient disruptions will occur. When no "gaps" exist in a region of state space, disruptive naive nodes will no longer be required and a tessellation of this region by the choice function hyperplanes, between neighbouring category centres, will have formed.

Moore (1989) proposes two types of stability for incremental clustering algorithms *viz.*

- *Stable* 1: no prototype vector can "cycle", or take on a value that it had at a previous time (provided it has changed in the meantime), and

- *Stable 2*: only a finite number of clusters are formed with infinite presentation of the data.

Moore modifies the condition of Stable 1 to include the case where a prototype vector may include a previous value but it must eventually stop moving. The condition of Stable 2 is also restated as:

"in a bounded input space, condition (2) is equivalent to requiring that prototype vectors do not get arbitrarily close to each other."

EUCART is Stable 1, in the modified sense. for a given category, the category centre will stop moving when the EUCART category reaches its maximum extent; the category centre may pass through a previous value but will converge towards its final position in the fully extended category. Analogously to fuzzy ART, (carpenter *et al.* 1991) $\left\|\mathbf{u}_j^E\right\|$ monotonically decreases and $\left\|\mathbf{v}_j^E\right\|$ monotonically increases until the category reaches its maximum diameter; at this point the category has stabilised.

EUCART is also Stable 2 because the input space is bounded and thus requires a finite number of hyperspheres to contain it. Category hyperspheres may extend beyond the input space but, where they do, no input vectors will be found there by definition; this "fictitious" input space allows a complete cover of the Euclidean input space by hyperspheres of a fixed radius and thereby obviates the requirement of collections of hyperspheres near the input space boundary with radii tending to zero.

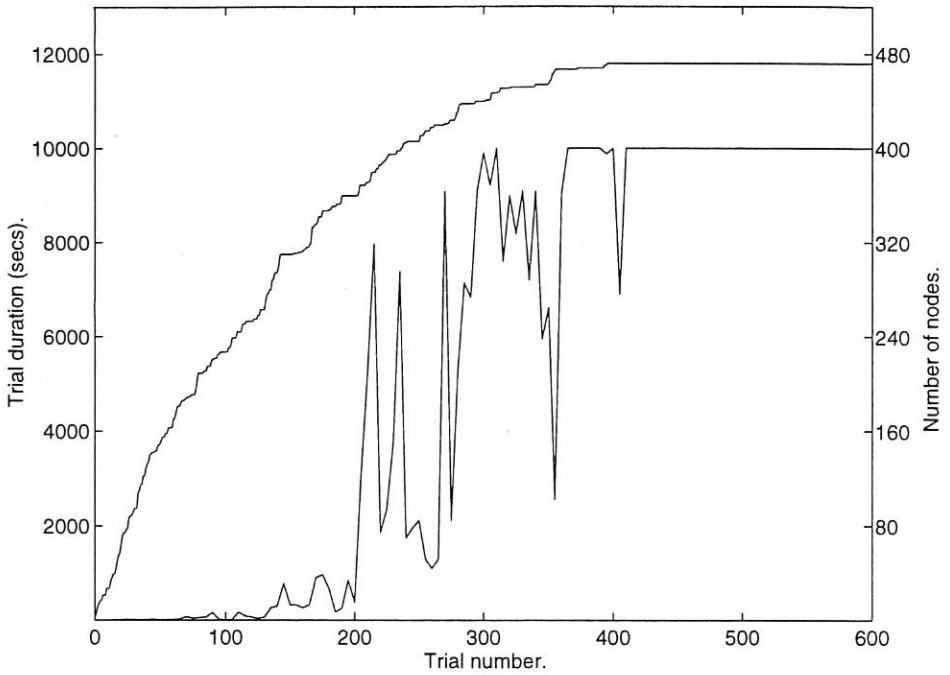

Figure 13. One typical run from the ensemble. Results are plotted as averages of five consecutive trials. Note the transient disruptions caused by the addition of new nodes.

The shortest run of the set of runs converges after just 10 trials with only 24 nodes. This set of control actions is almost certainly limited as comparatively little of state space has been explored. The controller would not be expected to be as robust and to possess as good disturbance rejection properties as those controllers with many more nodes indicating a wider experience of state space. From the point of view of robustness, the random perturbations caused by the introduction of naive nodes have a beneficial effect on the long term experience of the neurocontroller by extending its experience into new state space regions.

In many control applications these random perturbations by the naive nodes may not be desirable or practical when working within a real environment (it may be dangerous) but including a model of the environment alongside a neurocontroller may allow 'what if' probing by the neurocontroller to improve convergence. A better solution perhaps, would be to set failure limits for the reinforcement learning neurocontroller which lay within regions of performance recovery by an operator so that learning from failure did not necessarily entail disastrous consequences within a real operating environment. Failure would then represent undesirable system states to be avoided by a neurocontroller and which lead to an operator warning to allow manual recovery of performance.

The naivety of neurocontrollers with comparatively few nodes is considered in section 4.3 which illustrates the adaptiveness of the EUCART approach; there, it is shown that when new regions of state space are encountered, a EUCART-based neurocontroller is able to adapt without catastrophic forgetting (Sharkey and Sharkey, 1994). The next section will present a modified EUCART-based neurocontroller which attempts to reduce the disruption by naive nodes during incremental clustering.

4.2 A Modified EUCART Decoder Implementation

4.2.1 Nearest Neighbour Priming

As discussed previously in section 4.1, it was found that naive nodes, added to fill 'gaps' in state space coverage, often disrupted currently established control information. Although recovery and convergence eventually occurs, it would be desirable to minimise the disruption during learning. Where disruption is likely to be the most severe, it is because the region of influence of a naive node infiltrates established nodes in the neighbourhood of the new node. Thus, some state vectors which were previously encompassed by the original nodes are now nearer to the new node centre and thus elicit the control action determined by the new node parameters; the parameters have not yet had time to tend towards desirable values because the node has been newly created.

To reduce disruption when a new node is added, information from surrounding nodes must be taken into account. Instead of beginning with the initial control action weight of a new node set to zero, the weight values of n nearest neighbours can be combined to give an initial weight value. In the present modified implementation, a scalar weighted average of the form

$$z_0^{new} = \frac{1}{n}\sum_{i=1}^{n}\eta_i z_i$$

is assigned as the ASE weight for the new node, where η_i is the scalar *contribution weighting* of the i^{th} neighbouring state space category and z_i is the i^{th} ASE weight. The contribution weighting takes the following two factors into account,

- *category centre distance*; the further away the neighbouring state space category is from the new category centre, the smaller the contribution to the initial ASE weight should be, and

- *category node age*; the 'older' the neighbouring category in terms of learning experience, the more established the control action is (less likely to be disruptive); the contribution to the initial ASE weight should be reduced for recent (naive) categories.

Ideally, new node priming is determined by close, well established categories with desirable control actions. The form of the contribution weighting for i^{th} the nearest neighbour is,

$$\eta_i = \left(T_i(\mathbf{c}_{new})\bar{x}_i\right)^p$$

where $\mathbf{c}_{new}$ is the newly created category centre, $T_i(.)$ is the EUCART choice function for the i^{th} category node, $\bar{x}_i$ is the ACE trace for the i^{th} node and $p \geq 1$ is the power used for contrast enhancement. Note that $T_i(.) \leq 1$ and $\bar{x}_i \leq 1$ imply that $\eta_i \leq 1$.

The parameters used in the runs of the modified EUCART system were the same as those used in the runs of the unmodified version. The number of nearest neighbours, used to determine the initial ASE weights of new nodes, is n=5 and the power p=5 is used to contrast enhance the contribution weighting, η_i.

The K nearest neighbours technique coupled with a variant of ART was used by Zhang and Grant (1992) in conjunction with the boxes learning algorithm (Michie and Chambers, 1968). For a new input vector, if the input does not exceed a membership threshold, the K nearest neighbours to the input are selected and updated according to the degree of membership of the input with respect to the category nodes. Category centres are updated proportionately to input vector membership using a modified competitive learning scheme and represent the centre-of-gravity for a cluster of input patterns in state space encompassed by the category node. The K nearest neighbour method is used in this context to update *existing* nodes; if no nodes fulfil the membership criterion, then a new node is created.

In the present paper, K nearest neighbours are used to prime the new node to minimise disruption during the learning process. Figure 14 shows the results of ten runs using the parameters of the previous set of runs. This time, all ten runs converged within 600 trials.

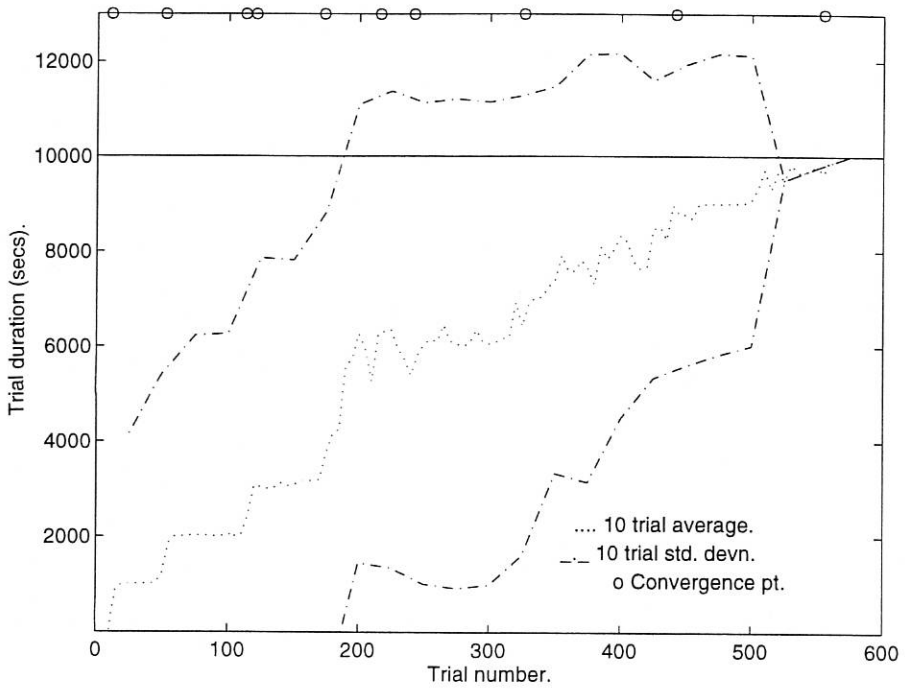

Figure 14. Simulation results showing the performance of the ASE / ACE system with the EUCART state space decoder using nearest neighbour priming. The trials were averaged over ten runs before plotting in bins of five trials; all runs terminated within 600 trials.

After about 250 trials, the average number of nodes for the modified version of EUCART (Figure 15) is similar to that of the eight run average of the original version. This indicates that the increased average for the ten runs using the original version of EUCART is caused by the two runs which did not converge within the 600 trial limit. The time required-to-convergence and the number of nodes are linked by the fact that an increase in the number of naive nodes requires an increase in learning time to modify the new parameters.

Priming is only effective in reducing disruption in regions of well-established nodes represent a desirable control mapping. Without priming, the effect of a new node is to issue a random control action which may cause the state space trajectory to enter new or weakly established regions of state space. Often, the result is that well established regions cannot be re-entered and failure occurs subsequently. Where state space areas are not well established, priming has little or no effect because of the contrast enhancement of:

i) *distance effects,* where neighbouring nodes are relatively far apart in sparsely represented regions,

ii) *experience effects,* where weakly established nodes contribute little information to the new node.

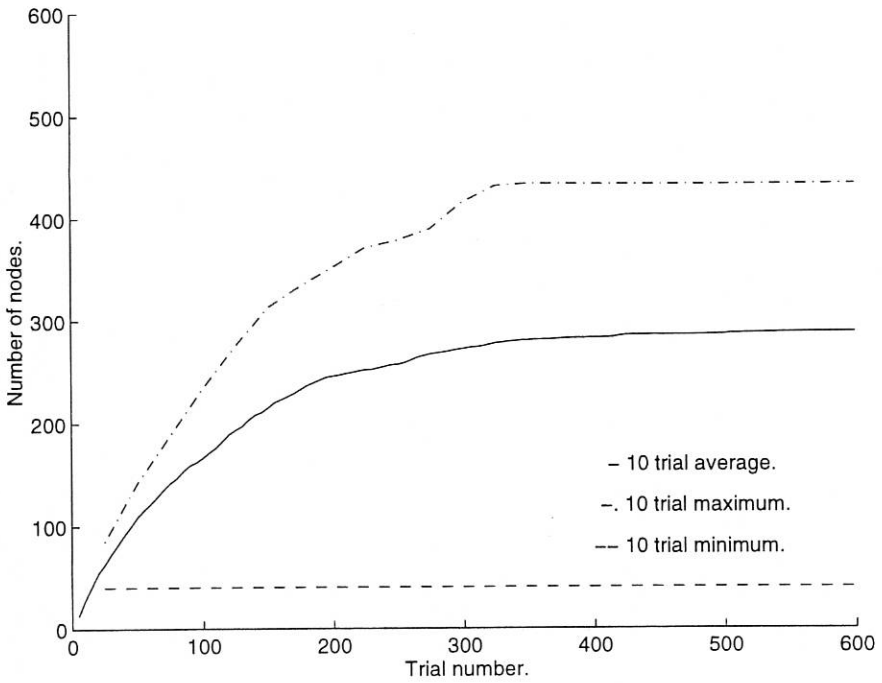

Figure 15 Simulation results showing the increase in the number of EUCART nodes when nearest neighbour priming is used.

Figure 16 shows a run using nearest neighbour priming which uses the same parameters as those used to produce the results of Figure 14.; the same random number seed was used to illustrate the difference in disruption effects. Comparing Figures 14 and 16 shows that the increase in the number of nodes is approximately the same until about trial 150 where the run using nearest neighbour priming begins to produce slightly fewer nodes and converges at around trial 250. The two peaks of Figure 16 that exceed 8000 seconds indicate that, although disruption occurs, the control mapping is becoming more effective. Without nearest neighbour priming in Figure 14, further disruption occurs for nearly 200 trials following the two peaks similar to those of Figure 16.

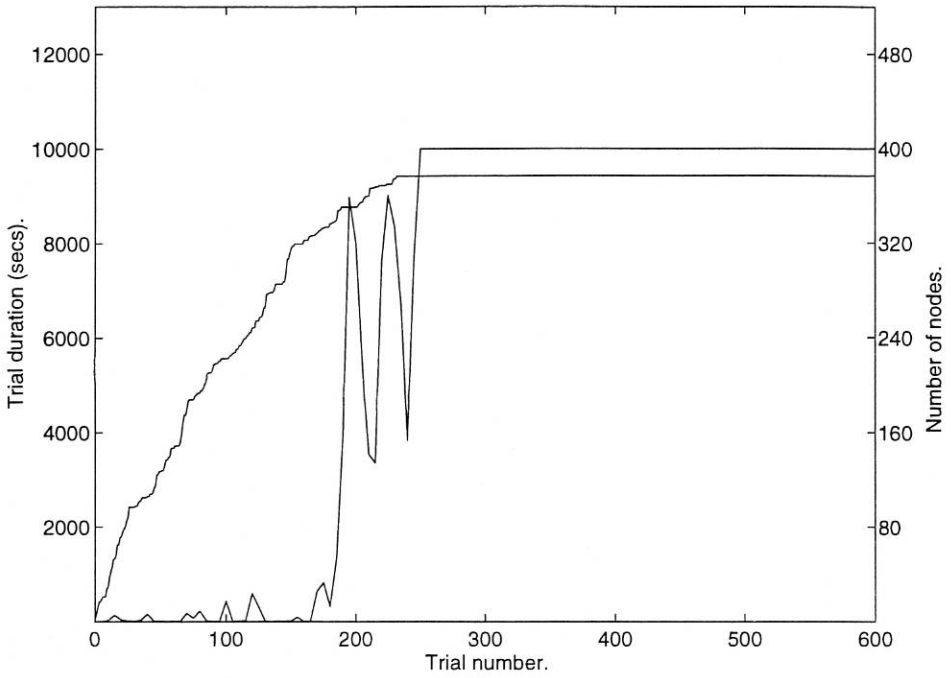

Figure 16. One typical run from the ensemble using nearest neighbour priming; the parameters are the same as those used in the run of Figure 9.

Figures 17 and 18 illustrate the effect of nearest neighbour priming upon one of the two runs which did not converge originally within the 600 trial limit. Figure 17 shows the original performance without priming and Figure 18 illustrates performance with priming.

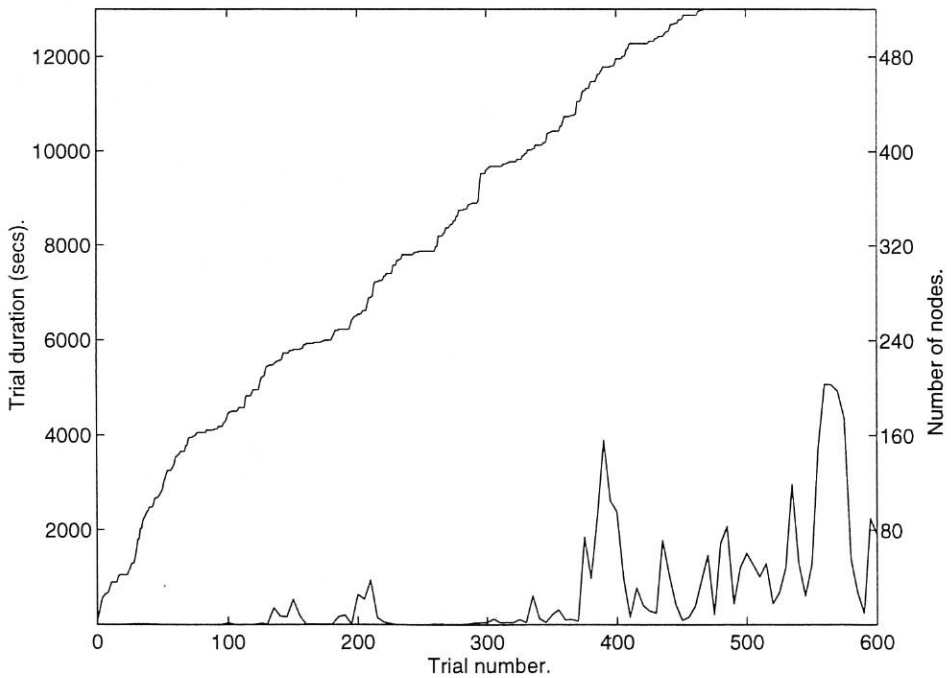

Figure 17. An example of a run without nearest neighbour priming which did not converge within 600 trials. Convergence occurred eventually after about 1200 trials. Note the rapid increase in nodes and the large variation in trial durations.

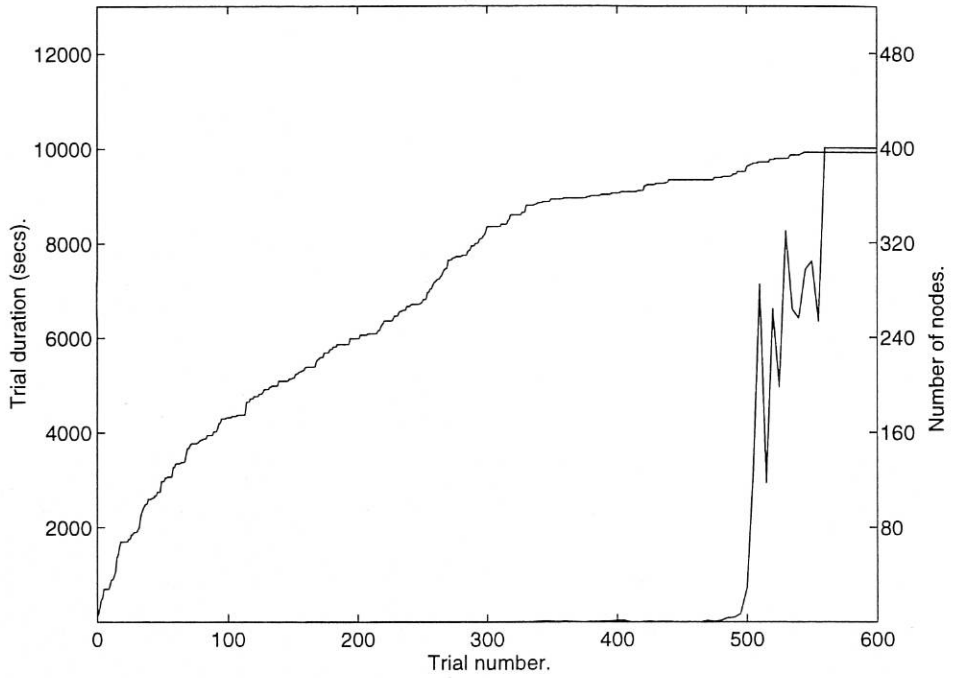

Figure 18. A run, using nearest neighbour priming, with the same parameters as those used in the run of Figure 17. Note the sizeable reduction in the number of nodes as compared to Figure 17.

4.3 Neurocontroller Adaptability with Different Initial Conditions

A question to ask is "what happens when different initial conditions are used in the cart-pole system simulation?". In other words, "how adaptive is a trained EUCART-based neurocontroller?". Any candidate neurocontroller must be plastic and must not suffer catastrophic forgetting when new information is encountered. Table 1 shows the results of a EUCART-based neurocontroller, using nearest neighbour priming, operated under different initial conditions following training; only the angle was changed in the simulations to illustrate the situation. With only 40 nodes, the controller has not explored much of state space and has to reduce its naivety through exploration of unknown regions.

New initial condition	No. of new trials required	No. of new nodes required
+1°	0	0
+3°	0	0
+6°	171	275
+11°	371	319

Table 1. Results from a naive trained neurocontroller, consisting of 40 nodes, and the effect of changing the initial angle. After retraining, the angle was reset; no disruption of previous learning was observed.

The neurocontroller was trained with an initial state vector specified of zero. After training, the initial pole angle was changed before restarting the simulation. To check that learning under the new initial condition did not disrupt previous experience, the initial condition was then reset to the original value after successful training with the new condition. Resetting the initial condition did not disrupt the established control mapping in any of the cases.

Initial conditions of +1° and +3° were dealt with easily by the 40 node neurocontroller and required no new nodes or further trials. For +6°, a further 275 nodes and 171 trials were required. Setting the angle to 11°, after resetting the neurocontroller, and training from zero initial conditions, resulted in a further 319 nodes (over and above the original 40 nodes) trained over 371 trials. The angle of 11° is near to the failure limit. These results indicate that a naive EUCART-based neurocontroller is able to adapt to the new conditions without disrupting previous learning. To illustrate the naivety of the 40 node neurocontroller, an simulation was carried out using a trained 377 node neurocontroller with a new initial condition of +6° for the angle; for this neurocontroller, only a single further trial was required to train a single new node. The more extensive experience of the 377 node neurocontroller, compared with the 40 node neurocontroller of Table 1, is reflected in the reduced requirement for extra learning.

5. Conclusions and Further Work

It has been shown that the EUCART state space decoder, in conjunction with the ASE / ACE subsystems, is able to learn a control mapping for a non-linear control problem. The resulting neurocontroller is autonomous and does not require *a priori* information other than a choice of operating parameters. The incremental clustering algorithm of EUCART successfully partitions state space and allows on-line adaptation to new regions which may not be accounted for by an *a priori* partitioning.

The EUCART decoder simulations also extend the BSA implementation by considering the effect of new initial conditions on a trained neurocontroller that has converged to the simulation ceiling using the "all-zero" initial state. Indeed, using a EUCART decoder has extended the generality of the BSA implementation of reinforcement learning by indicating the possibility of developing "general purpose" neurocontrollers; such controllers may not be as precise as those designed for specific tasks using high precision analysis and design techniques, but would be more readily applicable, "off-the-shelf", and ready to adapt through experience. This approach entails a movement away from highly accurate static mappings towards a more adaptive approach exemplified by the principle of increasing precision vs. decreasing intelligence (Saridis, 1989).

Isolating the state space decoder task from the control action learning task and treating it as a 'black box' allows the development of variant reinforcement learning networks which still retain the original ASE / ACE specification. The main operational criterion for a candidate state space decoder used in this way is that it assigns a unique representation to distinct regions of state space; the regions may overlap in places but the state space representation and parameter updating methods must account for this. For example, winner-takes-all dynamics can be used to choose a winning neurocontroller node or parameters for several nodes can be updated in proportion to their respective activation levels (membership functions). The latter approach is consistent with fuzzy rule bases where multiple rules may be activated. The present paper uses the winner-takes-all method for choosing prediction and control information for consistency with the original ASE / ACE implementation which uses a fixed non-overlapping state space partitioning. Although EUCART categories overlap, only one category is selected at any one time so potential conflict is avoided.

The EUCART self-organising state space decoder discussed in this paper has removed the need for such *a priori* restrictions but in doing so has introduced the problem of disrupted learning during incremental partitioning of state space. This disruption is inevitable as the introduction of new nodes causes overlapping which changes the state space tessellation and thus the established control mapping. Although the EUCART decoder system eventually stabilises, it is desirable to reduce transient effects during learning. The nearest neighbour modifications go some way towards reducing disturbances caused by the addition of new nodes but a more distributed representation of state space and the associated control mapping is desired while retaining the attractive properties of the ASE / ACE reinforcement learning system. The original BSA implementation does not preclude this. Indeed, the seminal paper of Barto *et al*, (1983) mentions this possibility.

The power law, for the nearest neighbour weighting using the fifth power, was chosen on the basis of empirical observation. Other forms of contrast enhancement law may be more suitable. The introduction of such parameters highlights one of the problems of self-organising systems; the danger is that by using self-organisation, other *a priori* assumptions are substituted for those assumptions that are to be removed. The requirement of numerous parameters can possibly reduce the utility of self-organisation over *a priori* structuring of information. On the point of *a priori* inclusion of information, Procyk and Mamdani (1979) state that

"it is impossible to design a controller which need not assume anything about its environment. One can only strive to lessen its dependency and sensitivity to it."

The minimisation of built-in assumptions about the environment must be a guiding principle in the development of neurocontrollers but with the proviso that, wherever possible and convenient, known facts can be included in the neurocontroller structure if performance will be improved by doing so. Having to learn known facts—that could otherwise be built-into a neurocontroller to improve performance—cannot always be justified by claims of autonomy.

Nearest neighbour methods could be used to compute both the control output and the predicted failure values for a given input vector by category membership value (Zhang and Grant, 1992) in conjunction with new node priming. This would probably reduce the disruption caused by the addition of new nodes and augment the limited applicability of new node priming by updating all nodes triggered by a state space trajectory entering overlapping state space regions. A method for distributed processing of predictions and control outputs within the ASE and ACE processing units is required if internal representations of the state space and control mapping are to be smoothed out.

Although nodes represent individual state space regions and their associated control actions, the neurocontroller is not at all transparent to an operator. The nodes, in effect, represent micro-rules of the form 'if the state space vector is in the region surrounding centre x then output y'. These numerical rules are not very meaningful and, in many cases, clusters of micro-rules could be replaced effectively by a more general rule. Pruning and generalisation of groups of micro-rules is possible but the associated technicalities may be obviated by using a more efficient state space representation to begin with; for example, using nodes to represent fuzzy rules. Fuzzy systems are much better suited to knowledge extraction (e.g. Berenji and Khedkar, 1992; Jang, 1992,1993; Jang and Sun, 1995) than networks using micro-rules but introduce other considerations such as the choice between a rule base with a fixed number of rules or a self-organising rule base; the task of rule extraction (Wang and Mendel, 1992) and the task of tuning the fuzzy membership functions.

The distribution of ASE / ACE dynamics is compatible with the fuzzy approach as it is possible that multiple rules are activated and contribute to the control or predictive outputs. Similarly, distribution of state space decoding across multiple input lines may reduce the effect of state space node overlap when EUCART is used.

In this paper, we have shown that the decoder section of the original BSA implementation provides a basis for the development of variant reinforcement learning architectures. The EUCART decoder is self-organising and is compatible with the original ASE /ACE formulation. Other types of state space decoder that are similarly compatible are possible. The very fact that the principles of self-organisation and reinforcement learning can co-exist together is an exciting prospect for artificial neural systems development and points a way forward to the development of autonomous neural systems that require much less outside intervention than at present.

References

Albus, J. A Model of the Brain for Robot Control (1979) parts I, II and III, *BYTE magazine*, June, July and August issues pp 10-34, pp. 55-95, pp 66-80. BYTE Publications Inc.

Albus, J. S. (1975a). A New Approach to Manipulator Control. The Cerebellar Model Articulation Controller (CMAC), *Trans. ASME. Jnl. Dyn. Sys. Meas. and Control.* **63** (3), 220 -227

Albus, J. S. (1975b). Data Storage in the Cerebellar Model Articulation Controller (CMAC), *Trans. ASME. Jnl. Dyn. Sys. Meas. and Control.* **63** (3), 228 -233

Anderson, C. W., (1989) Learning to Control an Inverted Pendulum using Neural Networks, *IEEE Control Systems Magazine,* April

Barto, A. G. (1992). Reinforcement Learning and Adaptive critic Methods, in White, D. A. and Sofge, D. A. (Eds.) *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches.*

Barto, A. G. and Sutton, R. S. (1982) Simulation of Anticipatory Responses in Classical Conditioning by a Neuron-Like Adaptive Element, *Behavioral Brain Research*, **4**, 221-235.

Barto, A. G. Sutton, R. S., and Anderson, C. W. (1983). Neuronlike Adaptive Elements that can Solve Difficult Learning Control Problems *IEEE Trans. Syst. Man. Cybern.* Vol. SMC-13, 834-846.

Berenji, H. R. and Khedkar, P. (1992). Learning and Tuning Fuzzy Logic Controllers Through Reinforcements. *IEEE Trans. On Neural Networks,* **3** (5), 724-740

Carpenter, G. A., & Grossberg, S, (1987a). A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing,* **37**, 54-115.

Carpenter, G. A., & Grossberg, S, (1987b). ART 2: Self-organisation of Stable Category Recognition Codes for Analog Input Patterns. *Applied Optics,* **26**, 4919-4930.

Carpenter, G. A., & Grossberg, S, (1989). ART 3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures. *Neural Networks,* **3**, 129-152.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., & Rosen, D. B.,(1992). Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Transactions on Neural Networks,* **3**, 698-712.

Carpenter, G. A., Grossberg, S. & Reynolds, J. H., (1991a). ARTMAP: Supervised Real-time Learning and Classification of Nonstationary Data by a Self-organizing Neural Network. *Neural Networks,* **4**, 565-588.

Carpenter, G. A., Grossberg, S., & Rosen, D. B.,(1991b). Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks,* **4**, 759-771.

Daynan, P. and Hinton, G.E. (1993) Feudal Reinforcement Learning in Henson S.J., Cowan, J. D. and Giles, D. L. (Eds.) Advances in Neural Information Processing, Morgan Kaufmann, San Mateo, CA.

Edelman, G. M. (1989) Neural Darwinism: The Theory of Neuronal Group Selection pp.-4, Oxford University Press, Oxford.

Friedland, B., (1987). Control System Design: An Introduction to State space Methods, McGraw-Hill Book Company, New York

Fujita, O. (1992). Optimization of the Hidden Unit Function in Feedforward Neural Networks *Neural Networks,* **5**, 755-764.

Grossberg, S., (1980). How Does a Brain Build a Cognitive Code? *Psychological Review,* **1**, 1-51.

Hebb, D. O. (1949) *Organization of Behavior* ,Wiley, NY.

Hilgard, E. R., and Bower, G. H., (1966). *Theories of Learning*. Century Psychology Series Appleton-Century-Crofts, New York

Hocking, L. M., (1991). Optimal Control: *An Introduction to the Theory with Applications*, Oxford Applied Mathematics and Computing Science Series, Clarendon Press, Oxford.

Hull, C. L. (1943). Principles of Behavior: *An Introduction to Behavior Theory*, D. Appleton-Century Company, New York.

Hull, C. L. (1951). *Essentials of Behavior*, Yale University Press, New Haven.

Hull, C. L. (1952) *A Behavior System: An Introduction to Behavior Theory concerning the Individual Organism.* Yale University Press, New Haven.

James, W (1892) Textbook of Psychology: Briefer Course, Macmillan & Co. Ltd, London.

Jang, R. J-S. (1992). Self-Learning Fuzzy Controllers Based on Temporal Back Propagation, *IEEE Trans. on Neural Networks*, **3** (5).

Jang, R. J-S. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. on Syst. Man and Cybern.,* **23**, (3), 665-685

Jang, R. J-S. and Sun, C-T. (1995). Neuro-fuzzy Modelling and Control. *Proc. IEEE*, **83**, (3)

Kandirkamanathan, V. and Niranjan, M (1992) Technical Report CUED/F-INFENG/TR.111, Cambridge University, Cambridge, UK.

Klopf, A. H., (1986), A Drive-Reinforcement Model of Single Neuron function: An Alternative to the Hebbian Neuronal Model in J. Denker (Ed.) *AIP Conference Proceedings 151, Neural Networks for Computing*, 77-85 New York, AIP

Klopf, A. H., (1988) A Neuronal Model of Classical Conditioning, Psychobiology, **16**, (2), 85-125.

Kohonen, T. (1989), Self-Organisation and Associative Learning, Springer-Verlag, Heidleberg.

Kosko, B., (1992). Neural Networks and Fuzzy Systems: A dynamical Systems Approach to Machine Intelligence (pp 263 - 298) Prentice-Hall International, Inc.

Lin, C-S. and Kim, H.(1991) CMAC-based Adaptive Critic Self-Learning Control. *IEEE Transactions on Neural Networks.* **2**, 5, pp. 530-533.

Marriott, S and Harrison, R. F. (1994). A modified Fuzzy ARTMAP Architecture for the Approximation of Noisy Mappings, *Research Report No 522,* The University of Sheffield, U.K.

Michie, D. and Chambers, R. A.,(1968). BOXES: an Experiment in Adaptive Control, in *Machine Intelligence 2,* E. Dale and Michie, D. Eds. Edinburgh: Oliver and Boyd.

Moore, B., (1989). ART 1 and Pattern Clustering. In Touretzky, D. *et al* (Eds.), *Proceedings of the 1988 Connectionist Models Summer School,* (pp 174-185) San Mateo, CA, Morgan Kaufmann Publishers.

Myers, C. E. (1992) *Delay Learning in Artificial neural Networks*, Chapman and Hall, London.

Pavlov, I. P. (1928). Lectures on Conditioned Reflexes, Vol. I, (Trans.) Gantt, W. H., Lawrence & Wishart Ltd, London.

Platt, J. C. (1991). A Resource Allocating Network for Function Interpolation. *Neural Computation* **3** (2), 215-225.

Prockyk, T. J. and Mamdani, E. H. (1979). A Linguistic Self-Organising Process Controller, *Automatica,* **15** 15-30

Ritter, H., Martinetz, T., and Schulten, K. (1992). *Neural Computation and Self-Organizing Maps: an Introduction*, Addison-wesley Publishing Company, Reading MA.

Saridis, G. N. (1989). Analytic Formulation of the Principle of Increasing Precision with Decreasing Intelligence for Intelligent Machines. *Automatica* **25**, 3, 461-467.

Selfridge, O. (1959). Pandemonium: A paradigm for learning, in *Symposium on the mechanisation of though processes*, London, HMSO.

Sharkey, N. E. and Sharkey, A. J. C. (1994). Understanding Catastrophic Interference in Neural Nets. *Research Report CS-94-4,* University of Sheffield, U. K.

Sutton, R. S. (1988) Learning to Predict by the Methods of Temporal differences, *Machine Learning,* **3**, 9-44.

Sutton, R. S. (Ed.) (1992). Reinforcement Learning: A Special Issue of *Machine Learning* on Reinforcement Learning Kluwer Academic Publishers.

Sutton, R. S., and Barto, A. G. (1981). Towards a Modern Theory of Adaptive Networks: Expectation and Prediction, *Psychological Review,* **88** (2), 135-170.

Sutton, R. S., and Barto, A. G. (1990). Time-Derivative Models of Pavlovian Reinforcement, in Gabriel, M. and Moore, J. *Learning and Computational Neuroscience: Foundations of Adaptive Networks.* MIT Press, Cambridge MA.

Sutton, R. S., Barto, A. G. and Williams, R. J. (1992). Reinforcement Learning is Direct Adaptive Optimal control, *IEEE Control Systems Magazine,* April, 19-22.

Tolle, H. and Ersu, E. (1992). The Cerebellar Model of J. S. Albus. in *Lecture Notes in Control and Information Sciences* Thomas, M. and Wyner, A. (Eds.), **172** 29-93 Springer-Verlag.

Wang, Li-Xin, and Mendel, J. M. (1992). Generating Fuzzy Rules by Learning from Examples, *IEEE Trans. Sys., Man and Cybern.* **22** (6), 1414-1427.

Zadeh, L. A., (1965). Fuzzy Sets. *Information and Control,* **8**, 338-353.

Zhang, B. and Grant. E. (1992) Using Competitive Learning For State-Space Partitioning, *Proceedings if the IEEE international Symposium on Intelligent Control,* 391-395

Appendix A

The cart-pole simulation was carried out as state in Barto *et al* (1983) with minor modifications. The state vector was reset to $x = \dot{x} = \theta = \dot{\theta} = 0$ after each trial; failure was indicated by a reinforcement signal of -1 when either the cart displacement, x or pole angle, θ left their ranges of [-2.4m, 2.4m] and [-12°,+12°] respectively. All trace variables were set to zero at the start of each trial. All weights were set to zero at the start of each run. Each run of the set of ten used random numbers from a different seed value. See Barto *et al* (1983) for further details.

The parameter values used for the ASE / ACE subsystems were a=0.8, b=0.5, δ=0.9 γ=0.98 and σ=0.01. Here, α and γ differ from the BSA implementation. As stated in the body of the text, the former was reduced substantially to prevent premature establishment of control actions. The latter was used to reduce the reinforcement prediction discounting but does not appear to have any significant effect; the change is noted here for completeness.

The simulation equations for the cart-pole system are the following non-linear differential equations (Barto *et al*, 1983):

$$\ddot{\theta} = \frac{g\sin\theta + \cos\theta \dfrac{-F - ml\dot{\theta}^2 \sin\theta + \mu_c \,\mathrm{sgn}(\dot{x})}{m_c + m} - \dfrac{\mu_p \dot{\theta}}{ml}}{l\left[\dfrac{4}{3} - \dfrac{m\cos^2\theta}{m_c + m}\right]}$$

$$\ddot{x} = \frac{F + ml\left[\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta\right] - \mu_c \,\mathrm{sgn}(\dot{x})}{m_c + m}.$$

The parameters are those used in Barto *et al* (1983) with no changes, and the system is simulated using Euler's method with a timestep of 0.02 seconds. A control force is applied at every timestep until failure occurs. The neurocontroller only has access to the cart-pole system states in the form of a state vector. It does not have privileged access to a model or any pre-existing cost function.

Appendix B
The Monotonic Increasing Property of the fuzzy ART Choice Function

When $\mathbf{w}_j$ is a fuzzy subset of $\mathbf{I}$, the Fuzzy ART choice function is of the form

$$f(x)=\frac{ax}{b+x}.$$

Theorem: For a function $f:[0,1]\rightarrow[0,1]$, $f(x)=\frac{ax}{b+x}$, where a and b are positive constants, given some $x_1,x_2\in[0,1]$ $f(x_2)\geq f(x_1)\Rightarrow x_2\geq x_1$.

In the paper, we refer to the above property as the "monotonic increasing property" or M.I.P.

Proof: For some $x_1,x_2\in[0,1]$ assume that $f(x_2)\geq f(x_1)$, i.e.

$$\frac{ax_2}{b+x_2}\geq\frac{ax_1}{b+x_1}.$$

Using the rules of inequalities we get

$$\frac{ax_2(b+x_1)}{(b+x_1)(b+x_2)}\geq\frac{ax_1(b+x_2)}{(b+x_1)(b+x_2)}$$

giving,

$abx_2\geq abx_1$

and

$x_2\geq x_1$. •

Similarly, it can be proved that

$x_2\geq x_1\Rightarrow f(x_2)\geq f(x_1)$.