A SELF-LEARNING RULE-BASED CONTROLLER EMPLOYING APPROXIMATE REASONING AND NEURAL NET CONCEPTS

by

Chuen-Chien Lee

Memorandum No. UCB/ERL M89/84

21 July 1989

A SELF-LEARNING RULE-BASED CONTROLLER EMPLOYING APPROXIMATE REASONING AND NEURAL NET CONCEPTS

by

Chuen-Chien Lee

Memorandum No. UCB/ERL M89/84

21 July 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A SELF-LEARNING RULE-BASED CONTROLLER EMPLOYING APPROXIMATE REASONING AND NEURAL NET CONCEPTS

by

Chuen-Chien Lee

Memorandum No. UCB/ERL M89/84

21 July 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A Self-Learning Rule-Based Controller Employing Approximate Reasoning and Neural Net Concepts

Chuen-Chien Lee

Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720

ABSTRACT

A self-learning controller is proposed as an intelligent controller for dynamic processes, employing a control policy which evolves and improves automatically. A key component of the controller is a rule-based system which provides a linguistic description of control strategy. This strategy has the form of a collection of fuzzy conditional statements which are implemented and manipulated using fuzzy set theory. The inference engine of the controller is based on the principles of approximate reasoning, while its learning capability is provided by neuron-like elements, which are derived from animal conditioning theory. It is shown that the system can solve a fairly difficult control learning problem. More concretely, the task is 1-D pole balancing, in which a pole is hinged to a movable cart to which a continuously variable control force is applied. Simulation results demonstrate that improved learning performance can be achieved in relation to previously described systems employing bang-bang control. Furthermore, the proposed controller is relatively insensitive to variations in the parameters of the system, e.g., changes in the length and mass of the pole, initial angle, failure criteria, and slanting the base of the car-pole system.

July 21, 1989

Research supported in part by NASA Grant NCC-2-275 and AFOSR Grant 89-0084.

A Self-Learning Rule-Based Controller Employing Approximate Reasoning and Neural Net Concepts

Chuen-Chien Lee

Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720

I. Introduction

There are many complex industrial processes which cannot be satisfactorily controlled by conventional methods due to the unavailability of quantitative data regarding input-output relations. And yet, skilled human operators can control such systems quite successfully without having any quantitative models in mind. Furthermore, the operation of many man-machine systems requires the use of rules of thumb, intuition, and heuristics. In such cases, the use of rule-based control based on approximate reasoning becomes an increasingly attractive alternative.

In fact, during the past several years, rule-based controllers based on fuzzy logic and approximate reasoning [8,9] have emerged as one of the most active and fruitful areas for research in the application of fuzzy set theory [17]. Such controllers have been used to deal with uncertainty in complex, ill-defined processes, particularly in man-machine systems, where they are effective in capturing the approximate and qualitative aspects of human reasoning. Among the representative applications of fuzzy-logic-based controllers are the subway system in the city of Sendai [16], container ship crane control [15], elevator control [5], nuclear reactor control [7,2], and automobile transmission control [6,14]. Experience has shown that a rule-based controller using approximate reasoning makes it possible to emulate and even surpass the decision-making ability of a skilled human operator.

Although there is an extensive literature describing various applications of fuzzy logic controllers using approximate reasoning, the acquisition of the rule base in such controllers is not as yet well understood. In past applications, fuzzy decision rules have been obtained either from verbal expressions or observations of human operator control actions. Since domain experts and skilled operators do not structure their decision-making in any formal way, the process of transferring expert knowledge into a usable knowledge base is tedious and unsystematic. Our research aims at the development of a better understanding of such problems, with a view to enhancing the potential of fuzzy logic-based controllers. Figure 1 shows a schematic representation of the proposed controller which includes a neural net and a rule-based controller employing approximate reasoning.

The problem of learning via credit assignment [4] is described in Section II. The statement of the pole-balancing problem follows. This problem may be viewed as a canonical example of dynamic control. Some concepts from earlier related work are given in Section III. They serve as a basis for comparison of previous and proposed approaches. The proposed approach, a combination of techniques drawn from fuzzy logic and neural network theory, is presented in Section IV. Here, a rule-based controller using approximate reasoning (fuzzy logic controller) is introduced, and a learning system with two neuron-like elements which are derived from animal conditioning theory, is proposed. Computer simulation results are described in Section V. The paper closes with a concluding remark in Section VI.

II. The Problem

A. Learning via Credit-Assignment

Many rule-based controllers with approximate reasoning have been built to emulate human decision-making behavior, but few are focused on a key aspect of human learning, namely, the ability to create fuzzy decision rules and modify them on the basis of experience.

In machine learning, the problem of learning to control physical dynamical systems has been, and remains, a challenging goal. In this context, the credit-assignment problem is often encountered in adaptive problem-solving systems, and is especially acute when evaluative feedback is delayed or infrequent. Basically, the credit-assignment problem, is to determine a strategy for assigning positive credit (reward) to desirable actions and negative credit (punishment) to undesirable actions in a way that would lead to the achievement of a specific goal. In what follows, we describe an approach to the building of an intelligent rule-based system that can learn to control a dynamical system without prior knowledge of its input-output relations.

B. A Case Study: The Pole-Balancing Problem

Our approach focuses on a paradigmatic control problem - the pole-balancing problem - which has been the object of several studies in the literatures of control and neural networks.

Figure 2 shows the pole balancing system. A rigid pole is hinged to a cart, which is free to move on a one-dimensional track. The pole can rotate in the vertical plane of the track and the controller can apply an impulsive force of bounded magnitude to the cart at discrete time intervals. By balancing the pole, we mean that the pole never deviates by more than, say, 12 degrees, from the vertical. The equations of motion of the cart-pole system are not known to the controller, which implies that the cart-pole system is treated as a black box. What is known is a vector describing the cart-pole system's state at every time step. If the pole falls, it receives a failure signal. After a failure signal has been received, the system is reset to its initial state and a new attempt is made. On the basis of this evaluative feedback, the controller must develop its own control strategy and learn to balance the pole for as long as possible. Since a failure signal usually occurs only after a long sequence of individual control decisions, the sparsity of this signal makes the credit-assignment problem nontrivial.

The cart-pole system is modeled by two second-order differential equations which include the nonlinearities and reactive forces of the physical system [3,1]. The dynamics of the cart-pole system are characterized by four state variables, namely, angle of the pole with respect to the vertical axis, θ, angular velocity of the pole, $\dot{\theta}$, position of the cart on the track, x, and velocity of the cart, $\dot{x}$. The equations in question are:

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

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

where

g: acceleration due to gravity, 9.8 meter/ sec^2,

m_c: mass of cart, 1.0 kg,

m: mass of pole, 0.1 kg,

μ_c: coefficient of friction of cart on track, 0.0005,

μ_p: coefficient of friction of pole on cart, 0.000002,

l: half length of pole, 0.5 meter,

F: applied force: [-10, +10] newtons, continuous.

In testing the performance of the system, the simulator was run by applying the Adams predictor-corrector method with a time step of 20 ms on a Sun workstation.

III. Previous Related Work

There are two noteworthy previous studies which have addressed the pole-balancing problem. The first is that of Michie and Chambers [13] in 1968. They constructed a program called BOXES that learned to balance the pole by applying two opposite constant forces. The second study is that of Barto, Sutton, and Anderson [1] in 1983, which used neuronlike adaptive elements to solve the same problem by using two constant forces. In general, both approaches can handle the credit-assignment problem that we mentioned. In both, the state space is partitioned into several non-overlapping regions and no symbolic reasoning techniques are employed. Both are limited to only two control actions: pushing the cart left or right with a force of fixed magnitude. The problem is thus one of bang-bang control.

In contrast to these approaches, we attempt to solve the problem through the use of symbolic problem-solving techniques, employing a fuzzy rule-based controller with approximate reasoning. Furthermore, a continuous control scheme is employed, namely, the controller can apply a force with a magnitude within [-10,+10] newtons. In this way, better performance of the controlled system may be achieved but the complexity of the problem is increased substantially. An overlapping partition of the state space forms a linguistic space. The overlapping partition enhances the speed of learning and robustness. We will have more to say about these issues at a later point.

IV. The Self-Learning Rule-Based Controller

In this section, we present the theory of the proposed intelligent controller, which has the capability to learn its own control policy for balancing the pole. Figure 3 shows a schematic representation of the intelligent controller configured for the pole balancing task. It consists of a neural net and a rule-based controller employing approximate reasoning. The neural net has only two neuronlike elements, namely, an associative critic neuron (ACN) and an associative learning neuron (ALN). These two elements contribute the learning component of the controller. The rule-based component has four elements: a rule base, a fuzzy decoder, decision-making logic, and a defuzzifier. In general, the rule base is a collection of fuzzy conditional statements which take the form of if-then rules. The fuzzy decoder inspects the incoming system state and fires the rules in parallel. The decision-making logic emulates human decision-making behavior by employing approximate reasoning. The defuzzifier takes a fuzzy control decision from the decision making logic and determines a crisp, i.e., a non-fuzzy control action for the cart-pole system.

In more detail, our system has a configuration similar to Barto's [1], but it differs on the component level as well as the functional level. In principle, it is a rule-based controller composed of a linguistic description of the control strategy, which is self-learned. This strategy takes the form of fuzzy conditional statements, e.g., *if the angle of the pole is positive large and the angular velocity is positive large, then the force is positive large.* The fuzzy conditional statements are also referred to as *fuzzy control rules.* These are implemented and manipulated by using fuzzy set theory [17,20,18]. The inference mechanism of the intelligent controller uses approximate reasoning [9]. The learning capability of the controller is provided by two neuronlike elements. The ACN is derived by using Pavlovian conditioning theory [11,10]. The proposed ACN model also provides a basis for understanding and explaining Pavlovian conditioning in animal learning studies [10]. The ALN derives from the Instrumental Conditioning Theory [11]. It is an associative memory system which remembers the temporal relationships between input and output. More information about the two neuronlike elements will be presented at a later point.

A. A Rule-Based Controller Employing Approximate Reasoning

In recent years, rule-based controllers employing approximate reasoning have emerged as one of the most active areas of research in the applications of fuzzy set theory [12,9]. Such reasoning [19] plays an essential role in the remarkable human ability to make rational decisions in an environment of uncertainty and imprecision. In essence, approximate reasoning is the process or processes by which a possibly imprecise conclusion is deduced from a collection of imprecise premises. By employing the techniques of fuzzy set theory [17], approximate reasoning (with precise reasoning viewed as a limiting case) can be studied in a formal way.

The concept of a fuzzy set may be viewed as an extension of an ordinary (crisp) set. In a fuzzy set, an element can be a member of the set with a degree of membership varying between 0 and 1. Thus, a fuzzy set F in a universe $U = \{u_i, i=1, ..., n\}$ is defined by its membership function $\mu_F : U \rightarrow [0,1]$. If the $\mu_F(u_i)$ are 0 or 1, the fuzzy set is an ordinary set. As a special case, a fuzzy singleton is a fuzzy set containing just one element with degree 1.

A concept which plays an important role in the applications of the theory of fuzzy sets is that of a *linguistic variables* [19]. To illustrate, if *speed* is interpreted as a linguistic variable, that is, a variable whose values are linguistic labels of fuzzy sets, then the values of *speed* might be

$$T(speed) = \{slow, moderate, fast, very\ slow, more\ or\ less\ fast, \cdots\}.$$

In a particular context, *slow* may be interpreted as, say, "a speed below about 40 mph", *moderate* as "a speed close to 55 mph" and *fast* as "a speed above about 70 mph". Figure 4 shows this interpretation in the framework of fuzzy sets.

The set-theoretic operations on fuzzy sets are defined via their membership functions. More specifically, let A and B be two fuzzy sets in U with membership functions μ_A and μ_B, respectively. The membership function $\mu_{A \cup B}$ of the *union* $A \cup B$ is defined pointwise for all $u \in U$ by

$$\mu_{A \cup B}(u) = \max\ \{\mu_A(u), \mu_B(u)\}.$$

Dually, the membership function $\mu_{A \cap B}$ of the *intersection* $A \cap B$ is defined pointwise for all $u \in U$ by

$$\mu_{A \cap B}(u) = \min\ \{\mu_A(u), \mu_B(u)\}.$$

If $A_1, \ldots, A_n$ are fuzzy sets in $U_1, \ldots, U_n$, respectively, the *Cartesian product* of $A_1, \ldots, A_n$ is a fuzzy set in the product space $U_1 \times \cdots \times U_n$ with the membership function

$$\mu_{A_1 \times \cdots \times A_n}(u_1, u_2, \cdots, u_n) = \min\ \{\mu_{A_1}(u_1), \cdots, \mu_{A_n}(u_n)\}.$$

Assume that the fuzzy sets A, A', B, and B' are the linguistic values of x and y. An example of approximate reasoning involving x and y is the following:

premise 1 : *x is* A',

premise 2 : *if x is A then y is B*,

consequent: *y is* B'.

For instance:

premise 1 : *the speed of a car is very high*,

premise 2 : *if the speed of a car is high then the probability of an accident is high*,

consequent: *the probability of an accident is very high.*

This type of fuzzy inference is based on the compositional rule of inference for approximate reasoning suggested by Zadeh [19].

A rule-based controller consists of a set of fuzzy control rules which are processed through the use of approximate reasoning. For simplicity, suppose that we have the two rules:

$$R_1 : \textit{ if } x \textit{ is } A_1 \textit{ and } y \textit{ is } B_1 \textit{ then } z \textit{ is } C_1,$$

or

$$R_2 : \textit{ if } x \textit{ is } A_2 \textit{ and } y \textit{ is } B_2 \textit{ then } z \textit{ is } C_2.$$

Approximate reasoning, given (x is A') and (y is B'), computes the degree of partial match between the user-supplied facts and the knowledge rule base as follows.

The degrees of match of (A_i and A) and (B_i and B) are given respectively by

$$\alpha_i = \max_u \min\{\mu_{A_i}(u), \mu_A(u)\},$$

$$\beta_i = \max_v \min\{\mu_{B_i}(v), \mu_B(v)\}$$

The firing strength of the i^{th} rule is given by

$$x_i = \min\{\alpha_i, \beta_i\}.$$

Hence, the i^{th} rule recommends a control decision as follows:

$$\mu_{C_i'}(w) = \min\{x_i, \mu_{C_i}(w)\}.$$

The consequences of multiple rules can be combined by a conflict-resolution process which treats the sentence connective *or* as a union operator. The combined consequence is then given by

$$\mu_C(w) = \max\{\mu_{C_1'}, \mu_{C_2'}\}.$$

The combination of consequences is illustrated in Figure 5.

In on-line processes, the states of a control system are essential to a control decision (action). The underlying data are usually obtained from sensors and are crisp. It may be necessary to convert these data into the form of fuzzy sets [8]. In practice, however, crisp data are frequently treated as fuzzy singletons. In this case, the corresponding inference mechanism is shown in Figure 6. Furthermore, in on-line control, the inference process should lead to a non-fuzzy control action. This necessitates the use of a *defuzzifier.* A defuzzifier can be implemented by using *max criterion, mean of maximum* or *center of gravity* algorithms [9]. The defuzzifier used here is employing the *mean of maximum* algorithm, which produces the mean value of all local control actions whose membership functions reach the maximum.

In what follows, the fuzzy control rules are assumed to be of the form

$$R_i : \textit{if } x \textit{ is } A_i \textit{ and } y \textit{ is } B_i \textit{ then } z \textit{ is } C_i, \quad i = 1, 2, \ldots, n \quad ,$$

where x, y, and z are linguistic variables representing the angle of the pole with respect to the vertical axis, angular velocity of the pole, and applied force, respectively; A_i, B_i, and C_i are the linguistic values (fuzzy sets) of the linguistic variables x, y, and z in their respective universes of discourse, [-12,+12] degrees, $\mathcal{R}$, and [-10, +10] newtons. The definitions of linguistic values A_i and B_i are shown in Figure 7 (a) and (b). The problem is to learn the linguistic values C_i, which take the form of triangles, defined on the control force universe [-10,+10] newtons. As shown in Figure 7 (c), the location of the vertex of such a triangle is to be learned, while the coordinates of the base are functions of the vertex location value, say in the extreme case, +/-2 newtons away from that vertex.

To summarize the ideas thus far discussed, Figure 7 (d) illustrates a 2-D linguistic state space. The x axis is θ with seven linguistic values; the y axis is $\dot{\theta}$ with three linguistic values. Thus, 7×3 fuzzy control rules are involved. Each fuzzy control rule corresponds to a fuzzy cell. The premise of a fuzzy control rule determines the cell's coordinates in the linguistic state space. The consequent of the rule is taken to be the content of the cell, which is to be learned by the proposed neurons, the ALN and ACN. Once a system input is sensed, the cells are fired in parallel. The fuzzy decoder takes the current state of the cart-pole system as an input and has n outputs (firing strengths) going to the ALN and ACN. Each output of the fuzzy decoder corresponds to a fuzzy cell. The activity of the output is the firing strength. The firing strength serves as an input to both the ALN and ACN, and is also used to compute the recommended control action in each rule (cell).

B. Learning with a Neural Net

As has been mentioned in Section II, the principal difficulty in the learning process is that the training information (failure signal) is very sparse. Many of the previously employed neural networks such as the Adaline, perceptrons, and Hopfield nets, are effective for the solution of supervised pattern classification problems. Although the basic idea underlying our approach is similar to Barto's [1], it departs from Barto's in some significant respects. More specifically, our network consists of the ALN and ACN which perform unsupervised learning. The ACN has to do with the criticism from the environment associated with the system state. The ALN takes the criticism and associates n fuzzy control actions with n fuzzy cells (the consequents of n fuzzy control rules). Since the ACN predicts the criticism at every time step, the ALN can continuously update itself before the failure signal occurs. This is the basis for the solution of the credit-assignment problem.

1. ACN

The ACN is derived from Pavlovian conditioning theory [10]. Pavlovian conditioning [11] was introduced by Pavlov in 1927. The best known example of Pavlovian conditioning comes from Pavlov's research on the conditioned reflex of salivation by dogs. Prior to conditioning, when a dog hears the sound of a bell, it pricks its ears. Then, when the food is presented to it, it salivates. If this sequence of events is repeated, the dog soon starts to salivate in reaction to the sound of the bell. In effect, the dog has been "conditioned" to react to the bell. Figure 8 illustrates the conditioning of a Pavlov dog. As can be seen, the sound of a bell can be used to predict the occurrence of salivation before the presence of food. This predictive relationship between food and the sound of a bell has important implications. Thus, the ACN captures this predictive nature of the Pavlovian conditioning.

The correspondence between Pavlovian conditioning and the behavior of our control system is as follows (see also Figure 9). Food corresponds to the evaluative feedback (failure signal). The salivation by reflex is equivalent to an *external reinforcement r(t)* with the value -1.0 if failure signal occurs, otherwise 0.0. The sound of a bell relates to the i^{th} fired fuzzy cell (fuzzy control rule) with firing strength x_i. The salivation resulting from the bell's sound is the *predictive reinforcement* $v_i(t)$ of the i^{th} fuzzy cell. It is worth noting that, in the extreme, the i^{th} rule with firing strength either 1.0 or 0.0 is the exact case of presence or absence of food in the conditioning of a Pavlov dog. In other words, our ACN operates in a continuous mode, which treats Pavlovian conditioning as a special case. In effect, the ACN attempts to predict the reinforcement $v_i(t)$ that can eventually be obtained from the environment by choosing a control action for that fuzzy cell.

As an extension of single-input-single-output analogy, multiple inputs in the ACN necessitate an output which is a weighted sum of the predictive reinforcements of all fired fuzzy cells. The weighted sum *p(t)* is the *total reinforcement* of all fired fuzzy cells at time t. Furthermore, an internal reinforcement $\hat{r}(t)$, the criticism, is generated as a temporal difference of the total predictive reinforcements.

As shown in Figure 3, the ACN has an external reinforcement input, $r(t)$, from the cart-pole system, n inputs, $x_i(t)$, $i=1, \ldots, n$, from corresponding fuzzy cells, and an output, $\hat{r}(t)$, as internal reinforcement signal (criticism) for the ALN and itself. The total reinforcement at time t is given by

$$p(t) = G(\sum_{i=1}^{n} v_i(t)x_i(t)),$$

where G could be a sigmoid-shaped function, identity function, mean of maximum algorithm or center of area algorithm. The associative learning rule for the i^{th} fuzzy cell is in part characterized by a local memory trace $\overline{x}_i(t)$ and the internal reinforcement $\hat{r}(t)$. The predictive reinforcement $v_i(t)$ of the i^{th} fuzzy cell (fuzzy rule, fuzzy system state) is updated by

$$v_i(t+1) = v_i(t) + \beta\hat{r}(t)\overline{x}_i(t),$$

where β is a positive learning-rate parameter. The local memory trace is defined by

$$\overline{x}_i(t+1) = \lambda\overline{x}_i(t) + (1-\lambda)|x_i(t)v_i(t)|,$$

where λ, $0 \le \lambda < 1$, is a trace-delay parameter. The trace takes the form of an exponential curve. It is strengthened by the degree of firing strength of the i^{th} fuzzy cell (fuzzy control rule) together with its current weight, and weakened if the rule is not fired. The trace thus keeps track of how long ago the i^{th} fuzzy rule fired and also how often it was fired. The internal reinforcement is calculated as

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

where γ, $0 \le \gamma < 1$, is a discount-rate parameter. The internal reinforcement serves as criticism, depending on a relative difference of $p(t)$ and $p(t-1)$. If the pole does not fall and $\gamma p(t) > p(t-1)$, then $r(t)=0$ and $\hat{r}(t)>0$, a reward is given. If the pole does not fall and $\gamma p(t) < p(t-1)$, then $r(t)=0$ and $\hat{r}(t)<0$, and a punishment is effected. The discount factor γ implies a bias for the condition in which $p(t)$ equals $p(t-1)$. More specifically, once the pole does not fall and keeps in the same state, a reward is given through the use of a discount factor. On the other hand, if the pole falls, then $p(t)=0$, $r(t)=-1$ and $\hat{r}(t)<0$, and a punishment is issued. If $p(t-1)$ fully predicts the occurrence of the failure, there is no punishment. As shown, a negative feedback mechanism is implicitly incorporated into the internal reinforcement.

The proposed ACN model might be viewed as an extension of the Sutton-Barto model [10]. More specifically, in the context of animal learning phenomena, a sigmoid-shaped acquisition curve is observed. This is not simulated in the Sutton-Barto model. In our model, it can be achieved by making a change in the associative strength proportional to the current associative strength [10]. It has been demonstrated by computer simulation that the ACN accounts for many phenomena observed in Pavlovian conditioning, such as a sigmoid-shaped acquisition curve, inter-stimulus interval effects, trace conditioning, and delay conditioning. A more detailed discussion of this aspect of our model is described elsewhere [10].

2. ALN

The ALN is derived from the Instrumental Conditioning Theory [11]. A simple example is teaching a dog to perform a trick. During training, if the dog does well, it is given a reward. If not, it is punished. After training, the dog has learned a new trick. The association of the dog's response and reinforcement has in effect been "conditioned". The correspondence between this conditioning and the ALN is as follows (see also Figure 10). A dog corresponds to the i^{th} fuzzy control rule with firing strength x_i. The response of the dog relates to the control force $(w_i f_i)$ of the i^{th} rule. The reinforcement as reward/punishment is equivalent to the internal reinforcement from the ACN. The ALN does the following: the i^{th} fuzzy control rule can produce correct control force of the i^{th} rule under the internal reinforcement from the ACN. In effect, the ALN is a content-addressable memory system which associates each fuzzy rule with an appropriate fuzzy control action.

As shown in Figure 3, the ALN has an internal reinforcement input, $\hat{r}(t)$, from the ACN, n inputs, $x_i(t)$, i=1, ..., n, from the fuzzy decoder, a control action input, $F(t)$, from the defuzzifer, and n associative weights w_i i=1, ..., n, as outputs for the rule base. Each associative weight $w_i(t)$ is transformed -- by using the concepts of dynamical normalization and fuzzification -- into a fuzzy set having the form of a triangle as described in the previous section. The location of the vertex of the triangle is given by

$$f_i(t) = H(w_i(t) + noise(t)), \quad i = 1, ..., n \ ,$$

where H is a dynamic sigmoid function which may be viewed as a dynamic normalization function and provides a continuous output within the range [-10,+10]. For the purpose of computer simulation, the following function is used:

$$H(x,t) = \begin{cases} \dfrac{10x}{T(t)+x} & x>0, \\ \varepsilon & x=0, \\ \dfrac{10x}{T(t)-x} & x<0. \end{cases}$$

where $T(t) = k_1 \max\limits_i w_i(t)$ is an offset-tuning parameter which determines the slope of the sigmoid-shaped curve; and k_1 is a constant. The associative learning rule for each $w_i(t)$ is

$$w_i(t+1) = w_i(t) + \bar{\alpha}(t)\hat{r}(t)e_i(t),$$

$$\bar{\alpha}(t) = \frac{\alpha k_2}{k_2 + t},$$

where $\bar{\alpha}$ is a dynamical positive learning-rate parameter with a initial value α and k_2 is a weight-freeze parameter. The weight-freeze parameter determines the decreasing rate of the dynamical learning rate $\bar{\alpha}$. $\hat{r}(t)$ is the criticism from the ACN. The associativity trace, $e_i(t)$, is given by

$$e_i(t+1) = \delta e_i(t) + (1-\delta)F(t)x_i(t),$$

where δ, $0 \leq \delta < 1$, is another trace-decay parameter. The associativity trace takes the form of an exponential and it remembers for how long and how often a fuzzy control rule has fired as well as what control action was taken at that time.

Figure 11 illustrates the signal flow of our proposed controller during a learning process. A brief summary of the interpretation of variables used here is also given in Table I. Once a system state $(\theta,\dot{\theta})$ is sensed, the set of fuzzy control rules (fuzzy cells) is fired in parallel. A set of firing strengths (x_i) is then generated. The firing strength is used to compute the recommended action in each rule and also serves as input to both ALN and ACN. The information about the system state is then fed into the two neuronlike elements by the set of firing strengths. More specifically, the firing status of each rule is encoded by its firing strength. The firing strength together with the predictive reinforcement (v_i, or desirability) of the i^{th} fuzzy rule (fuzzy system state) generates the local memory trace ($\bar{x}_i$, desirability trace) of the i^{th} fuzzy rule (cell). The total reinforcement, p, or equivalently, the desirability of all fired fuzzy cells, is computed based on the firing strength and the reinforcement (desirability) of each fuzzy rule (fuzzy cell, fuzzy system state). A non-fuzzy control action, F, is determined after the processes of inference combining and defuzzification. The control action, F, together with the firing strength, x_i, of each rule contributes the associativity trace, e_i, of each rule. After applying the control action to the plant, a goal evaluation, r, is made, which takes binary values. Based on the yes-no evaluation, the criticism, $\hat{r}$, which is a more informative evaluation, is generated. It plays an important role in the solution of the credit-assignment problem. The weights (v_i, w_i) in learning rules are thus updated on the basis of the criticism and their own local memory trace, ($\bar{x}_i$, e_i). A fuzzy control force in each rule is generated from the w_i by the use of dynamic normalization and fuzzification.

V. Simulation Results

We implemented our system as shown in Figure 3 on a Sun workstation. For comparison purposes, we also implemented Barto's system [1] for solving the same problem. The parameter values used in our simulation were: α=1000, β=0.5, γ=0.95, δ=0.9, λ=0.8, ε=0.1, k_1=0.015, and k_2=2500. The noise in the force-generating equation of the ALN is zero-mean Gaussian with a standard deviation of 0.01. The learning system was tested for 6 runs. Each run consisted of 60 trials. A run was called "success" whenever the number of steps before failure was greater than 500,000. The external reinforcement $r(t)$ was -1 when the failure signal occurred, otherwise, it was 0. Every trial began with the same initial cart-pole states, θ=0, $\dot{\theta}$=0, x=0, $\dot{x}$=0, and ended with a failure signal when $|\theta|$>12 degrees. All memory traces, x_i and e_i, were set to zero. In each run, the noise was provided by a Gaussian random number generator with varying seeds; all the weights, w_i; were set to zero, and a lower bound v_i (=-0.0001) was set to all the weights.

A. Learning / Training

Figure 12 and 13 illustrate the simulation results based on our proposed controller and Barto's system, respectively. In every case, both systems are capable of learning to balance the pole. However, experiments show that our system has a better learning performance. The proposed controller learns to balance the pole within 15 trials. Most of these cases involve less than 10 trials and are too close to be distinguished in Figure 12. The performance of Barto system varies around 50 trials. The Gaussian noise did have some effect on the Barto's system.

Additional observations were made on the state trajectory of the angle of the pole with respect to the vertical axis. We observed the data after the systems learned their own control strategy. The data showed that, in every case, our controller could keep the angle within a smaller region compared with Barto's. Figure 14 and 15 illustrate two sets of these data from our system and Barto's, respectively.

B. Adaptation

Adaptation is intended to adjust to unforeseen changes in environmental conditions using prior knowledge. Training involves constructing a knowledge base of an application domain (e.g. a pole-balancing task) with little a priori domain knowledge. The capability of learning to solve new tasks by modifying previous learned knowledge (adaptation) is compared with that of starting from scratch (training). Extensive simulation studies of such schemes have been carried out. They show that the proposed controller tolerates a wide range of uncertainty as well as a lack of system information, e.g., parameter changes in the length and mass of the pole, various initial states of the angle of the pole, changes of failure criteria, and a slanted cart-pole system.

The adaptation experiments were based on pre-learned knowledge by employing the same parameter settings as that in the last section. The length and mass of the pole were 0.1kg and

1.0m, the angle constraint for failure evaluation was -/+12°, and the initial value of the angle of the pole with respect to the vertical axis is 0.0°. The system took 6 trials to learn the task.

In the first set of experiments, the system is required to adapt to changes in the length and mass of the pole. Four experiments were done as shown in Table II. The first two were to increase the original mass of the pole by a factor of 10 and 20, respectively. The last two were to replace the original pole by two shorter poles. The length and mass of the first pole were reduced to 2/3 of the original values, while the second one is 1/3. Without pre-training, the system took 45, 208, 15, and 9 trials to learn these tasks. However, with the pre-trained knowledge, the system successfully completed these tasks without any further trials.

In the second set, we added a more severe constraint on the angle of the pole for failure evaluation as shown in Table III. The angle constraints were changed from +/-12° to +/-6°, and to +/-3°, respectively. The system needed 39 and 104 trials to learn these tasks with no initial knowledge. With pre-training, the system adapted to the first task without further trials and to the second one with 31 trials.

In the third set, the system was required to adapt to the changes in the length and mass of the pole (by a factor of 2/3) and angle constraint (+/–6°). In Table IV, training took 16 trials, while adaptation needed 4 more trials only (6 trials).

In the fourth set, we changed the initial states of the cart-pole system in five experiments as shown in Table V. The initial values (angle of the pole, angular velocity) were moved to (5.7°. 0). (10.0°. 0). (10°. 0), (5.3°, 17.19°/*sec*), and (–8.59°, 40.11°/*sec*), respectively. Without pre-training, the system needed 11, 11, 6, 11, and 13 trials to learn these two tasks, respectively. The system with pre-trained knowledge completed the tasks without further training except in one case.

Finally, the cart-pole system was lifted at the right end in such a way that the base of the system and the surface of the table formed an angle of 12°. The system took 35 trials to balance the pole. However, the system with the trained knowledge needed 8 more trials; to be more specific, it took 8+6 trials to complete the new task.

VI. Concluding Remark

In this article, we have proposed a symbolic problem-solving approach to a class of learning control problems. More specifically, we have attempted to develop an intelligent control scheme by integrating human decision-making (in terms of a rule-based controller using approximate reasoning) and animal learning behavior (in terms of two neuronlike elements). The proposed rule-based controller learns and improves its rule base for better control strategy from experience. In this way, we avoid an ad-hoc rule-tuning process which is usually inefficient and lacking in consistency. Computer simulation results show that the learning capability of our controller represents an improvement over previous approaches. This is achieved in part by employing approximate reasoning as the inference engine of the controller and by employing a continuous control scheme rather than bang-bang control. Furthermore, our controller is relatively insensitive to variations in the parameters of the system environment. In addition, the controller can be primed with pre-trained control knowledge which minimizes rapid changes during adaptation.

The approach described in this paper may be viewed as a step in the development of a better understanding of how to combine a fuzzy logic based system with a neural network to achieve a significant learning capability. We plan to address various aspects of this important issue in subsequent papers.

Acknowledgment

I am greatly indebted to Professor Lotfi A. Zadeh of the University of California, Berkeley for his encouragement of this research. The assistance of Professor Zadeh is gratefully acknowledged.

References

[1] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems". *IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-13, no. 5,* pp. 834-846, 1983.

[2] J. A. Bernard, "Use of a rule-based system for process control", *IEEE Control Systems Magazine, vol. 8, no. 5,* pp. 3-13, Oct. 1988.

[3] R. H. Cannon, *Dynamics of Physical Systems,* New York: McGraw-Hill, 1967.

[4] J. Carbonell and P. Langley "Learning, machine," in *The Encyclopedia of Artificial Intelligence, vol. 1* S. C. Shapiro, ed., New York: John Wiley and Sons, 1987, pp. 464-488.

[5] Fujitec, "FLEX-8800 series elevator group control system," *Fujitec Co., Ltd,* Osaka, Japan, 1988.

[6] Y. Kasai and Y. Morimoto, "Electronically controlled continuously variable transmission," *Proc. Int. Congress on Transportation Electronics,* Dearborn, Michigan, Oct. 1988.

[7] M. Kinoshita, and T. Fukuzaki, T. Satoh, and M. Miyake, "An automatic operation method for control rods in BWR plants," *Proc. Specialists' Meeting on In-core Instrumentation and Reactor Core Assessment,* Cadarache, France, 1988.

[8] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, Part I," *to appear in IEEE Trans. on Systems, Man, and Cybernetics.*

[9] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, Part II," *to appear in IEEE Trans. on Systems, Man, and Cybernetics.*

[10] C. C. Lee "Modeling behavior substrates of associative learning and memory: adaptive neuronal models," *UC, Berkeley Technical Report UCB/CSD-89/495,* Feb. 1989.

[11] N. J. Mackintosh, *The Psychology of Animal Learning,* New York: Academic Press, 1974.

[12] J. Maiers and Y. S. Sherif, "Applications of fuzzy set theory," *IEEE Trans. on Systems, Man, and Cybernetics vol. SMC-15, no. 1,* pp. 175-189, 1985.

[13] D. Michie and R. A. Chambers, "Boxes: an experiment in adaptive control," *in Machine Intelligence 2,* Oliver and Boyd, Edinburgh, 1968, pp. 137-152.

[14] Nissan, "New auto systems use fuzzy logic," *New York Times,* July 1, 1989.

[15] S. Yasunobu and G. Hasegawa, "Evaluation of an automatic container crane operation system based on predictive fuzzy control," *Control Theory and Advanced Technology, Vol. 2, no. 3,* 1986.

[16] S. Yasunobu and S. Myamoto, "Automatic train operation by predictive fuzzy control," in *Industrial Applications of Fuzzy Control,* M. Sugeno, ed., Amsterdam: North Holland, 1985.

[17] L. A. Zadeh, "Fuzzy sets," *Information and Control, vol. 8,* pp. 338-353, 1965.

[18] L. A. Zadeh, "A rationale for fuzzy control," *Trans. ASME, J. Dynamic Systems, Measurement, and Control, vol. 94* pp. 3-4, 1972

[19] L. A. Zadeh, "Outline of a new approach to the analysis complex systems and decision processes," *IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-3,* pp. 28-44, 1973.

[20] H. J. Zimmermann, *Fuzzy Sets, Decision Making, and Expert Systems,* Boston: Kluwer Academic Publisher, 1987.

Figure Captions

Fig. 1. A schematic representation of a self-learning rule-based controller.

Fig. 2. The cart-pole system.

Fig. 3. The proposed intelligent controller configured with the pole-balancing component.

Fig. 4. Diagrammatic representation of various linguistic values of speed.

Fig. 5. Diagrammatic representation of approximate reasoning using a fuzzy input.

Fig. 6. Diagrammatic representation of approximate reasoning using a crisp input.

Fig. 7. (a) Linguistic values of the angle, (b) angular velocity, and (c) applied force. (d) A two-dimensional linguistic state space.

Fig. 8. A schematic paradigm of Pavlovian conditioning procedure.

Fig. 9. The correspondence of Pavlovian conditioning and the ACN.

Fig. 10. The correspondence of Instrumental Conditioning and the ALN.

Fig. 11. The signal flow of the proposed intelligent controller.

Fig. 12. Learning performance based on the proposed intelligent controller.

Fig. 13. Learning performance based on Barto's system.

Fig. 14. (a) State performance of the pole angle based on the proposed controller. (b) State performance of the pole angle based on Barto's system.

Fig. 15. (a) State performance of the pole angle based on the proposed controller. (b) State performance of the pole angle based on Barto's system.

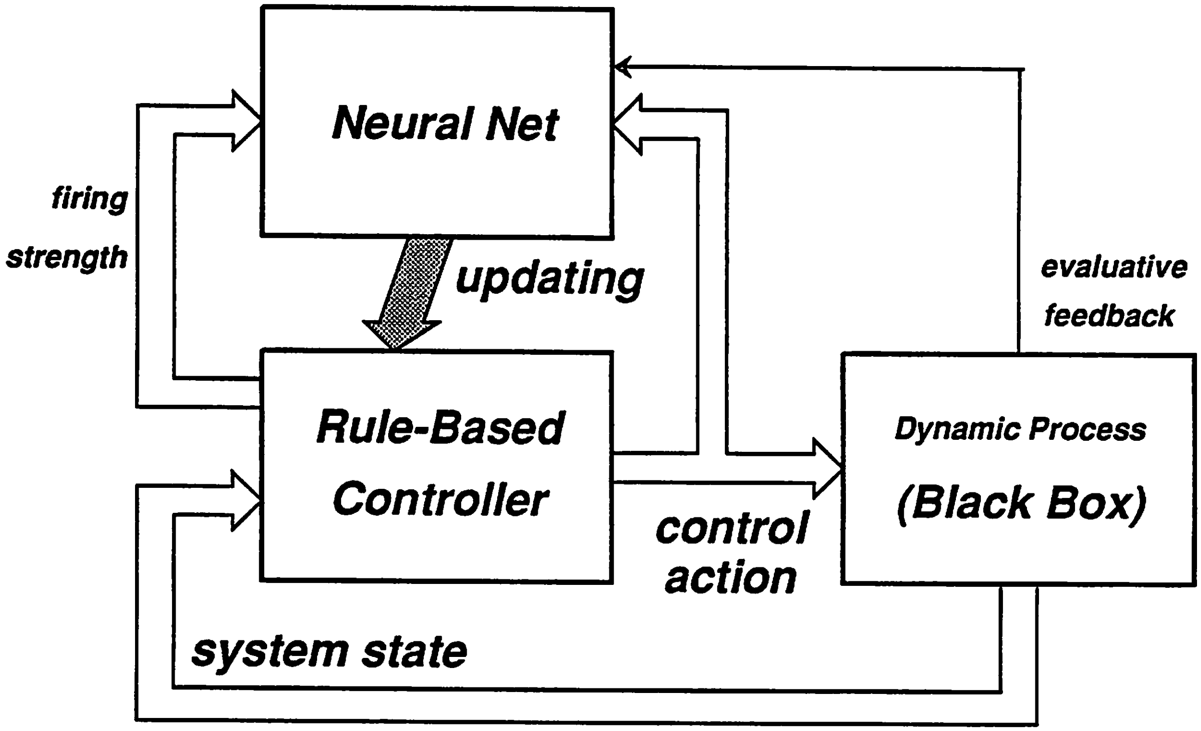

Neural Net
firing
strength
updating
evaluative
feedback
Rule-Based
Controller
Dynamic Process
(Black Box)
control
action
system state

Fig. 1

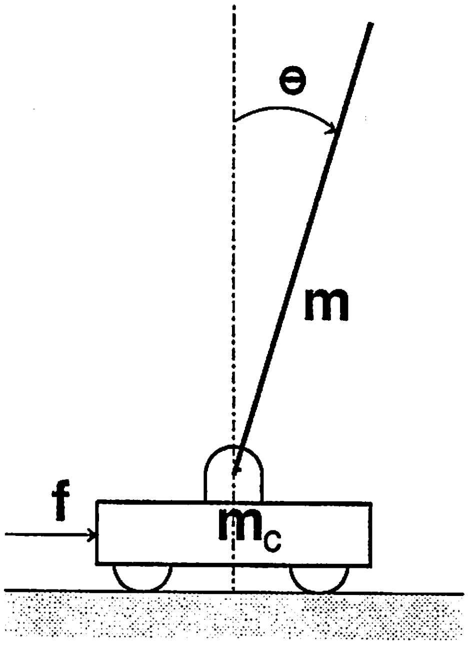


Fig. 2

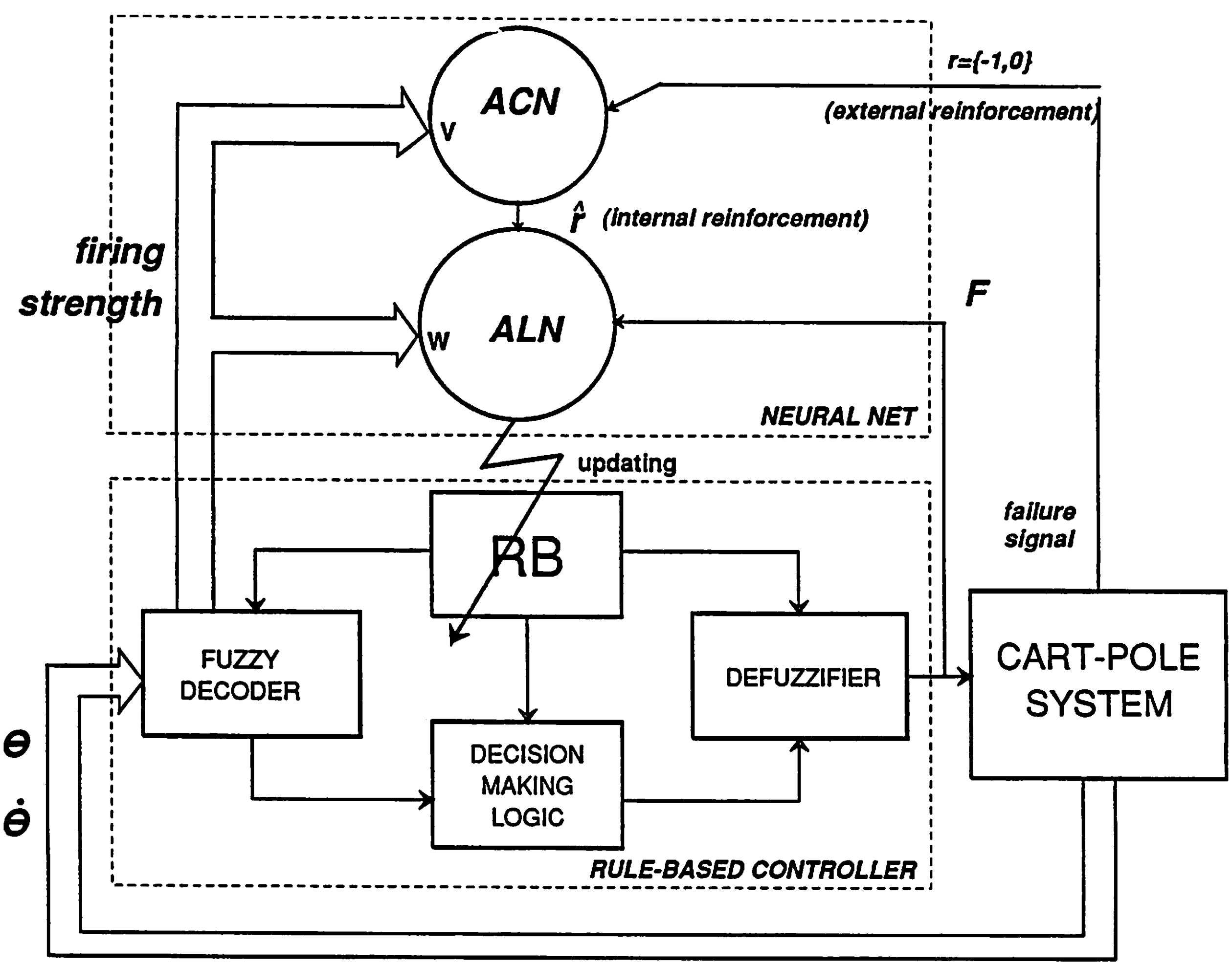

ACN
V
r={-1,0}
(external reinforcement)
$\hat{r}$ (internal reinforcement)
firing
strength
ALN
W
F
NEURAL NET
updating
failure
signal
RB
FUZZY
DECODER
DEFUZZIFIER
CART-POLE
SYSTEM
θ
$\dot{\theta}$
DECISION
MAKING
LOGIC
RULE-BASED CONTROLLER

Fig. 3

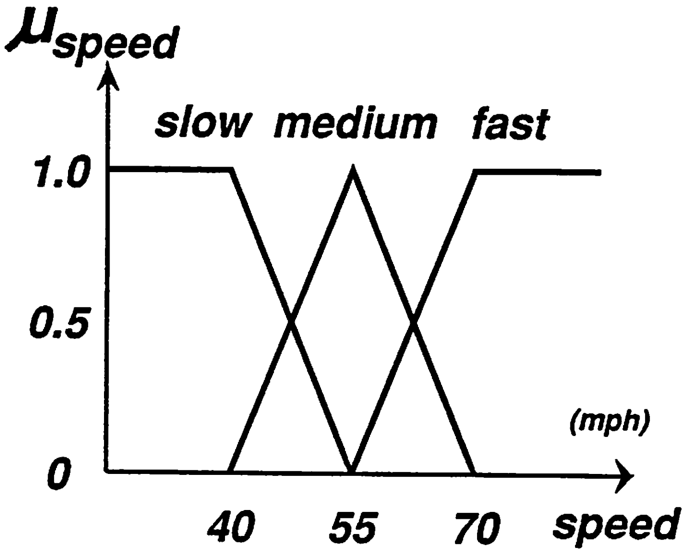


Fig. 4

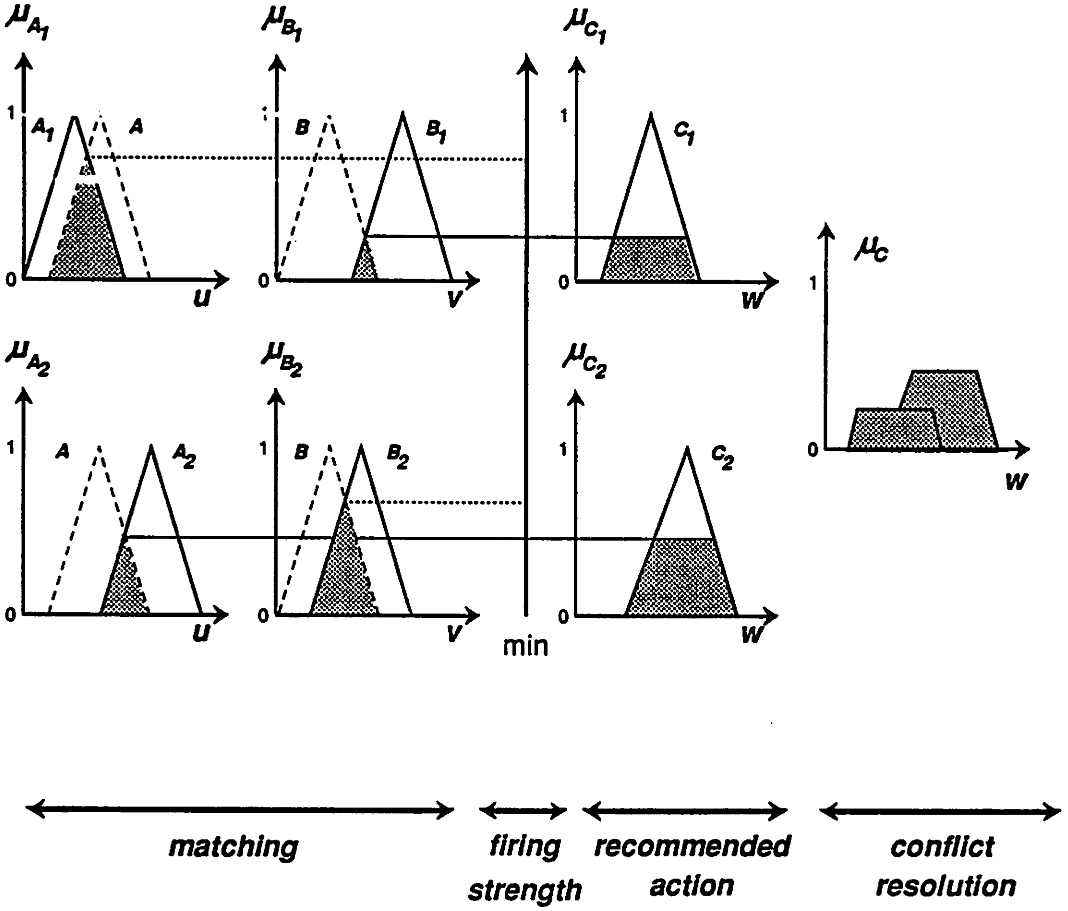

μ_{A_1}
1
A_1
A
0
u
μ_{B_1}
B
B_1
0
v
μ_{C_1}
1
C_1
0
w
μ_C
1
0
w
μ_{A_2}
1
A
A_2
0
u
μ_{B_2}
1
B
B_2
0
v
min
μ_{C_2}
1
C_2
0
w
matching
firing strength
recommended action
conflict resolution

Fig. 5

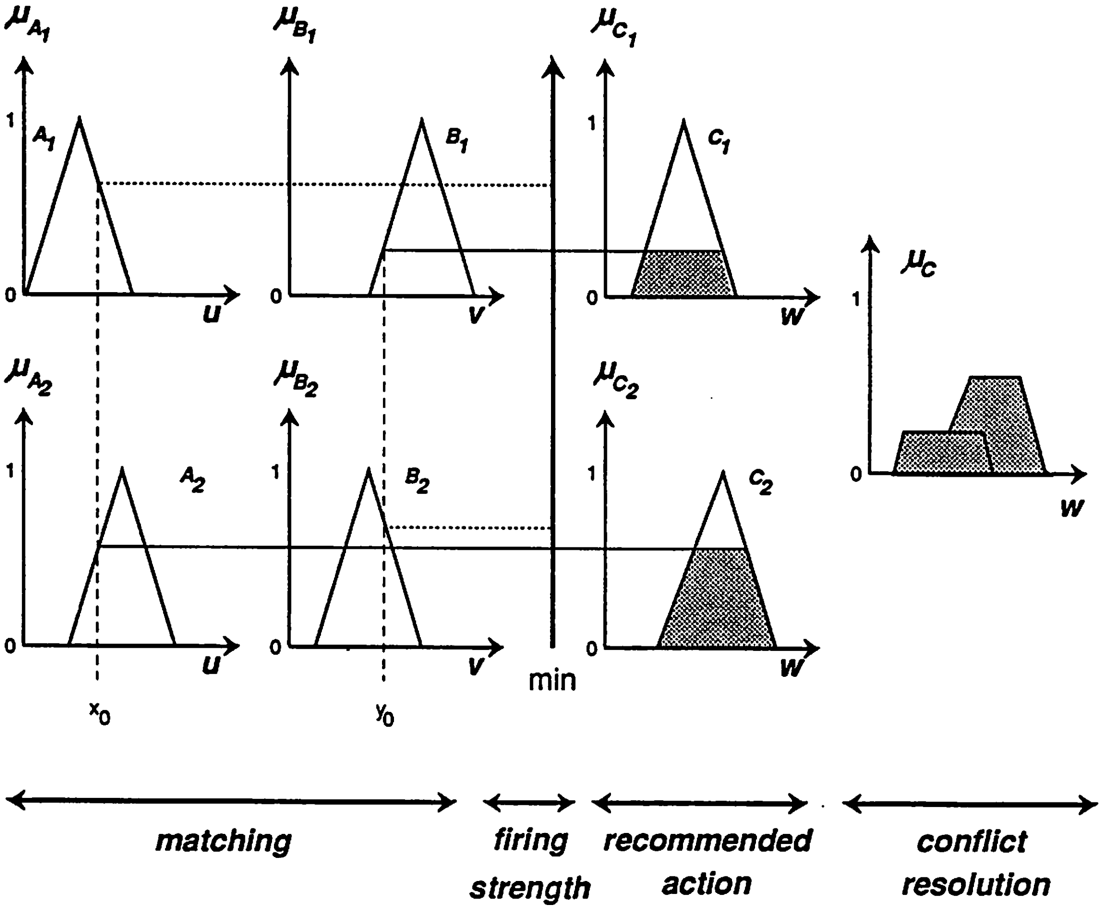
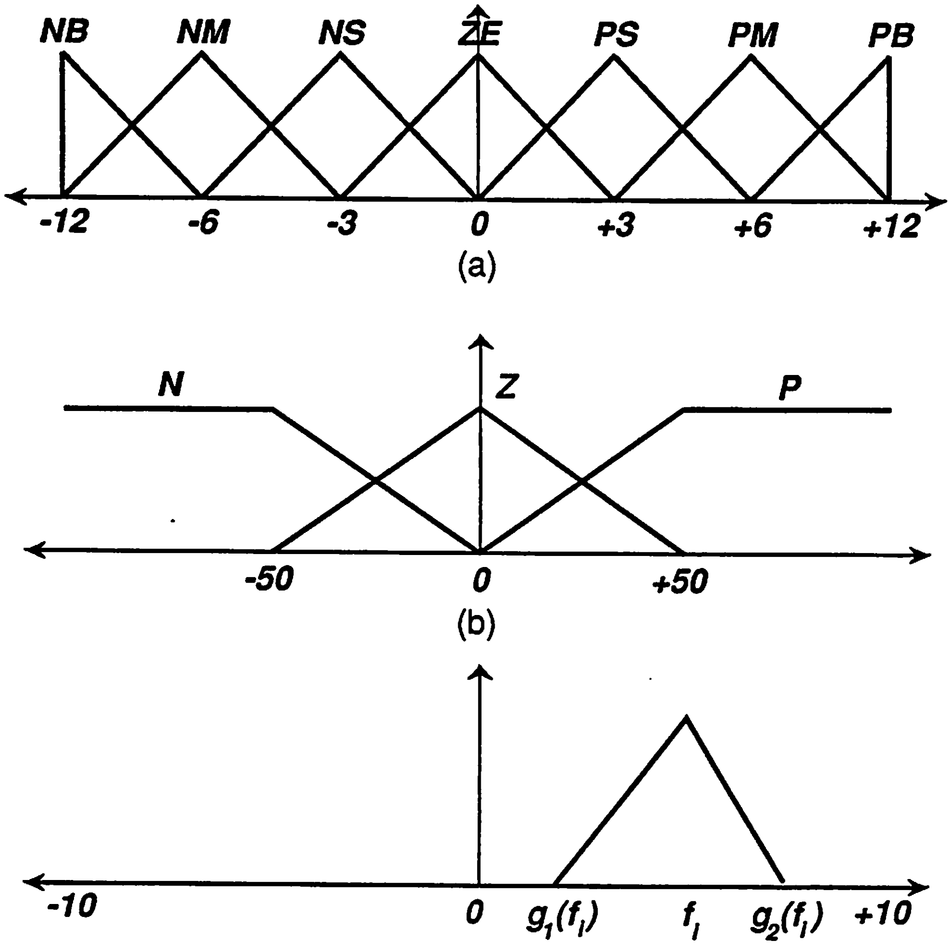

μ_{A_1}
μ_{B_1}
μ_{C_1}
1
A_1
B_1
C_1
0
u
0
v
0
w
μ_C
1
0
w
μ_{A_2}
μ_{B_2}
μ_{C_2}
1
A_2
1
B_2
1
C_2
0
u
0
v
min
0
w
x_0
y_0
matching
firing strength
recommended action
conflict resolution

Fig. 6

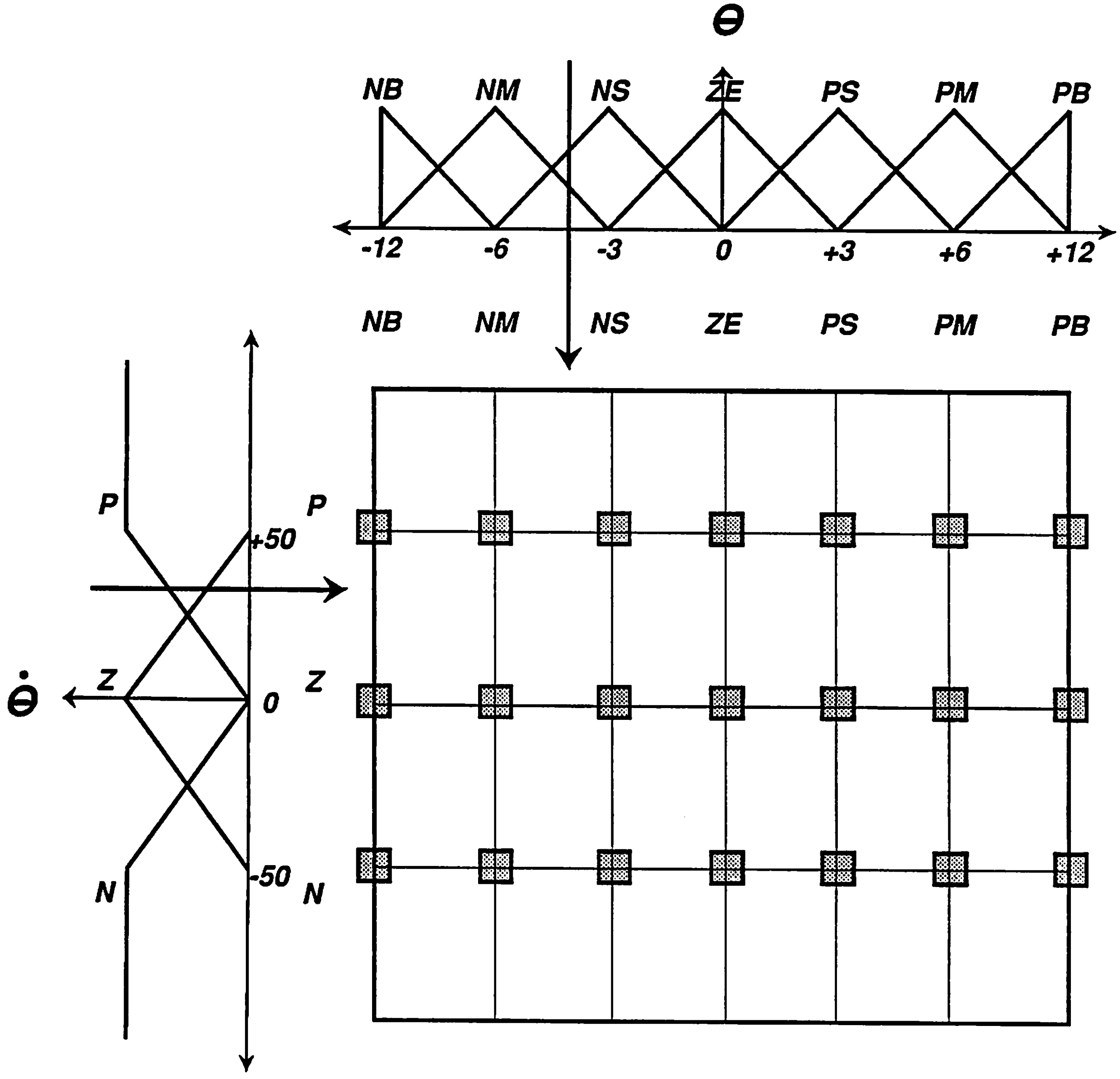


(a)

(b)

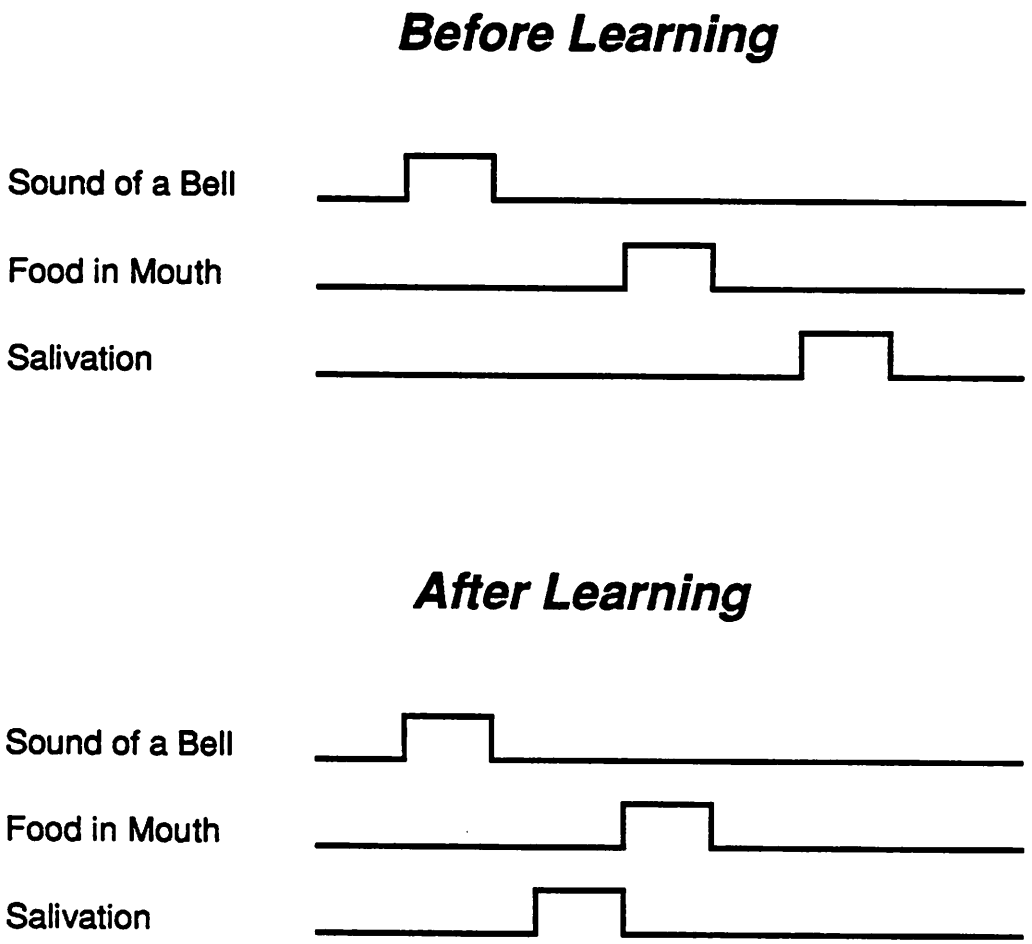
(c)

Fig. 7


Θ
NB
NM
NS
ZE
PS
PM
PB
-12
-6
-3
0
+3
+6
+12
NB
NM
NS
ZE
PS
PM
PB
P
P
+50
Θ̇
Z
0
Z
-50
N
N

(Θ X Θ̇ , F)

Fig. 7 (d)



Before Learning

After Learning

Fig. 8

Analogy

food in mouth	↔	*failure signal*
sound of a bell	↔	*ith fired fuzzy control rule* *(firing strength)*
salivation *(by reflex)*	↔	*external reinforcement (-1.0)* *(by failure signal)*
salivation *(by bell sound)*	↔	*prediction of reinforcement*

Fig. 9

Analogy

a dog ⟷ *ith fuzzy control rule (firing strength)*

response of a dog ⟷ *control force of ith rule* (w_i / f_i)

reinforcement (candy/beating) ⟷ *internal reinforcement* $(\hat{r})$

Fig. 10

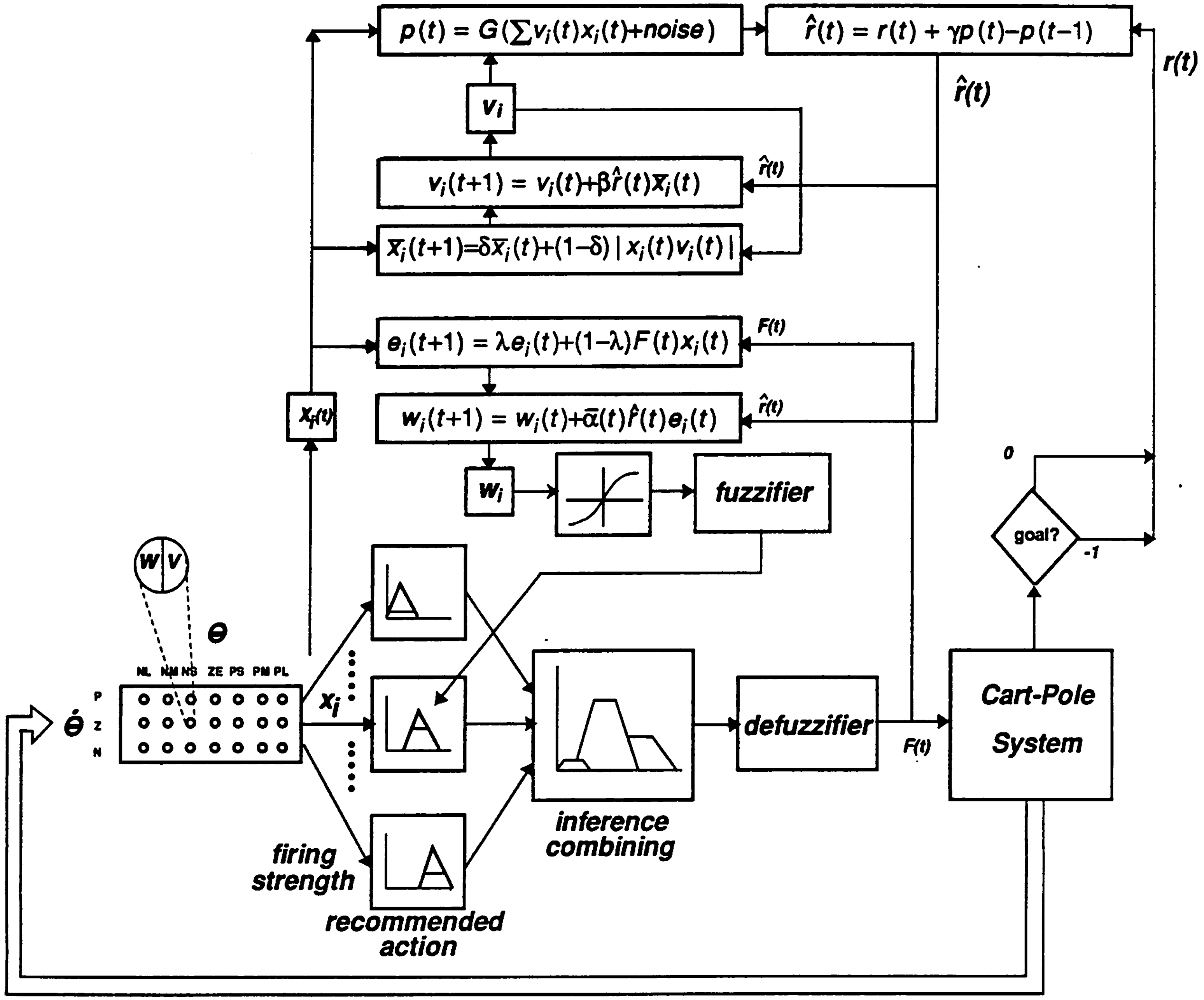

$p(t) = G(\sum v_i(t)x_i(t)+noise)$
$\hat{r}(t) = r(t) + \gamma p(t)-p(t-1)$
$r(t)$
$\hat{r}(t)$
v_i
$v_i(t+1) = v_i(t)+\beta\hat{r}(t)\bar{x}_i(t)$
$\hat{r}(t)$
$\bar{x}_i(t+1)=\delta\bar{x}_i(t)+(1-\delta)\,|\,x_i(t)v_i(t)\,|$
$e_i(t+1) = \lambda e_i(t)+(1-\lambda)F(t)x_i(t)$
$F(t)$
$x_i(t)$
$w_i(t+1) = w_i(t)+\bar{\alpha}(t)\hat{r}(t)e_i(t)$
$\hat{r}(t)$
w_i
fuzzifier
0
goal?
-1
W V
θ
NL NM NS ZE PS PM PL
P
Z
N
$\dot{\theta}$
x_i
defuzzifier
$F(t)$
Cart-Pole System
inference combining
firing strength
recommended action

Fig. 11

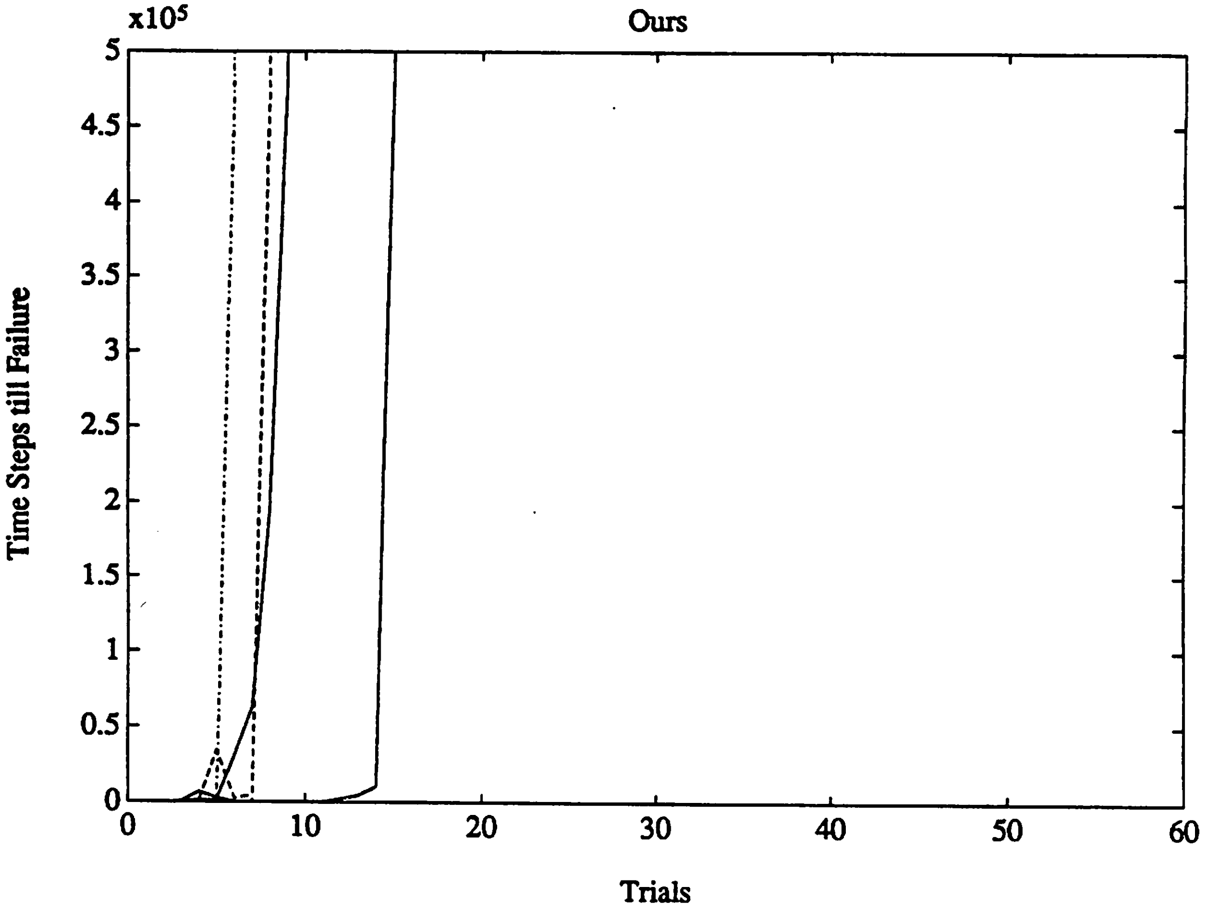

Ours
x10^5
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
Time Steps till Failure
0
10
20
30
40
50
60
Trials

Fig. 12

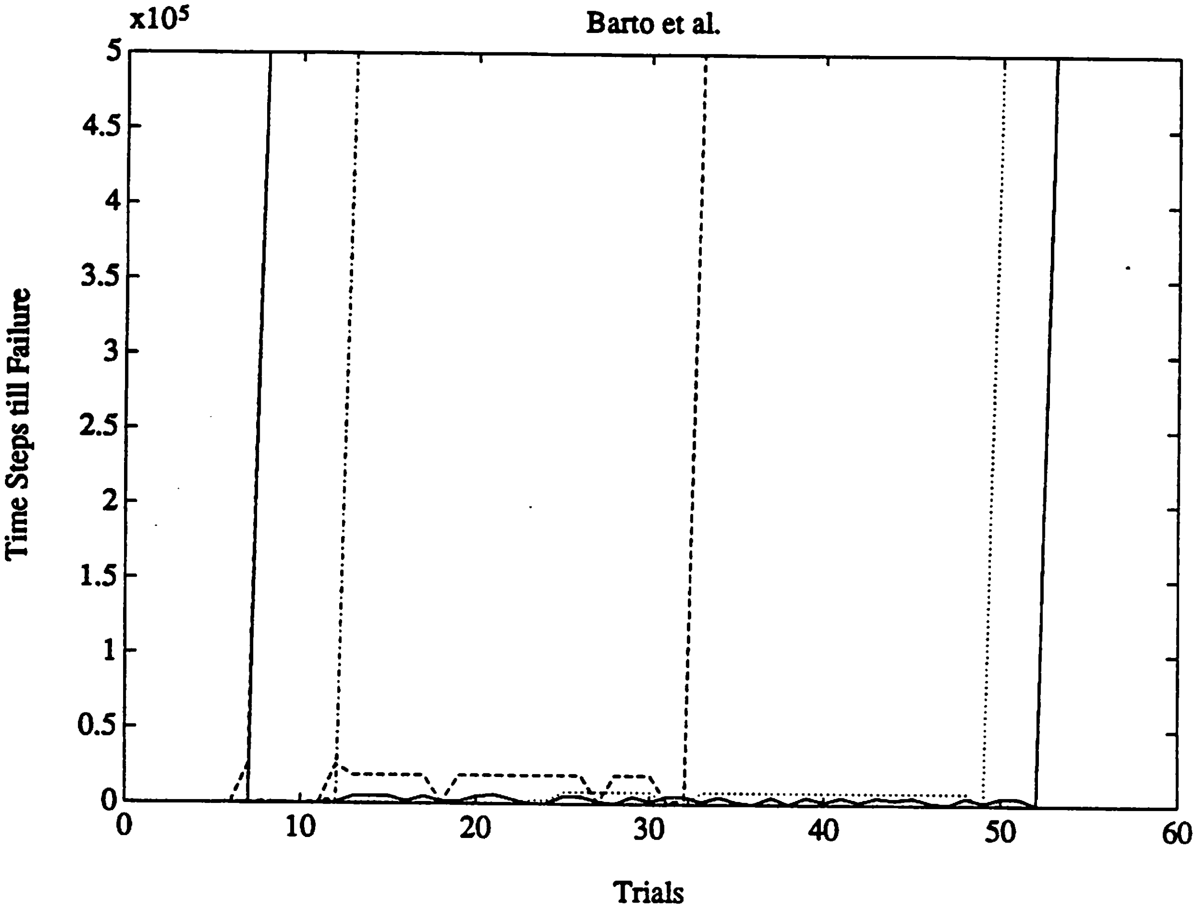

x10^5
Barto et al.
Time Steps till Failure
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
10
20
30
40
50
60
Trials

Fig. 13

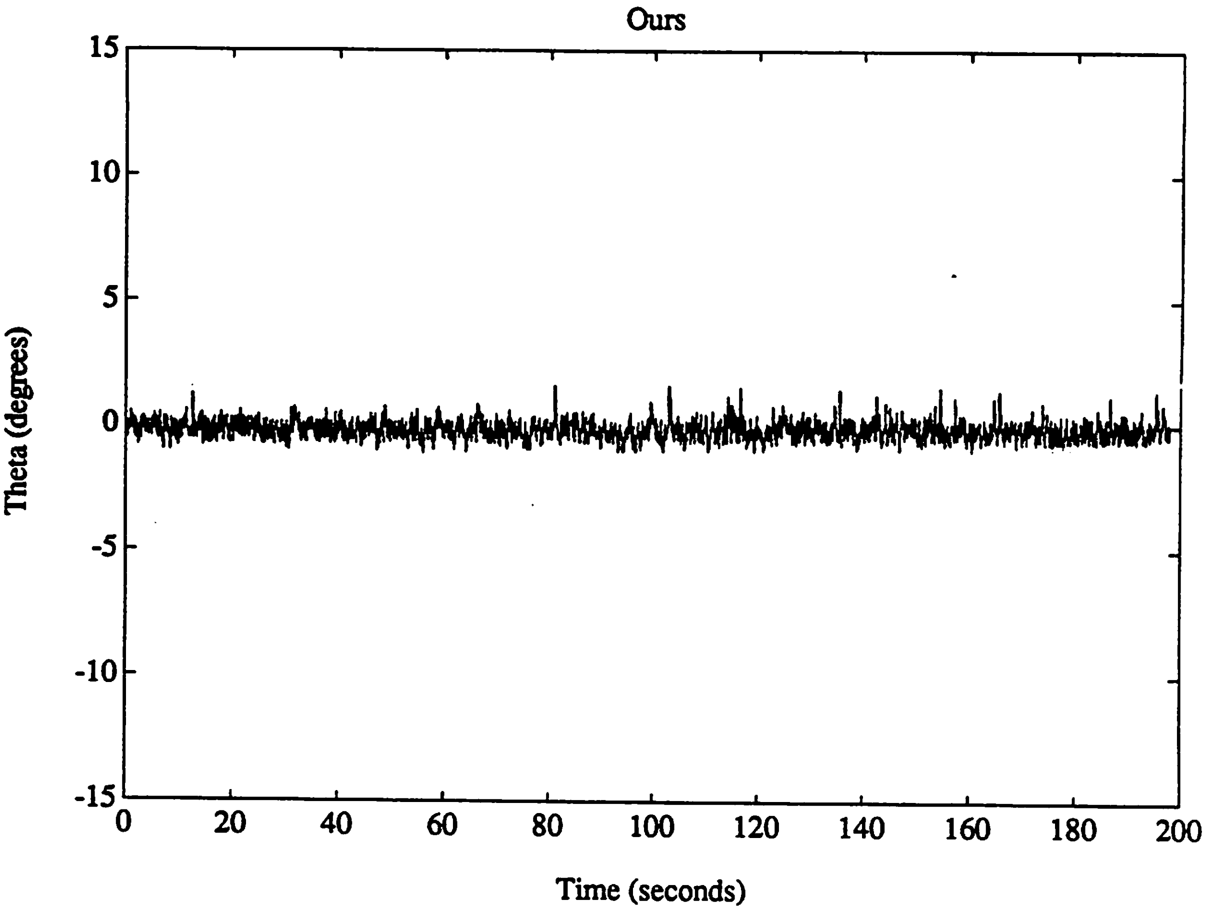

Ours
15
10
5
0
-5
-10
-15
Theta (degrees)
0
20
40
60
80
100
120
140
160
180
200
Time (seconds)

Fig. 14 (a)

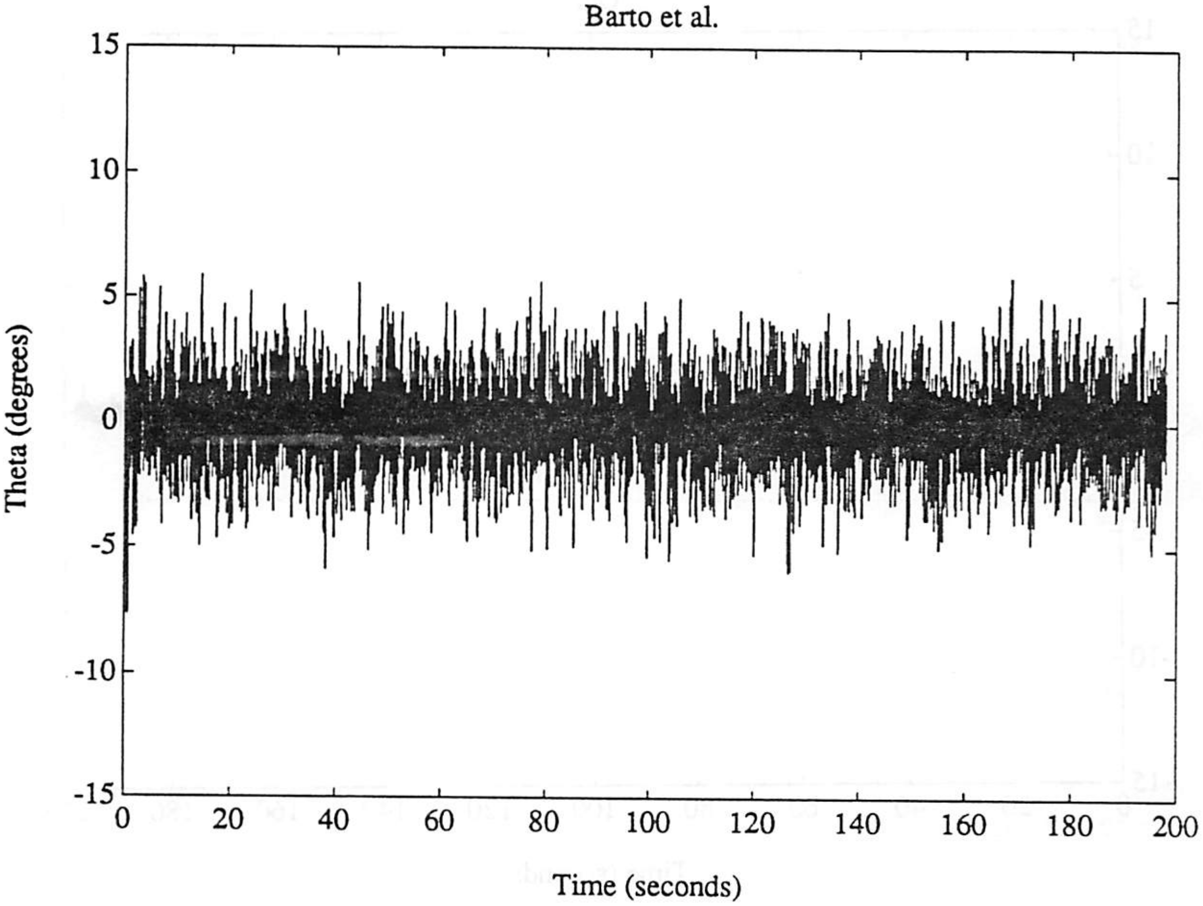

Barto et al.
15
10
5
0
-5
-10
-15
Theta (degrees)
0
20
40
60
80
100
120
140
160
180
200
Time (seconds)

Fig. 14 (b)

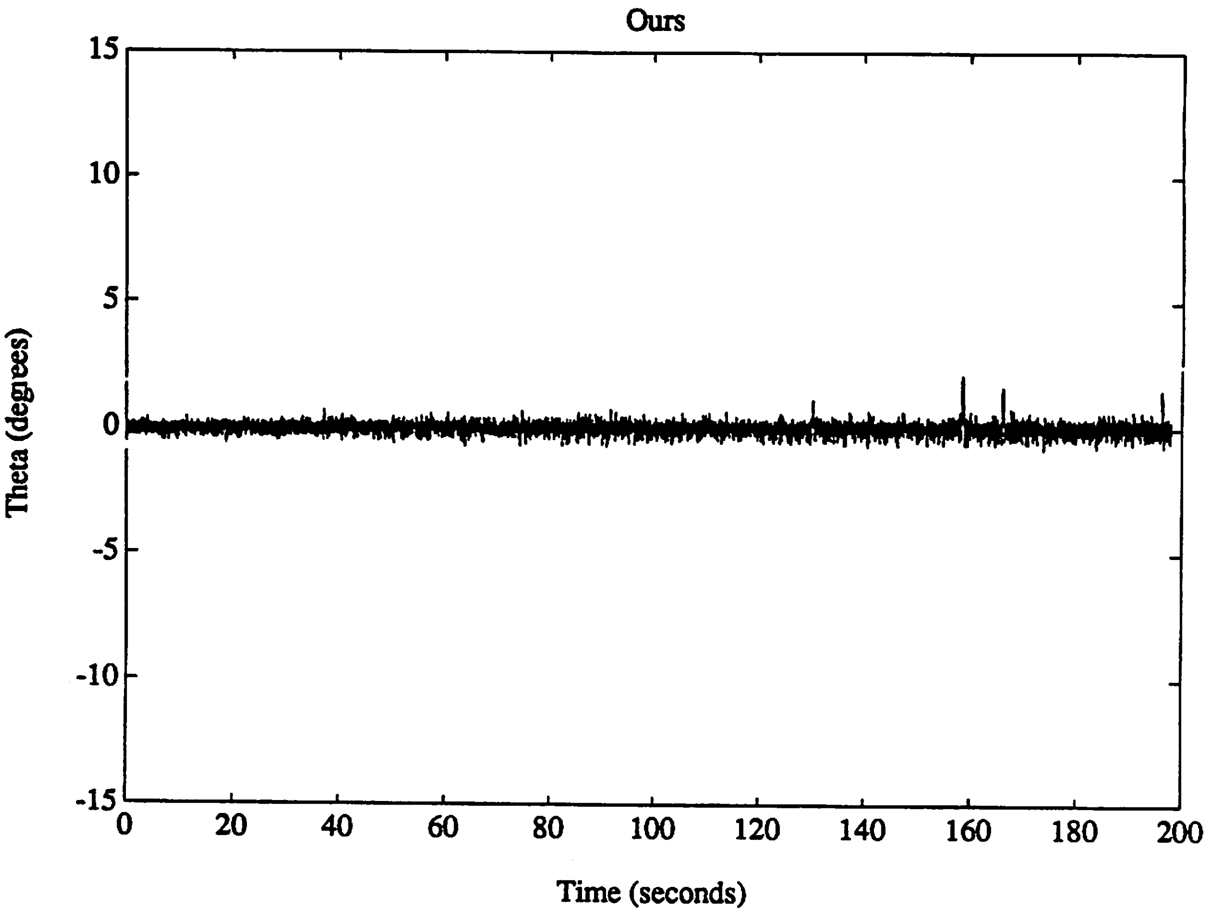

Ours
15
10
5
0
-5
-10
-15
Theta (degrees)
0
20
40
60
80
100
120
140
160
180
200
Time (seconds)

Fig. 15 (a)

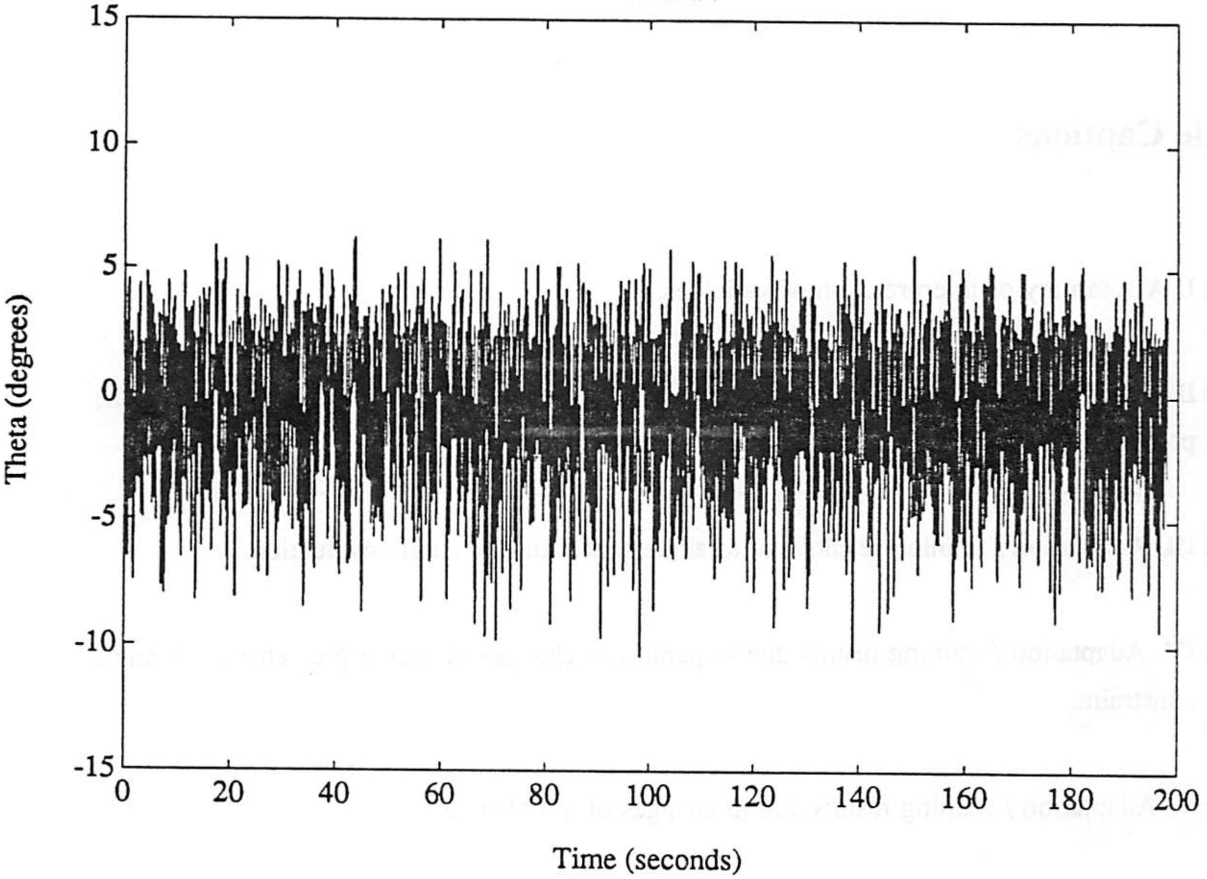

15
10
5
0
-5
-10
-15
Theta (degrees)
0
20
40
60
80
100
120
140
160
180
200
Time (seconds)

Fig. 15 (b)

Table Captions

Table I. A summary of interpretation of variables.

Table II. Adaptation / training results due to parameter changes of the length and mass of the pole.

Table III. Adaptation / training results due to angle constraint for failure evaluation.

Table IV. Adaptation / training results due to parameter changes of a pole plus change of angle constraint.

Table V. Adaptation / training results due to changes of initial state.

Table I
A summary of variables

subindex i: the ith fuzzy cell (fuzzy control rule, fuzzy system state).

$\theta(t), \dot{\theta}(t)$: angle of the pole and angular velocity.
$F(t)$: control force applied to the plant.
$x_i(t)$: firing strength at time t.
$r(t)$: goal evaluation, which takes binary values.

$V_i(t)$: predictive reinforcement (desirability) of the i^{th} fuzzy system state.
$\bar{x}_i(t)$: desirability trace which is a function of $x_i(t)$, $v_i(t)$ and $\bar{x}_i(t-1)$.
$p(t)$: total reinforcement (desirability) of all fired fuzzy cells at time t.
$\hat{r}(t)$: criticism for current motion.
G: a sigmoid-shaped function.

$w_i(t)$: control force before dynamic normalization and fuzzification.
$f_i(t)$: control force before normalization.
$e_i(t)$: associativity trace which is a function of $F(t)$, $x_i(t)$ and $e_i(t-1)$, keeping track of firing frequency and history of the i^{th} fuzzy cell and control action.
$\bar{\alpha}(t)$: dynamical learning rate.

Table II

parameter changes of
the length and mass of pole

Pole		*Adaptation*	*Training*
mass	*length*	*no. of additional trials*	*from scratch*
10x	*same*	*0*	*45*
20x	*same*	*0*	*208*
(2/3)x	*(2/3)x*	*0*	*15*
(1/3)x	*(1/3)x*	*0*	*9*

Table III

angle constraint
for failure evaluation
(originally, -/+12)

Angle Constraint	*Adaptation* *no. of add. trials*	*Training* *from scratch*
+/-6	*0*	*39*
+/-3	*31*	*104*

Table IV

parameter changes of a pole
plus change of angle constraint

Pole		*Angle*	*Adaptation*	*Training*
mass	*length*	*Constraint*	*no. of add. trials*	*from scratch*
(2/3)x	*(2/3)x*	*+/-6*	*4*	*16*

Table V

changes of initial state

(started with 0)

Initial State		*Adaptation*	*Training*
angle	*ang. vel.*	*no. of add. trials*	*from scratch*
5.73	0.0	0	11
-8.59	0.0	0	11
10.0	0.0	1	6
5.30	17.19	0	11
-8.59	40.11	0	13