Smooth Trajectory Tracking of Three-Link Robot: A Self-Organizing CMAC Approach

Kao-Shing Hwang, Member, IEEE, and Ching-Shun Lin

Abstract—A neuro-fuzzy system which is embedded in the conventional control theory is proposed to tackle physical learning control problems in this paper. The control scheme is composed of two elements. The first element, the fuzzy sliding mode controller (FSMC), is used to drive the state variables to a specific switching hyperplane or a desired trajectory. The second one is developed based on the concept of the self-organizing fuzzy cerebellar model articulation controller (FCMAC) and adaptive heuristic critic (AHC). Both compose a forward compensator to reduce the chattering effect or cancel the influence of system uncertainties. A geometrical explanation on how the FCMAC algorithm works is provided and some refined procedures of the AHC are presented as well. Simulations on smooth motion of three-link robot is given to illustrate the performance and applicability of the proposed control scheme.

I. INTRODUCTION

Many conventional control theories have successfully dealt with a large class of control problems by generating control efforts based on analytical models. However, analytical models tend to become complex and nonlinear so that they can’t be solved by conventional methods. Besides, most of the industrial processes whose reference models are unavailable due to the system noises and environmental disturbances. An alternative way to deal with the problem is to combine the conventional control theory with the fuzzy logic and neural networks. The incentive to combining these different strategies is to develop a more effective control approach for the systems with the characteristics of nonlinearity and uncertainties.

The sliding mode control provides a good performance in trajectory tracking of some nonlinear systems. Whereas, a notorious characteristic of variable structure control approach is the discontinuity around the switching hyperplane, that means some of the state variables are vibrant. One of the methods to tackle the problem is to utilize a feedforward compensator to cancel unpredictable influence of system uncertainties. A self-organizing fuzzy cerebellar model articulation controller (FCMAC) is proposed as a compensator for this purpose. In the approach, a neural network is designed to alleviate the chattering behavior, which is always observed in the sliding mode controller. Since the original CMAC is based on the supervised learning and not suitable for on-line control, a reinforcement learning algorithm which is similar to the adaptive heuristic critic (AHC) proposed by Barto et al. is applied to supporting this requirement [4]. The synaptic weights of the FCMAC are updated by a heuristic reinforcement signal which represents the temporal difference error between two successive predictions of event failure. At the standpoint of the AHC, the FCMAC is employed as a state variable quantizer (decoder). The AHC predicts the degree of the event failure and resolves the compensating control signal while the quantized state variables is fed in.

This article is organized as follows: the previous works and proposed control schemes are introduced in Section II and III, respectively. The simulation results are illustrated in Section IV. The feasibility and performance of the proposed controller is discussed in Section V. Finally, a conclusion is drawn in Section VI.

II. HYBRID NEURO-FUZZY CONTROL SYSTEM

The proposed hybrid control system consists of a primary fuzzy controller incorporating with sliding mode control technique, called the fuzzy sliding mode controller (FSMC) [7], [11], [21], [23], [30], [31], and an FCMAC compensator. The FCMAC with self-organizing capability is designed for alleviating the chattering effect resulting from the discontinuity of the sliding mode control law. The architecture of the proposed hybrid neuro-fuzzy system is shown in Fig. 1.

A. Fuzzy Sliding Mode Control

In the following we describe the $n$-dimensional nonlinear SISO system

$$y^{(n)} = f(y, y_t, \ldots, y^{(n-1)}) + bu, \quad b > 0 \quad (1)$$

Fig. 1. The architecture of a hybrid control system.
where \( f(\cdot) \) is an unknown continuous function; \( b \) is an unknown positive constant; \( u_S \in R \) is the system input and \( y \in R \) is the system output. Then the error signal vector can be expressed as
\[
e_{i} = r_{i-1}^{(i-1)} - y_{i-1}^{(i-1)}, \quad i = 1, 2, \ldots, n - 1
\] (2)
where \( r_{i-1}^{(i-1)} \) is the \((i-1)\)th reference input. The dynamic equations (1) and (2) can be transformed into the following form:
\[
\begin{align*}
\dot{e}_i &= e_{i+1}, \\
\dot{e}_{n} &= r_{n}^{(n)} - f - bu_S,
\end{align*}
\] (3)
The variable structure control system is generally characterized by a control law that is discontinuous on a predefined switching hyperplane to obtain a desired response. This is accomplished by using high-speed switching relay-type control, which forces the trajectories of the system onto the switching hyperplane. When the state variables is maintained on the plane, the control is usually referred as in the sliding mode control.

The switching hyperplane of the FSMC is defined by
\[ S = \mathbb{C}^T e \] (4)
where \( \mathbb{C} = [c_1, c_2, \ldots, c_{n-1}]^T \); \( c_i > 0 \) is a constant parameter and \( i = 1, 2, \ldots, n - 1 \).

The equivalent control law to maintain the state trajectory on the switching hyperplane (i.e., \( S = 0 \)) is determined by [21]
\[
u_{eq} = u_S|_{S=0} = b^{-1}(-f + r_{n}^{(n)} + \mathbb{C}^T \cdot e)
\] (5)
where \( \mathbb{C}' = [c_1, c_2, \ldots, c_{n-1}]^T \). During the sliding mode, some excellent features are theoretically achieved, such as robustness, disturbance rejection, low overshoot, and high accuracy [8], [10], [27], [28]. The output of the proposed FSMC consists of a fuzzy control signal \( u_f \) obtained from the fuzzy logic controller and a hitting control signal \( u_h \). In other words, the combined control signal from the FSMC applied to the plant can be expressed as follows:
\[ u_S = u_f + u_h \] (6)
The signal \( u_h \) should drive the state variables to hit the switching hyperplane no matter what kind of uncertainties the system may have. The hitting control signal can be derived from the following equations. Let \( V_S \) be a Lyapunov-like function for \( S \)
\[ V_S = \frac{1}{2} S^2 \] (7)
then the derivative of is given by
\[ \dot{V}_S = SS \] (8)
The condition for the sliding mode occurring on the switching hyperplane is [13]
\[ V_S = SS < 0 \] (9)

If \( \mathcal{F} \) is the upper bound of \( f(\cdot) \) and \( b \) is the lower bound of \( b \), i.e., \( |f(\cdot)| \leq \mathcal{F} \), and \( 0 < b \leq b \) then
\[ \dot{V}_S = Sb(u_{eq} - u_f - u_h) \leq |Sb|(|u_{eq}| + |u_f|) - Sbu_h \] (10)
To guarantee the stability of FSMC, (10) must less or equal to zero. So the hitting control signal \( u_h \) can be selected as
\[ u_h = \text{sign}(S)[b^{-1}(\mathcal{F} + |r_{n}^{(n)}| + |\mathbb{C}^T \cdot e| + |u_f|]
\] (11)
The study is concentrated on the method of dividing the control signal \( u_S \) into two parts, \( u_f \) and \( u_h \) instead of the switching hyperplane design. In addition to deriving the control signal to control the plant, the proposed system also applies a compensator to reducing the chattering and system uncertainty for the FSMC. The structure of the compensator is developed based on the concept of the CMAC and AHC, which are briefly illustrated in the following sections.

B. Self-Organizing Cerebellar Model Articulation Controller

The architecture of the cerebellar model articulation controller (CMAC) consists of two processing stages [1], [2], [9], [15], [22], [26]. Firstly a nonlinear transformation maps the input state variable \( x \in R^m \) into a higher-dimensional vector \( v \in \{0,1\}^m \) where \( n \) and \( m \) is the dimension of \( x \) and \( v \), respectively. The vector \( v \) is a sparse vector in which at most \( C \) of its components are excited (\( C \) is called a generalization parameter which is the ratio of generalization width to quantization width). The corresponding cells are nonzero if the state variable falls in the excited region as shown in Fig. 2. The input state variables which are close to each other in receptive fields produce similar output vectors. It resembles the capacity of a biological organism to generalize learning experience from one to another. It is referred as local generalization.

The mapping is realized by feeding the binary output on the sensor layer into the logical AND units. Each AND unit receives an input from each receptive field and it is sparsely interconnected to the next logic OR operation unit. Fig. 3 depicts the schematic diagram of two-dimensional (2-D) CMAC operations with \( C = 3 \). After the mapping operation, each subset is relative to the others along hyperdiagonal in the input hyperspace. Besides, each element falling within the subsets on the hyperspace has a value of one. The nonlinear
mapping for a 2-D CMAC can be represented as an \( m_1 \times m_2 \) matrix \( g(x) \). The result of the mapping is drawn in Fig. 4.

The output \( y \in \mathbb{R}^d \) of a CMAC is generated by the second stage through the following equation:

\[
y = \sum_{j=1}^{m_2} \left( \sum_{i=1}^{m_1} g_{ij}(x) \cdot W_{ij} \right)
\]  

(12)

where \( g_{ij}(x) \) is the element of \( g(x) \); \( W_{ij} \) is the element of the \( m_1 \times m_2 \) real-valued weight matrix \( W \). The CMAC weights, in general, are initially all zeros and the memory elements have a null response to any input until it has been trained. During the weights are being trained, the desired output \( y_d \) is compared with the CMAC’s output \( y \) for each input. The values of the selected weights are adjusted to reduce the errors between the desired and actual ones.

The weight-update equation of the CMAC is given by

\[
W_{ij}(t + 1) = W_{ij}(t) + \rho \left( \frac{y_d - y}{N} \right)
\]  

(13)

where \( \rho, 0 < \rho < 1 \), is the learning rate; \( N \) is number of weights which contribute to the output \( y \).

Since the delta rule of (13) needs the desired output provided, the supervised learning is not suitable for the CMAC applied to the on-line control. Therefore a self-organizing CMAC which implements a reinforcement learning algorithm called the adaptive heuristic critic (AHC) is developed to tackle this problem [3], [4].

The AHC algorithm is an alternative form of self-organizing learning algorithm. It follows the BOXES scheme in quantizing state space and constructs a neural network controller to balance a pole mounted on a cart, which can move along one dimension as shown in Fig. 5. The state variable of the cart-pole system is coded into a sparse vector with 162 components and each box is represented by a state vector. A constant force with opposite direction, \( \pm 10Nt \), is applied as a control signal, and updated every 0.02 s. Four feedback state variables, \( \theta, \dot{\theta}, h, \dot{h} \), represent the pole angle with respect to the vertical axis, angular velocity, position and velocity of the cart, respectively. Failure occurs when the pole falls over \( 12^\circ \) or the cart moves beyond \( \pm 2.4m \) from the origin.

The state variables of the cart-pole system are sampled and quantized into an \( n \)-component binary vector \( \mathbf{X} = (x_1(t), x_2(t), \ldots, x_n(t)) \). The components of vector \( \mathbf{X} \) are all zeros except for one in the position corresponding to the state of the system at that instant. The reinforcement control function can be divided into two tasks: the action function and the evaluation function. The action function and the evaluation function are implemented by an associative search element.
(ASE) and an adaptive critic element (ACE), respectively. At each synapse of the ASE exists two trace signals: the long-term trace $W_i(t)$ that determines control actions and the short-term trace $\hat{e}_i(t)$ that is required to update the long-term trace. The update rules of ASE are expressed as follows:

$$
\hat{e}_i(t+1) = \delta \hat{e}_i(t) + (1 - \delta)Y(t)x_i(t)
$$

$$
w_i(t+1) = w_i(t) + \alpha \hat{r}_i(t)\hat{e}_i(t)
$$

(14) (15)

where $\delta$ is the ASE trace decay rate; $\alpha$ is the positive learning rate; $\hat{r}_i(t)$ is the internal reinforcement signal; $\hat{e}_i(t)$ is the eligibility function at time $t$ of input pathway $i$ which indicates how much credit an earlier state has for the result of current system status. The element’s output $Y(t)$ is determined from the input quantized vector as follows:

$$
\text{net} = \sum_{i=1}^{n} w_i(t)x_i(t)
$$

$$
Y(t) = \text{sign}(\text{net} + \text{noise}(t))
$$

$$
= \begin{cases} 
1, & \text{if } \text{net} + \text{noise}(t) \geq 0 \\
-1, & \text{if } \text{net} + \text{noise}(t) < 0.
\end{cases}
$$

(16) (17)

On the other hand, the ACE provides an evaluation of problem states and functions as a prediction of failure. A single action during balancing attempt can be rewarded or punished by comparing the evaluations of the state preceding the action with the state following the action. The capability allows the learning to occur on the trial and not only upon failure. The following equations illustrate the operation of the ACE:

$$
\tau_i(t+1) = \lambda \tau_i(t) + (1 - \lambda)\hat{e}_i(t)
$$

$$
v_i(t+1) = v_i(t) + \beta \hat{r}_i(t)\tau_i(t)
$$

(18) (19)

where $\lambda$ is the ACE trace decay rate; $\beta$ is the positive learning rate; $\tau_i(t)$ indicates the effect that $x_i(t)$ has on the current results. The output of the evaluation unit is the inner product of the input quantized vector and the unit’s weight vector

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

(20)

where $n$ is number of excited states. Unlike the single-step prediction or the supervised learning method, which assigns credit to the difference between the predicted and actual output, the temporal difference (TD) methods assign credit according to the difference between temporally successive predictions [25]. The goal of the critic is to predict the following quantity:

$$
p(t) = \sum_{k=0}^{\infty} \gamma^k r(t+k+1)
$$

(21)

where $r(t+k+1)$ is the external reinforcement signal at time $t+k+1$ and $0 \leq \gamma < 1$, is a discount factor which determines the level of influence of future reinforcements on the selection of the current action; $p(t)$ is the prediction of infinite discounted cumulative outcomes. If the reinforcement predictions in (21) are assumed to be correct, $p(t)$ can be rewritten as

$$
p(t) = r(t+1) + \gamma \sum_{k=0}^{\infty} \gamma^k r(t+k+2)
$$

$$
= r(t+1) + \gamma p(t+1).
$$

(22)

Hence, the internal reinforcement signal is an appropriation to the temporal difference error

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

(23)

This function is learned via the TD method, that adjusts the weights of the AHC network in proportion to the difference between reinforcement predictions on the consecutive steps.

From the viewpoint of the AHC, the vague credit-assignment problem usually exists in the reinforcement learning. The problem can be hence relaxed to some extent by applying the CMAC as the state variable quantizer (decoder) in the AHC, because the CMAC has the characteristic of sparse quantization.

### III. System Implementation

A. Scheme of Fuzzy Logic Control

The hybrid control system is developed to control a three-link robot. The controller consists primarily of a FSMC, which incorporates the fuzzy logic control with the sliding mode control technique, and a FCMAC with self-organizing capability. The fuzzy control strategy takes the form of fuzzy inference statements [14], [18], [19], [32]. For example, if the error of the link is “positive big” and the derivative of the error of the link is “positive big,” then the torque strategy deriving is “positive big.” The fuzzy inference statements are also referred as the fuzzy control rules. For simplicity, the fuzzy term set of input variables in this work has the cardinality five, given by $F = \{NB, NS, ZO, PS, PB\}$, where NB, NS, ZO, PS and PB linguistically denotes Negative Big, Negative Small, Zero, Positive Small and Positive Big, respectively. For instance, the fuzzy rule table of the robotic trajectory control can be designed as Table I.

<table>
<thead>
<tr>
<th>Rule Table of Robotic Trajectory Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_i$</td>
</tr>
<tr>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
</tr>
<tr>
<td>ZO</td>
</tr>
<tr>
<td>PS</td>
</tr>
<tr>
<td>PB</td>
</tr>
</tbody>
</table>

The membership functions with a nonisosceles triangle shape in Fig. 7 are used as the fuzzification function for the purpose of fine-tune control for small errors and coarse control for large errors, respectively. In the approximate reasoning and defuzzification, the max-min compositional operators and the center of gravity method (COG) are adopted, respectively.
B. Fuzzy Cerebellar Model Articulation Controller

Unlike the conventional fuzzy rule base controller, a neuro-fuzzy system does not need extract control rules from the experts. The learning is based on observations of the input/output relationship of the system. By carefully inspecting the operation process of the CMAC and fuzzy logic algorithm, there exists some striking similarities between both systems. For example, they perform function approximation in an interpolation look-up table manner with the principle of dichotomy and generalization. Moreover, the nonlinear mapping of the CMAC can be regarded as the subset of the aggregation operation on fuzzy sets.

The standard univariate basis function of the CMAC is binary so that the network modeling capability is only piecewise constant. Whereas the univariate basis functions with higher-order piecewise polynomial, which can generate a smoother output, have recently been investigated [6], [12], [16]. A crisp set can be considered as a special form of fuzzy set, to which an instant can to some degree either belong or not belong. The property is similar to the problem whether the state variable excites a specific region in the sensor layer of the CMAC or not. The membership function, \( \mu_k(x) \rightarrow [0, 1] \), associates each state variable \( x \) with a number. It represents the grade of membership of \( x \) in \( \mu_k(x) \). Three different shape triangles are adopted for the generalization parameter \( C = 3 \), so that only one maximal output exists in the hyperspace after aggregation operation. These membership grades have a peak value at the center of the excited region, and decrease as the input moves toward the edge of the excited region. Fig. 8 depicts the organization of the overlapped receptive fields in the input space. Different grades of membership are assigned to the corresponding cell in \( \mu_k \) if one of the quantization region is excited by a given state variable.

The sequence of \( \mu_{13} > \mu_{12} > \mu_{11} ; \mu_{22} > \mu_{21} > \mu_{23} ; \mu_{31} > \mu_{32} > \mu_{33} \) is notable for a state variable \( x \) which is near the left of the excited region in Fig. 8. In the extreme case, when every state variable \( x \) falls on the center of its corresponding excited region, a fuzzy singleton \( \mu = 1 \) is generated at the intersection of every subset in the hyperspace. The 2-D FCMAC operations, where each subset is offset relative to the others along hyperdiagonal in the input hyperspace, are illustrated in the following schematic diagram.

The nonlinear mapping of the FCMAC is implemented by replacing the logic AND and OR operation in the CMAC with the commonly used T-norm and T-conorm operations, respectively. The comparison of CMAC and FCMAC with some T-norm and T-conorm dual operators is listed as Table II [29]. The nonlinear mapping result of the FCMAC is shown in Fig. 10. The algebraic product of T-norm and algebraic sum of T-conorm are adopted in this work since they produce smoother output surfaces and can make the system analysis available as well.

C. Self-Organizing FCMAC Controller

One of the objectives of this work is to incorporate the FCMAC with the AHC algorithm to reduce the effect of the system uncertainty [20]. A major characteristic of the proposed system is the capability of using the experience with incomplete knowledge about the plant to predict its behavior. However, some control schemes should be modified if they can be implemented successfully.

1) In most of the studies on the TD procedures, the AHC’s weights are updated after either the presentation of
TABLE II
THE COMPARISON OF CMAC AND FCMAC

<table>
<thead>
<tr>
<th>CMAC</th>
<th>FCMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>characteristic function</strong></td>
<td><strong>membership function</strong></td>
</tr>
<tr>
<td>AND operation</td>
<td>T-norm</td>
</tr>
<tr>
<td>$a \land b$</td>
<td>Min(a,b) Fuzzy Intersection</td>
</tr>
<tr>
<td></td>
<td>Max(0,a+b-1) Bounded Product</td>
</tr>
<tr>
<td></td>
<td>ab Algebraic Product</td>
</tr>
<tr>
<td>OR operation</td>
<td>T-conorm</td>
</tr>
<tr>
<td>$a \lor b$</td>
<td>Max(a,b) Fuzzy Union</td>
</tr>
<tr>
<td></td>
<td>Min(1,a+b) Bounded Sum</td>
</tr>
<tr>
<td></td>
<td>a+b-ab Algebraic Sum</td>
</tr>
</tbody>
</table>

![Fig. 9. The nonlinear mapping of 2-D FCMAC.](image)

new state variables or a complete training sequence. However, if the changes of the state variables and the weights of the ACE induce an alteration to the internal reinforcement signal within a sequence, it probably leads to unstable in the training process. The refinement process is taken as follows [24], [25]:

$$\text{(24)}$$

It ensures that the changes in prediction owing to $x(t)$ and $x(t+1)$ are effective in causing weights alteration.

2) The prediction can be transformed into

$$p(t) = H \left( \eta \sum_{i=1}^{n} u_i(t)x_i(t) \right)$$  \hspace{1cm} (25)

where $H(\cdot)$ could be a sigmoid-shaped function [17] and $\eta \in (0,1]$ is an user-specified parameter to prevent from saturation, if there are too many training cycles. The purpose of adopting the sigmoid function is to prevent the $p(t+1)$ and $p(t)$ from domiming the terms in the right side of (23). A hyperbolic tangent function $\tanh(\cdot)$ is chosen in this work since it is signed and null at the initial stage of each training procedure.

![Fig. 10. The nonlinear mapping result of 2-D FCMAC.](image)

3) Equation (16) is modified by the hyperbolic tangent function $\tanh(\cdot)$, so it preserves the learning direction $Y(t)$ in updating the eligibility function $E_i(t)$. Equations (16) and (17) can be rewritten as

$$\text{net} = \tanh \left( \sigma \sum_{i=1}^{n} w_i(t)x_i(t) \right)$$  \hspace{1cm} (26)

$$Y(t) = \text{sign}(\text{net})$$  \hspace{1cm} (27)

where $\sigma$ is used to adjust the shape of the hyperbolic tangent function in case of

$$\sum_{i=1}^{n} w_i(t)x_i(t) \ll 0.01.$$  \hspace{1cm} (28)

The control method then changes into a continuous-type control rather than the bang-bang control.

4) The compensator’s output $u_C$ is simply assigned as the multiplication of net of (26) and a gain $k$. The applied control signal is formalized as follows:

$$u_T = u_S + u_C = u_h + u_f + k \cdot \text{net}$$  \hspace{1cm} (28)

where $u_S$ is the output of the FSMC.

The modified learning algorithm can learn to balance a pole within 15 trials by the proposed control approach. That shows a similar performance described in [17]. The procedures of the control process are illustrated in Fig. 11.

IV. SIMULATION RESULTS

In the paper, a three-link robot shown in Fig. 12 is used for simulation. As for the training pattern generation for the robot, the Van der Pol Equation is used as the trajectory planner for the robot motion [5], [13]:

$$\dot{z}_1 = z_2$$  \hspace{1cm} (29)

$$\dot{z}_2 = -z_1 + \varepsilon (1 - z_2^2)z_2.$$  \hspace{1cm} (29)

Since the human’s motion exhibits a stable limit cycle, The training patterns $\phi_H$ and $\phi_iH$ are generated by adopting $\varepsilon = 1$ and multiplying $z_{1*}, z_{2*}$ by the factor 5, 0.1, respectively.
The control objective is to drive the plant state vector $\phi_k$ and $\dot{\phi_k}$ to track a specified desired state vector $\phi_{k_d}$ and $\dot{\phi_{k_d}}$, respectively. For the $i$th link, the error signal vector can be represented as $\epsilon_i = \phi_{ki} - \phi_{k_d}$ and $\dot{\epsilon}_i = \dot{\phi}_{ki} - \dot{\phi_{k_d}}$, which is feedback to the FSMC through the fuzzy rule table in Table I.

The fuzzy rule table is notably the same as the one that has been used in the cart-pole system controller, but the output of each FSMC in the link is changed into a cumulative control signal of the original output $\tau_i(t)$. In other words, the torque of the $i$th link can be represented as

$$
\tau_{Si}(t + 1) = \tau_{Si}(t) + \tau_{Si}(t).
$$

(30)

Fig. 11. The flowchart of self-organizing FCMAC.

Through such modification the FSMC is able to force each link to produce the desired output $\phi_{k_d}$, $\dot{\phi_{k_d}}$. However, the angular velocity $\dot{\phi_k}$ vibrates between the both sides of the desired trajectory $\dot{\phi_{k_d}}$. It is owing to the discontinuous control signal of the FSMC and the coupling interaction between the links. To overcome the coupling effect, a learning algorithm is adopted to compensate the influence from the system uncertainties. In control systems, it is difficult to determine a priori what the best input/output mapping is for a specific task. So it would be more appropriate to employ reinforcement learning mechanism, which has the ability to find solutions for problems by receiving a suitable evaluation of its performance. The criteria in the implementation depend on whether the state variables exceed a specific boundary or not. The angular velocity $\dot{\phi_k}$ is detected for critic and pre-processed with a moving average signal procession defined by

$$
\dot{\phi_{c}}(t) = \sum_{j=0}^{m} k_j \cdot \dot{\phi}(t-j)
$$

(31)

where $\dot{\phi_{c}}(t)$ is the angular velocity to be criticized. By using the signal processor a spike of angular velocity can be distinguished from a ripple one so that the punishment can be given more correctly.

The architecture of the proposed three-link robot controller is illustrated in Fig. 13. Since the dynamics of the link 1 can significantly affect the behavior of the links 2 and 3, the torque $\tau_{T1}$ is used as an input to the compensators 1 and 2. This allows the controller of links 2 and 3 to compensate for the coupling effects form the link 1 and hence reduce the angular velocity vibration. The torque $\tau_{T2}$ is fed to the compensator 2 for the same reason. If the memory requirement of the FCMAC is taken into consideration, the other signals fed into the compensator should be only the state variables and the applied control signal of each controller.

Each link is equipped with a FSMC controller individually, but the compensator is only applied to the links 2 and 3. The coefficients of the switching hyperplane are given as $C_2 = [10.1]^T$, $C_3 = [11.1]^T$, and $C_3 = [0.11.1]^T$ for the FSMC’s on the links 1, 2, and 3, respectively. The Van der Pol equation parameter $\varepsilon = 1$ is utilized as the trajectory planner, which generates the training patterns $\phi_{Ah}$ and $\phi_{Ad}$ for the robot motion. The angle $\phi_{j2}$ is in phase with angle $\phi_{k2}$, whereas the angle $\phi_{j1}$ is out of phase with the other angles. The proposed hybrid controller is adopted to track the desired trajectory $\phi_{k1}$ and $\phi_{k2}$. In Fig. 14–19, two configurations of the
hybrid control are compared. Part (b) is the result of the self-organizing compensator equipped with the fuzzy quantization as introduced in Section III-B, but Part (a) is the one without fuzzy quantization.

From Figs. 14–16, the proposed controller provides a good performance on position trajectory control except that the tracking error still exists. Although only the angular velocity $\dot{\phi}_q$ is feedback into the critic, the position tracking accuracy is not affected by the compensator significantly. Unfortunately, the compensator is not capable of handling the vibration existing in the angular velocity as observed from Figs. 17–19. Figs. 20–22 show the magnified portion of Fig. 19 at the different intervals. The compensator with learning mechanism can not discriminate a spike of angular velocity from a ripple one, and give an appropriate control action to inhibit the vibration (see Fig. 20). It is due to the learning interference, which is discussed in the following section, and the sparse partition of the state space. On the average, the learning performance of the FCMAC-based controller is better than the CMAC-based one (see Fig. 21). The better outcome of

---

Fig. 13. The architecture of three-link robot control using FSMC and self-organizing FCMAC controller.

Fig. 14. Angle $\phi_1$ of three-link robot.
the proposed control scheme is illustrated in Fig. 22, since the control action is not punished by the AHC any more.

V. DISCUSSION

A. Optimization of FSMC

The simulation results show that the realization of the adaptive hybrid control system is possible. However, too many parameters in the controller need to be determined, so it causes difficulty to achieve an optimal design. In the FSMC, the performance depends on selection of control strategies and how to construct the switching hyperplane. The dynamic behavior of the control system can be governed by adjusting the parameters of a switching hyperplane \( C = [c_1, c_2, \ldots, c_{n-1}]^T \). In deciding how to set up the parameters of switching hyperplane, one must strike a balance between robustness and convergent speed. If a larger parameter \( c_i \) is used, the state variables will converge fast, but probably lead to instability. Moreover, when many sets of the FSMC’s are employed to control a physical system simultaneously, it is too complex and difficult to tune the parameters of each FSMC. Therefore it is suitable to take a time-varying switching hyperplane in different configurations. For instance, a larger \( c_i \) can be used for the FSMC on the link 1 to drive actively the state variables to the specific switching hyperplane, whereas the other two links move at low speed by using small \( c_i \). The parameter \( c_i \) in the link 1’s FSMC is reduced when the set-point of the link 1 is achieved. Then the parameter \( c_i \) in the other two links’ FSMC’s are increased to accelerate the links to the desired trajectory. In this way, the external disturbance or the nonlinear coupling effect between the links can also be rejected efficiently. Besides, the effect of gravity can be ignored in the FSMC since the desired outputs varies slowly from the current point to the next one, and the FSMC can handle this problem successfully.
B. Performance of FCMAC

The resolution of quantization is sparse with respect to the generalization parameter $C = 3$ in the simulation. If the subsequent training input is chosen in the same neighborhood as the previous one, some memory elements that were adjusted in the previous training sessions might be improperly altered. The property is called learning interference. That is similar to retroactive inhibition experience by biological organisms when presented with highly similar stimuli for which different responses are required. It is better to construct a smaller-dimensional subnetworks so that new input variables can be innovated without significantly affecting the information already stored there. Besides, one must face the trade-off problem between generality and memory capacity in deciding how to partition the state space. A fine quantization with many tiny regions promises more accurate approximation of nonlinear functions, but is time-consuming in learning the correct output for each region. Learning ability can be improved with a coarse quantization if the input seldom enters the region. On the other hand, more memory should be allocated to the region if input concentrates in the region. Obviously, an adaptive learning algorithm based on experience is still needed for the quantization of state space.

The modeling capability of the FCMAC can only be achieved over a bounded input space because the number of basis function is generally finite. The graded neighborhood functions of the FCMAC overcome the problem of discontinuous response over neighborhood boundaries due to the crisp neighborhood functions of the CMAC. The region on which generalization is effective tends to remain constant over the whole input space even for different inputs. This is undesirable since, for many problems, training patterns will be tightly clustered together in some regions, and widely spaced in others. Because of the intrinsic local generalization property of the FCMAC, it can approximate the functions successfully that are slowly varying. In other words, the FCMAC can
approximate the function which is monotonic between the partite cells. However, it always fails to approximate functions that oscillate rapidly or are highly nonlinear. The decision on the way of partitioning the control problem for a FCMAC depends on what type and how many of feedback variables are available. As well, which aggregation operation is employed, and which updating formula is used in training. By using the FCMAC, more information can be extracted from a specific memory than the CMAC if the learning interference does not occur. Whereas, the FCMAC may be unstable in some learning rates and generalization parameters as well as the CMAC. There is, unfortunately, no general principles about how to select the parameters of the FCMAC proposed in any related research literature.

Because of the property of local generalization, the FCMAC is more robust than the conventional controller since it counts a region rather than just a point that the state variable has fallen on. In other words, a neuro-fuzzy controller can realize not only the standard linear controllers, but also their robust control behaviors. In summary, the partitioning strategies of input space, the type of the membership function, and the aggregation operations must be taken into consideration if one would apply this technique in practice.

C. Stability of AHC

Reinforcement learning does not require a priori dynamic model, but learns on the basis of the experience obtained directly from the environment. If the future reinforcement predictions \( p(t+1) \) is right, the weight of ASE and ACE would be adjusted in the correct direction. The primary disadvantage of the reinforcement learning introduced is that many repeated experiences, in general, are required to learn an optimal control strategy, especially if the system starts at a poor initial condition. In addition, unlike the supervised learning, the optimization problem is much harder because the only information about the desired trajectory is the plant’s objective...
states. Since the eligibility function is used to estimate how much credit an earlier state should have, there is no other useful action evaluation feedback to the controller with regard to its transient performance. It tends to ignore a destructive control action or state variable which happens seldom, so that the prediction becomes invalid. In this case, it is better to adopt the FCMAC (or CMAC) as the quantizer, since the neighboring cells of the FCMAC can share the same memory if they have similar input. This sharing capability provides generalization, so the information can be interpolated even for the cells that have not yet been entered. The performance of the AHC also depends on the discount factor $\gamma$ as well as $\rho(t+1)$. Note that the closer $\gamma$ is to 1 the longer the term of future external reinforcement signal is accounted for in the critic prediction. However, as $\gamma$ approaches 1 the learning generally tends to be more unstable. In fact, hitherto no TD method has ever been proved stable or convergent to the correct predictions except the extreme case for linear TD(0) which has been verified in [25]. It causes to the specific boundary (\$X$ and $\phi$) is fixed in our application. If the stricter criterion is used, more control action will be punished. It probably leads to learning interference. It is appropriate to take a varying criterion if the stability and convergence of the AHC should be proved.

VI. CONCLUSION

The proposed hybrid control scheme has a learning mechanism realized by a self-organizing FCMAC. The learning mechanism plays a role on the compensator of the main control module, the FSMC. This FCMAC-based neural network has the capability of approximating complex or nonlinear functions with multidimensional inputs. In the article, the AHC algorithm is treated as the weight-update rule of the FCMAC. This control scheme can be implemented on any nonlinear system by giving appropriate criteria as the performance measurement. The simulation shows that a control task can be learned even with very little a priori knowledge.
To design a biped robot, one may not be able to specify a desired trajectory of the limbs, but the objectives of the robot motion control, such as moving forward, maintaining equilibrium and safety, and so on, can be imposed on the system. In such cases, the reinforcement learning-based controller has better performance than the conventional ones. The proposed control scheme can be extended to a hierarchical structure of the FCMAC-based controller which is able to assign the multiple objectives into each modular. Although the uncertainty and interaction between the subsystems can be handled in the learning process of the neuro-fuzzy control system, the application is, unfortunately, still limited due to the problems of the memory capacity, stability of neural network, decision of numerous parameters, and so forth.

ACKNOWLEDGMENT

The authors would like to thank the reviewers and C.-S. Sheng, Yuan-Ze Institute of Technology, Taoyuan, Taiwan, R.O.C., for their helpful extensive comments, which have significantly improved the presentation of this research.

REFERENCES


Kao-Shing Hwang received the B.S. degree in industrial design from National Cheng Kung University, Taiwan, R.O.C., in 1981, and the M.M.E. and Ph.D. degrees in from Northwestern University, Evanston, IL, in 1989 and 1993, respectively.

Since August 1993, he has been an Associate Professor of Electrical Engineering at National Chung Cheng University, Chia-Yi, Taiwan, R.O.C. He was a Design Engineer at the Sanyo Electric Co., Taipei, Taiwan from 1983 to 1985. From 1987 to 1988, he was with the C&D Microsystem Co., Plastow, NH, as a System Programmer, designing PC drivers and graphic animation. He is also the Director of the Information Management Division at the University computer center. His areas of interest are neural networks and learning control, robotic compliance, and collision avoidance.

Ching-Shun Lin was born in Taiwan in 1971. He received the B.S. degree in 1994 from the National Taiwan Ocean University, Keelung, Taiwan, R.O.C., and the M.S. degree in 1996 from the National Chung-Cheng University, Chia-Yi, Taiwan, both in electrical engineering.

In addition to the modeling and simulation of artificial neural networks, his research interests include control theories, fuzzy systems, genetic algorithms and machine learning.