Proceedings of the
American Control Conference
Seattle, Washington • June 1995

A New Design of Adaptive Robust Fuzzy Controller for Nonlinear Systems

Feng-Yih Hsu[1] and Li-Chen Fu[1,2]

Dept. of Electrical Engineering[1]
Dept. of Computer Science & Information Engineering[2]
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

This paper presents an adaptive robust fuzzy control architecture for a class of nonlinear dynamic systems which are either ill-defined or rather complex. The control objective is to adaptively compensate for the unknown plant nonlinearity, which is represented as a fuzzy rule-base consisting of a collection of *if-then* rules. The algorithm embedded in the proposed architecture can automatically update fuzzy rules and, consequently, is guaranteed to be globally stable and to drive the tracking errors to a neighborhood of zero. Focused on realization, hardware limitations such as traditional long computation time and excessive memory-space usage are also relaxed by incorporating heuristic concepts, which reveals the flexible feature of this architecture. Simulations are run for the control of a simplified circuit system, and results show that the proposed control architecture is featured in fast convergence.

1 Introduction

During the past decade, intelligent control methodologies have gradually been recommended to solve a number of complicated problems that, especially, conventional control methodologies are hard to handle. Those methodologies often use biologically motivated techniques and processes, and are referred to as neural networks, fuzzy-rule base, knowledge-base, or other learning schemes. Unlike general conventional schemes based on a complete theory and algorithmic structure, they are in general hardly evaluated. Therefore, it is imperative to make efforts on bridging the gap between the conventional control scheme and the intelligent ones [1]. Recently, analysis on intelligent control has attracted enormous research interests [2]-[8]. Ideas behind those schemes are to strengthen their theoretic basis but at the price of expensive implementation by using massive networks and extensive rule-tables or complex functions, which generally lead to difficulties in real implementation due to hardware limitations such as long computation time and excessive memory-space usage. On the other hand, the heuristic nature of intelligent control often has substantial benefit for a controlled system. Hence, an integrated consideration is suggested in this paper.

In this paper, we present a new fuzzy control methodology which can be applied to the control of a class of nonlinear systems. It is known that fuzzy logic controllers (FLC) have been widely applied in industry. An important advantages of using FLC is that fuzzy theories can capture the approximate, qualitative aspects of human knowledge and reasoning. Apparently, such control provides a rather feasible alternative for a plant that is complex or ill-defined [10]. To extend the range of application of FLC, the current research efforts have been made as how to automatically generate rules or how to adaptively update the existing rules. There is a common feature in those proposed schemes, namely, the fuzzy theory is always combined with some other control architectures such as neural networks [10], [11], adaptive control [8], learning control [9], etc. Particularly, adaptive control concept is often incorporated into some other intelligent schemes. For examples, Messner et al. in [7] proposed a new adaptive learning scheme which uses some kernel function of time as a general regressor. Contrarily, in [2], a spatial Fourier transform is adopted to furnish a more general kernel function of state which is then realized by a Gaussian radial network. But unlike the two, the fuzzy basis function is assumed in [8] with a viewpoint that a heuristic algorithm used for control tends to be more flexible in satisfying the approximation need. Apart from the above adaptive concept, another robust control concept also has to be adopted to ensure that the overall controlled system is stable. Thus, a control synthesis is proposed here, which combines the adaptive fuzzy control approach and a robust control approach to generate the final adaptive robust fuzzy control (ARFC) scheme.

2 Problem Formulation

Consider a class of nonlinear systems described by the following dynamic equations expressed in canonical form:

$$x^{(n)}(t) = f(x(t), \dot{x}(t), x^{(2)}(t), \cdots, x^{(n-1)}(t)) + b(x(t), \dot{x}(t), x^{(2)}(t), \cdots, x^{(n-1)}(t))u(t), \qquad (1)$$

where $u(t)$ is the scalar control input, f is an unknown function, and b is an unknown positive control gain such that $b \geq \delta$ with $\delta > 0$. Our control objective is to track a desired trajectory, i.e., to force the plant state (vector) $\mathbf{x} = [x, \dot{x}, x^{(2)}, ..., x^{(n-1)}]^T$ to follow a specified desired trajectory $\mathbf{x}_d = [x_d, \dot{x}_d, x_d^{(2)}, ..., x_d^{(n-1)}]^T$. After defining the tracking error (vector) $\mathbf{e} = \mathbf{x}_d(t) - \mathbf{x}(t) = [e, \dot{e}, e^{(2)}, ..., e^{(n-1)}]^T$, our goal is thus to design a control $u(t)$ which ensures that $\mathbf{e}(t) \to 0$ as $t \to \infty$.

Generally, the controller design will require the knowledge of f and b to some degree of accuracy. But for more complex or ill-defined systems, f and b can not be easily estimated well, which will then result in uncertainties in plant modeling. Commonly speaking, the uncertainties are often denoted as a deviation of the actual plant from the nominal plant. Let $F = -b^{-1}f$ and $B = b^{-1}$, then the uncertainties are denoted by $\Delta F = -b^{-1}f - \widehat{F}$ and $\Delta B = b^{-1} - \widehat{B}$, where $\widehat{F}$ and $\widehat{B}$ are nominal version of $-b^{-1}f$ and b^{-1}, respectively. Given this nominal plant subject to the foregoing control objective, a nominal controller $\widehat{u}$ is designed as follows:

$$\widehat{u} = \widehat{F} + \widehat{B}x_r + k_s s, \tag{2}$$

where s is a sliding mode variable $s = \lambda_0 e + \lambda_1 \dot{e} + ... + e^{(n-1)} = [\lambda_0, \lambda_1, ..., 1]^T \mathbf{e} = \lambda^T \mathbf{e}$ with the transfer function $\Delta(p) = p^{n-1} + \lambda_{n-2}p^{n-2} + \cdots + \lambda_1 p + \lambda_0$ being a Hurwitz polynomial, and $k_s > 0$, $x_r = x_d^{(n)} + \lambda_0 \dot{e} + \cdots + \lambda_{n-2}e^{(n-1)}$. If we let the real control be $u = \widehat{u} + \Delta u$, where Δu is to be specified later, then the dynamic equation of the sliding mode variable s is given as:

$$b^{-1}\dot{s} = -k_s s + \Delta F + \Delta B x_r - \Delta u. \tag{3}$$

Apparently, in addition to the nominal controller mentioned above, the system will need an extra control term, namely, Δu to compensate for the existing uncertainties. Since ΔF and ΔB are not available, from a practical standpoint, ΔF and ΔB will be approximated by $\widehat{\Delta F}$ and $\widehat{\Delta B}$, respectively, as closely as possible. With this observation in mind, a general nonlinear compensation function extra to the nominal control is devised as follows:

$$\Delta u = u_f + u_b + u_s. \tag{4}$$

where u_f, u_b, u_s are given as $u_f = \widehat{\Delta F}$, $u_b = \widehat{\Delta B}x_r$, and $u_s = U(\mathbf{x}, x_r)sgn(s)$. Assume that $|u_f - \Delta F| \le \epsilon_f(\mathbf{x})$, $|u_b - \Delta B x_r| \le \epsilon_b(\mathbf{x}, x_r)$, and $U(\mathbf{x}, x_r) \ge \epsilon_f + \epsilon_b$, then the tracking errors will asymptotically converge to zero. Since $\widehat{F}$ and $\widehat{B}$ are hard to be approximated by using conventional control schemes, some intelligent control concepts need to be adopted. Moreover, since u_s is not continuous, undesirable chattering may appear due to excessive magnitude design of u_s, which therefore require a suitable u_s not only ensuring the system robustness but also the performance of the controlled system. Consequently, in this paper, an ARFC is proposed to perform exactly as such an intelligent nonlinear controller.

3 Control Algorithm

In this section, the ARFC will be introduced.

3.1 Fuzzy Knowledge Representation

From a practical point of view, a complex or ill-defined system is always represented by variables which are associated with some physical meanings such as the Mach number, the angle-of-attack in a flight system, or voltage, current in a circuit system. Hence, the system originally described by (1) may be represented as follows:

$$\begin{aligned} x^{(n)} &= f(z_{f_1}(x, \cdots, x^{(n-1)}), \cdots, z_{f_i}(x, \cdots, x^{(n-1)}), \cdots, \\ &z_{f_{n_f}}(x, \cdots, x^{(n-1)})) + b(z_{b_1}(x, \cdots, x^{(n-1)}), \cdots, \\ &z_{b_j}(x, \cdots, x^{(n-1)}), \cdots, z_{b_{n_b}}(x, \cdots, x^{(n-1)}))u(t), \end{aligned} \tag{5}$$

where z_{f_i} and z_{b_j} are variables of the system, $i = 1, 2, \cdots, n_f$ and $j = 1, 2, \cdots, n_b$, with n_f, n_b being the dimensions of $\mathbf{z}_f$ and $\mathbf{z}_b$, respectively. Given this notion, a fuzzy rule-based system is built.

As a general description of fuzzy knowledge representation [8], a fuzzy rule-base consists of a family of fuzzy *If-then* rules. We denote $\mathbf{z} = [z_1, z_2, \cdots, z_i, \cdots, z_m]^T$ as an input linguistic vector in the discourse universe $U_{\mathbf{z}}$ and $L_j = \{L_j^1, L_j^2, \cdots, L_j^{\alpha_j}, \cdots, L_j^{R_j}\}$ as a family of fuzzy sets associated with the membership functions $\mu_{L_j^{\alpha_j}}$ (see Fig. 1) with respect to the variable z_j, where $L_j^{\alpha_j}$ is a fuzzy set in L_j. Besides, the cores of the family of fuzzy sets, L_j, are denoted as $Z_j = \{Z_j^1, Z_j^2, \cdots, Z_j^{\alpha_j}, \cdots, Z_j^{R_j}\}$, where $Z_j^{\alpha_j}$ is a core satifying $Z_j^1 < Z_j^2 < \cdots < Z_j^{\alpha_j} < \cdots < Z_j^{R_j}$ (also see Fig. 1).

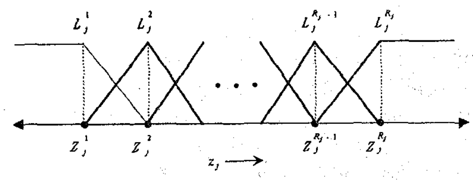

Fig. 1 Fuzzy enviroment of the input linguistic variable

Then, an example of the i-th fuzzy rule is represented as follows:

$$\begin{aligned} R[i]: \ \ If \ \ z_1 \ \text{is} \ L_1^{\alpha_1(i)} \cdots \ \text{and} \ z_j \ \text{is} \ L_j^{\alpha_j(i)} \cdots \\ and \ \ z_m \ \text{is} \ L_m^{\alpha_m(i)}, \ \ \text{then} \ y \ \text{is} \ Q^{\beta(i)}, \end{aligned} \tag{6}$$

where $L_j^{\alpha_j(i)} \in L_j$, y is an output linguistic variable, and $Q = \{Q^1, Q^2, \cdots, Q^\beta, \cdots, Q^{R_y}\}$ is a family of fuzzy sets associated with the membership functions μ_Q with respect to the output variable y, with Q^β being a fuzzy set in the family Q. And, let the cores of the family of fuzzy sets, Q, be denoted as $Y = \{Y^1, Y^2, \cdots, Y^\beta, \cdots, Y^{R_y}\}$, where Y^β is a core satisfying $Y^1 < Y^2 < \cdots < Y^\beta < \cdots < Y^{R_y}$. Generally, $\mathbf{z}$ is not necessarily identical to the state $\mathbf{x}$.

Furthermore, every rule is fired with a weighting function $\rho_i(\mathbf{z})$, which is determined by the membership functions and a compositional operator. Hence, $\rho_i(\mathbf{z})$ can be expressed as follows:

$$\rho_i(\mathbf{z}) = \begin{cases} \mu_{L_1^{\alpha_1(i)}}(z_1) \cdot \cdots \cdot \mu_{L_m^{\alpha_m(i)}}(z_m), \\ \qquad \text{if sup-product operator;} \\ min\{\mu_{L_1^{\alpha_1(i)}}(z_1), \cdots, \mu_{L_m^{\alpha_m(i)}}(z_m)\}, \\ \qquad \text{if sup-min operator.} \end{cases} \tag{7}$$

where $\mu_{L_j^{\alpha_j(i)}}(z_j)$ is denoted as a membership function, which is a positive function with $\mu_{Z_j^{\alpha_j(i)}}(z_j) \le 1$ as shown in Fig. 1 and reaches the maximum value when $z_j = Z_j^{\alpha_j(i)}$. In expression (7), the compositional operator is selected to be either the sup-product or the sup-min operator. Finally, using defuzzification function, the output y is expressed as follows:

$$y = \frac{\sum_{i=1}^{R_t} D_i(\rho_i(\mathbf{z}))Y^{\beta(i)}}{\sum_{i=1}^{R_t} D_i(\rho_i(\mathbf{z}))}, \tag{8}$$

where $D_i(\rho_i(\mathbf{z}))$ is denoted as a defuzzification function with the core $Y^{\beta(i)}$ for the i-th rule commonly expressed as follows:

$$D_i(\rho_i(\mathbf{z})) = \begin{cases} \rho_i(\mathbf{z}), & \text{if center-average-defuzzifier;} \\ \int_{-\infty}^{\infty} \mu_{Q^{\beta(i)}}(y)dy - \int_{y^{-*}}^{y^*}[\mu_{Q^{\beta(i)}}(y) - \rho_i(\mathbf{z})]dy, & \\ \qquad \text{if center-of-area defuzzifier;} \end{cases} \tag{9}$$

where y^{-*} and y^* are solution values of y satisfying $\rho_i(\mathbf{z}) = \mu_{Q^{\beta(i)}}(y)$ with $y^* \geq y^{-*}$ given the values $\mathbf{z}$ and R_t is the total number of rules generally equal to $R_1 \times R_2 \times \cdots \times R_m$. Then, we denote a fuzzy basis function $\xi_i(\mathbf{z})$ expressed as follows:

$$\xi_i(\mathbf{z}) = \frac{D_i(\rho_i(\mathbf{z}))}{\sum_{i=1}^{R_t} D_i(\rho_i(\mathbf{z}))} \tag{10}$$

so that equation (8) can be rewritten as:

$$y = \sum_{i=1}^{R_t} Y^{\beta(i)}\xi_i(\mathbf{z}) = \Theta^T\xi(\mathbf{z}), \tag{11}$$

where $\Theta = [Y^{\beta(1)}, Y^{\beta(2)}, \cdots, Y^{\beta(R_t)}]^T$ is regarded as a parameter vector and $\xi = [\xi_1, \xi_2, \cdots, \xi_{R_t}]^T$ is regarded as a regressor vector. Apparently, when R_t is large, the computation time and memory-space usage must be considered in real implementation.

As a result, we focus our attentions on computation time first, then the domain set of the fired rules and that of the total rules should be clearly distinguished. First, collect the indices of the fired rules as the following set:

$$I = \{i : \rho_i(\mathbf{z}) > 0\}, \tag{12}$$

where $\rho_i(\mathbf{z})$ is the weighting function of the i-th fired rules. Our aim is to figure out that the number of I, $\#n(I)$, is less than the number of total rule R_t during firing the fuzzy rules along with the expression (8) satisfied.

At the beginning, define the domain set of the total rules with respect to $\mathbf{z}$ as

$$E = \{\mathbf{z} : Z_j^1 \leq z_j \leq Z_j^{R_j}, j = 1, \cdots, m\}$$

Therefore, if $\mathbf{z}$ falls into E, then the fuzzy rules are fired to compensate the unknown functions. Partition this domain set into the finite $m - cells$, which are defined as

$$E_\alpha = \{\mathbf{z} : Z_j^{\alpha_j} \leq z_j \leq Z_j^{\alpha_j+1}, j = 1, \cdots, m\},$$

where $\alpha = \alpha_1 \times \cdots \times \alpha_j \times \cdots \times \alpha_m$ is a product index and satisfies $Z^\alpha \in Z$, $Z_j^{\alpha_j} < Z_j^{R_j}$. To union the all collection of E_α, we can obtain $E = \bigcup_{Z^\alpha \in Z} E_\alpha$.

A $\delta - box$ is defined as an equivalent of E_α,

$$\begin{aligned} \Omega_\alpha(Z^\alpha, \delta(\mathbf{z})) &= \{\mathbf{z} : Z_j^{\alpha_j} \leq z_j \leq Z_j^{\alpha_j} + \delta_j(z)\}, \\ &= E_\alpha \end{aligned}$$

where $j = 1, \cdots, m$, $\delta(\mathbf{z}) \in \Re^{m\times 1}$ and the j-th element of $\delta(\mathbf{z})$ is defined as $\delta_j(\mathbf{z}) = Z_j^{\alpha_j+1} - Z_j^{\alpha_j}$. Define the set of all corner points of δ-box Ω_α as $P_\alpha = \{Z^{c(\mathbf{z})} : c(\mathbf{z}) \in \prod_{j=1}^m \{\alpha_j, \alpha_j + 1\}\}$. Hence the number of P_α, $\#n(P_\alpha)$ is equal to 2^m.

Consider the membership function (Fig. 1):

$$\mu_{L_j^{\alpha_j}}(z_j) = \begin{cases} 1, & \text{as } z_j = Z_j^{\alpha_j}; \\ 0, & \text{as } z_j \geq Z_j^{\alpha_j+1} \text{ or } z_j \leq Z_j^{\alpha_j-1}; \\ x_j^{\alpha_j}, & x_j^{\alpha_j} \in (0,1), \text{ otherwise,} \end{cases} \tag{13}$$

for $\alpha_j = 1, 2, \cdots, R_j$ and $j = 1, 2, \cdots, m$, then the following proposition is given:

Proposition 1 *If the membership functions are given as eq. (12) and $\mathbf{z}$ is the linguistic vector, $\mathbf{z} \in E$, then exists a δ-box, $\Omega_\alpha(Z^\alpha, \delta(\mathbf{z}))$ such that $\mathbf{z} \in \Omega_\alpha(Z^\alpha, \delta(\mathbf{z}))$, and $\#n(I) \leq \#n(P_\alpha)$, where $\#n(I)$ and $\#n(P_\alpha)$ are the number of fired fuzzy rules and that of corner points of δ-box, respectively.*

Hence, the computation time is drastically reduced, especially, as R_t is very large.

3.2 Adaptive Robust Fuzzy Control

Referring to section 2, we use dynamical equation (5) to replace equation (1) so that a conventional control law is straightforwardly given as follows:

$$u(t) = \widehat{u} + u_f + u_b + u_s, \tag{14}$$

where $\widehat{u}$, u_f, u_b, u_s are defined as in section 2 and here expressed in terms of the suggested knowledge representation as $\widehat{u} = \widehat{F}(\mathbf{z}_b, \mathbf{z}_f) + \widehat{B}(\mathbf{z}_b)x_r + k_s s$, $u_f = \widehat{\Delta F}(\mathbf{z}_b, \mathbf{z}_f)$, $u_b = \widehat{\Delta B}(\mathbf{z}_b)x_r$, $u_s = |U_s(\mathbf{z}_b, \mathbf{z}_f, \mathbf{x}_r)|\, sgn(s) = (u_{s1}^* + u_{s2}^*|x_r|)sgn(s)$, with $u_{s1}^*, u_{s2}^* > 0$. Note that ARFC is designed to approximate u_f, u_b, u_s, as shown in Fig. 2, by first letting $F(\mathbf{z}_F) = -b^{-1}(\mathbf{z}_b)f(\mathbf{z}_f)$ where $\mathbf{z}_F$ is given as $\mathbf{z}_F = [\mathbf{z}_b^T\ \mathbf{z}_f^T]^T$, and $\mathbf{z}_s = [\mathbf{z}_F^T\ s]^T$.

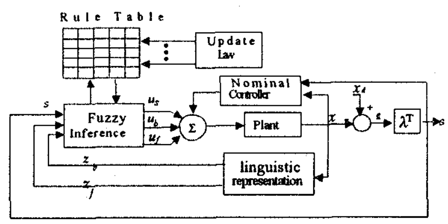

Fig. 2 The architecture of ARFC

Then, denote the input linguistic vectors as $\mathbf{z}_b$, $\mathbf{z}_F$, $\mathbf{z}_s$ and define the cores of the families of the fuzzy sets associated with $\mathbf{z}_b$, $\mathbf{z}_F$ and $\mathbf{z}_s$, respectively, as follows:

$$Z_{b_j} = \{Z_{b_j}^1, Z_{b_j}^2, \cdots, Z_{b_j}^{R_{bj}}\}, \quad \text{for} \quad j = 1, 2, \cdots, n_b$$

$$Z_{F_j} = \{Z_{F_j}^1, Z_{F_j}^2, \cdots, Z_{F_j}^{R_{Fj}}\}, \quad \text{for} j = 1, 2, \cdots, n_F$$

$$Z_{s_j} = \{Z_{s_j}^1, Z_{s_j}^2, \cdots, Z_{s_j}^{R_{sj}}\}, \quad \text{for} \quad j = 1, 2, \cdots, n_s$$

where $n_F = n_b + n_f$, and $n_s = n_F + 1$, whereby the control law can now be formally established as follows:

$$u_f = \theta_F = \sum_{i=1}^{m_F} \Theta_{F_i}\xi_{F_i}(z_F) = \Theta_F^T\xi_F(\mathbf{z}_F); \tag{15}$$

$$u_b = \theta_b x_r = \sum_{i=1}^{m_b} \Theta_{b_i}\xi_{b_i}(z_b)x_r = \Theta_b^T\xi_b(\mathbf{z}_b)x_r; \tag{16}$$

$$\begin{aligned} u_s &= \theta_{s1} + \theta_{s2}|x_r| = \sum_{i=1}^{m_s}(\Theta_{s1_i} + \Theta_{s2_i}|x_r|)\xi_{s_i}(\mathbf{z}_s) \\ &= \Theta_{s1}^T\xi_s(\mathbf{z}_s) + \Theta_{s2}^T\xi_s(\mathbf{z}_s)|x_r|, \end{aligned} \tag{17}$$

where θ_F, θ_b, θ_{s1}, θ_{s2} are denoted as output variables, Θ_F, Θ_b, Θ_{s1}, Θ_{s2} are denoted as parameter vectors, ξ_F, ξ_b, ξ_s are denoted as regressor vectors as in equation (11), and m_F, m_b, m_s are denoted as the total number of rules for u_f, u_b, and u_s, respectively. Our goal is to design optimal parameter vectors and regressor vectors such that the controlled system has minimal tracking errors and robust features. First of all, we must prove that bounds on the optimal approximation errors actually depend on the ARFC design, since that optimal parameter vectors are defined as follows:

$$\begin{aligned} \Theta_F^* &= arg\ min[sup_{\mathbf{z}_F \in E_F} \left|\Theta_F^T \xi_F(\mathbf{z}_F) - \Delta F\right|] \\ \Theta_b^* &= arg\ min[sup_{\mathbf{z}_b \in E_b} \left|\Theta_b^T \xi_b(\mathbf{z}_b) x_r - \Delta B x_r\right|] \\ \Theta_{s1}^* &= arg\ min[sup_{\mathbf{z}_s \in E_s} \Theta_{s1}^T \xi_s(\mathbf{z}_s) sgn(s) \geq u_{s1}^*] \end{aligned}$$

$$\Theta_{s2}^* = arg\ min[sup_{\mathbf{z}_s \in E_s} \Theta_{s2}^T \xi_s(\mathbf{z}_s) sgn(s) \left|x_r\right| \geq u_{s2}^* \left|x_r\right|]$$

where E_F, E_b and E_s are are expressed as follows:

$$\begin{aligned} E_F &= \{\mathbf{z}_F : Z_{F_j}^1 \leq z_{F_j} \leq Z_{F_j}^{R_{F_j}}, j = 1, 2, ..., n_F\} \\ E_b &= \{\mathbf{z}_b : Z_{b_j}^1 \leq z_{b_j} \leq Z_{b_j}^{R_{b_j}}, j = 1, 2, ..., n_b\} \\ E_s &= \{\mathbf{z}_s : Z_{s_j}^1 \leq z_{s_j} \leq Z_{s_j}^{R_{s_j}}, j = 1, 2, ..., n_s\}. \end{aligned}$$

To simplify the problem, some mild assumptions are given as $\Delta B(\mathbf{z}_b)$, $\Delta F(\mathbf{z}_F) \in C^1$(continuously differentiable), $\mid \Delta F(\mathbf{z}_F) \mid < F^U(\mathbf{z}_F)$, $\mid \Delta B(\mathbf{z}_b) \mid < B^U(\mathbf{z}_b)$ and $\mid \frac{dB(\mathbf{z}_b)}{dt} \mid \leq U_b$, where U_b is a constant. Based on the above assumptions, the following proposition will derive some useful results to be used in the sequel.

Proposition 2 *If the linguistic vectors $\mathbf{z}_F$ and $\mathbf{z}_b$ fall into the δ-boxes, $\Omega_\alpha(Z_F^\alpha, \delta_F(\mathbf{z}_F))$ and $\Omega_\beta(Z_b^\beta, \delta_b(\mathbf{z}_b))$, respectively, then the optimal approximation errors $\epsilon_F(\mathbf{z}_F)$ and $\epsilon_b(\mathbf{z}_b)$ will be bounded as follows: $|\epsilon_F| \leq g_F(\mathbf{z}_F)^T \delta_F(\mathbf{z}_F)$, $|\epsilon_b| \leq g_b(\mathbf{z}_b)^T \delta_b(\mathbf{z}_b)$, where $\epsilon_F(\mathbf{z}_F) = \Delta F - \Theta_F^{*T} \xi_F$ and $\epsilon_b(\mathbf{z}_b) = \Delta B - \Theta_b^{*T} \xi_b$; $g_F(\mathbf{z}_F) \in \Re^{n_F}$, $g_b(\mathbf{z}_b) \in \Re^{n_b}$, and their elements are defined as $g_{F_i} = sup_{\mathbf{z}_F \in \Omega_\alpha} [|\frac{\partial \Delta F(\mathbf{z}_F)}{\partial \mathbf{z}_{F_i}}|]$ for $i = 1, \cdots, n_F$, $g_{b_j} = sup_{\mathbf{z}_b \in \Omega_\beta} [|\frac{\partial \Delta B(\mathbf{z}_b)}{\partial \mathbf{z}_{b_j}}|]$ for $j = 1, \cdots, n_b$,*

Based on Proposition 2, the approximation errors can be reduced by adjusting δ_F and δ_b, and, hence, the robust controller u_s with a high gain is not necessary in ARFC. Furthermore, let the control law be redesigned as

$$u = d(t)(u_f + u_b + u_s) + (1 - d(t))U(\mathbf{z}_F, x_r)sgn(s) + \hat{u}, \quad (18)$$

where

$$k_s > \frac{1}{2}U_b \quad and \quad d(t) = \begin{cases} 1, & \text{as } \mathbf{z}_s \in Z_s^B; \\ 0, & \text{otherwise.} \end{cases}, \quad (19)$$

$$U(\mathbf{z}_F, x_r) = F^U(\mathbf{z}_F) + B^U(\mathbf{z}_b)|x_r| \quad (20)$$

with the update laws beging given as follows:

$$\begin{aligned} \dot{\Theta}_{F_i} &= r s_\Delta \xi_{F_i}(\mathbf{z}_F)\ as\ \mathbf{z}_F \in Z_F^B & (21) \\ \dot{\Theta}_{b_i} &= r s_\Delta \xi_{b_i}(\mathbf{z}_b) x_r\ as\ \mathbf{z}_b \in Z_b^B & (22) \\ \dot{\Theta}_{s_1 i} &= r s_\Delta \xi_{s_i}(\mathbf{z}_s)\ as\ \mathbf{z}_s \in Z_s^B & (23) \\ \dot{\Theta}_{s_2 i} &= r s_\Delta \xi_{s_i}(\mathbf{z}_s) |x_r|\ as\ \mathbf{z}_s \in Z_s^B, & (24) \end{aligned}$$

for some $r > 0$, and

$$\begin{aligned} s_\Delta &= \begin{cases} s - a & \text{as } s < a; \\ s - b, & \text{as } s > b; \\ 0, & \text{otherwise.} \end{cases} \\ \dot{s}_\Delta &= \dot{s}, & (25) \end{aligned}$$

where a, b are some constants and $a \leq Z_{s_{n_s}}^{\alpha_{n_s}} \leq 0 \leq Z_{s_{n_s}}^{\alpha_{n_s}+1} \leq b$.

Given the above result, we are now ready to state the closed-loop properties of the system (5) subject to our proposed ARFC in the following theorem.

Theorem 1 *If the control law and the update law are given as in equation (18) and as in equations (21)-(24), respectively, then the tracking errors will asymptotically converge to a neighborhood of zero.*

3.3 Two-Stage Design

From the analysis in subsection 3.2 , it is clear that the ultimate tracking errors depend on the ranges of the deadzone, and our objective is either to raise higher precision or to reduce the total number of rules. Hence, in addition to the optimal parameter vectors, optimal regressor vectors also need to be considered. Therefore, the choice of appropriate membership functions is essential for fuzzy controller design. Here we will choose the cores of fuzzy sets. In general, the cores of the relevant fuzzy sets are formed in two ways: (1) to use a data table; (2) via the form of function. For the former, the data set has to be memorized, which, however, requires extra memory-space. It even costs more computation time to find the irrelevant addresses of rules. Contrarily, the latter does not face the previous difficulties, and seems to be very hard in finding a suitable function form. Based on these fact, an alternative is to combine the two means together. Hence, we add a fuzzy rule-base before the previous control algorithm. This architecture is shown in Fig. 3, where a ramp function is used to determine the original cores of the family of the fuzzy sets.

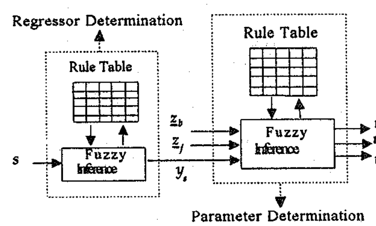

Fig. 3 The architecture of Two-Stage design

Apparently, it is very intuitive that if s is large, then Δ_s is large, whereas if s is small, then Δ_s is small. This concept lays down an efficient arrangement of the rule-space and simultaneously raises higher precision because of the smaller deadzone range. Hence, an example of i-th linguistic rule is represebted as follows:

$$R[i]: \quad \text{If } s \text{ is } LS^{(i)}, \text{ then } y_s \text{ is } LY_s^{(i)},$$

where y_s is an output linguistic variable and $LY_s = \{LY_s^1, LY_s^2, \cdots, LY_s^l\}$ is the family of the fuzzy sets with respect to y_s with the cores of the fuzzy sets being $Y_s = \{Y_s^1, Y_s^2, \cdots, \cdots, Y_s^l\}$. After the fuzzy rule-base is designed, we let y_s be an input linguistic vector to the second stage, and further let the cores of fuzzy sets of the second stage with ramp function form be as follows:

$$Y_s^{(new)} = \{1, 2, \cdots, R_s'\}. \quad (26)$$

Therefore, memory-space of the data set and the number of rule may be reduced in the case of $l < R_s' < R_{s_{n_s}}$.

3.4 Example

Consider a circuit system with the nonlinear uncertainties and the mathematical equations are given as follows:

$$\begin{aligned}
\dot{i}_L &= -\frac{R_L R_C + R(R_L + R_C)}{L(R+R_C)} i_L - \frac{R}{L(R+R_C)} v_C + \frac{u - v(R)}{L}, \\
\dot{v}_C &= \frac{R}{C(R+R_C)} i_L - \frac{1}{C(R+R_C)} v_C, \\
v_o &= \frac{R R_C}{R+R_C} i_L + \frac{R}{R+R_C} v_C, \\
|i_L| &\le 20 \quad and \quad |u| \le 200
\end{aligned}$$

where the coefficients are $C = 8 \times 10^{-6}$, $L = 1.2 \times 10^{-3}$, $R_C = 0.1$, and $R_L = 0.3$. Uncertainties R and $v(R)$ are assumed to be $R = R_0 + 0.5 R_0 sin(2\pi \times 0.1 v_o/156)$ and $v(R) = v_r sgn(R - R_0)$, where $R_0 = 120$, $k_s = 0.0001$ and $\lambda = 20000$. The desired trajectory is given as $v_{od} = 156 sin(2\pi \times 60t)$ and our objective is to force the output v_o to follow this desired trajectory. A nominal model is estimated by opening the circuit system at the terminals of loading R, then ARFC will approximate the unknown nonlinear function of $R(v_o)$ and $v(R)$. To demostrate the robustness of ARFC with respect to the uncertainties $v_r = 10$ are considered in our simulation.

Let the linguistic vectors be given as $\mathbf{z}_b = \mathbf{z}_F = [v_o, i_L]^T, \mathbf{z}_s = [v_o, s]^T$. At the beginning, we use the conventional adaptive sliding mode control that is identical to the case of total number of fuzzy rules of u_F, u_b, u_{s1}, u_{s2} set to 1, i.e., only one paprmater is given and the fuzzy basis function is always equal to 1. Secondly, let the number of fuzzy rules of u_F, u_b, u_{s1}, u_{s2} be given as 21×21 respectively, and $\delta_F, \delta_b, \delta_s$ be constant. Then, using the two-stage design of ARFC in subsection 3.3, the rule number is reduced to 9×9 , and the forward input data-table is given as $\{-s_{max}, -0.08 s_{max}, 0, 0.08 s_{max}, s_{max}\}$ whereas the output data-table is given as $\{1, 4, 5, 6, 9\}$. The membership functions, compositional operator and defuzzification function are selected to be triangular form, sup-min operator and center of area in these cases respectively. The deadzone range is given as $[\alpha, \beta] = [-0.1 s_{max}, 0.1 s_{max}]$ in these cases, where s_{max} is defined as the largest core with respect to linguistic rule: If s is *very large*. Simulation is performed by using the two loop structure to accurately approximate the real system. The sampling time of control servo is given as 10 μ sec in outer loop, and inner loop is to simulate the dynamic equation with fixed step size 1 μ sec. Fig. 4 shows the results of simulation in the cases of $v_r = 10$. At the beginning, we only use *PD* controller to compensate for the uncertainties in the first periods of sine wave, then incorporate the controllers metioned above. Fig. 4a shows the responses of the conventional adaptive sliding control with a boundary layer given.

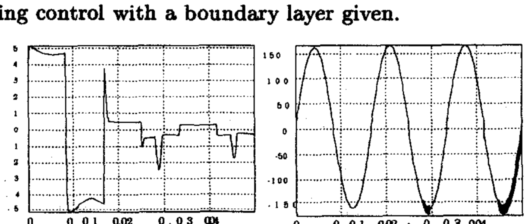

Fig. 4a Results of adaptive sliding mode controller

An unsatisfactory tracking error along with a chattering of control input voltage appears and eventually the controlled system will become unstable. This phenomenon may be caused by the excessively large control gains which require the faster sampling time of control servo to respond to. Conceivally, the ARFC can be free from this type of drawback as shown in the Fig. 4b. Furthermore, Fig. 4c shows the two-stage design architecture with a similar performance but less memory-space for storing fuzzy rules.

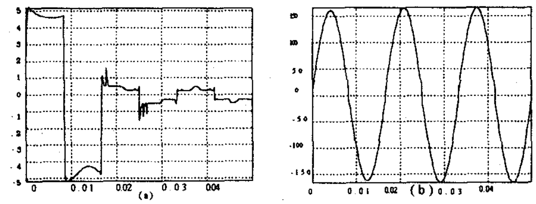

Fig. 4b Results of ARFC

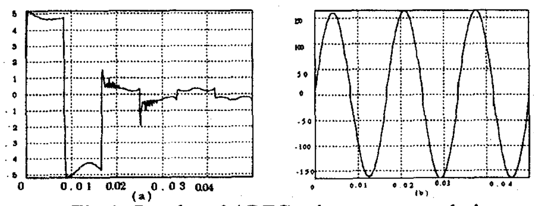

Fig 4c Results of ARFC using two-stage design

Fig. 4: (a) tracking error (b) control input voltage u

4 Conclusions

We presented a novel architecture for fuzzy control and applied it to a class of nonlinear systems with uncertainties. The proposed control scheme avoided the excessively large control gains and guaranted the global stability of the system with the feature in fast converge of tracking errors. In addition to the circuit system mentioned above, applications to robot manipulators is also studied in the past work [14]. Furthermore, an application to a five degree-of-freedom (DOF) articulated robot arm is also actually implemented in real-time at present work.

References

[1] K. M. Passino, "Bridging the gap between Conventional and Intelligent Control', *IEEE Cont. Syst. Magaz., vol. 13, no. 3, pp. 12-18, 1993*

[2] R. M. Sanner and J-J E. Slotine, "Gaussian Networks for Direcet Adaptive Control", *IEEE, Trans. on Neural Networks, vol. 3, no. 6, pp. 837-863, 1992.*

[3] R. M. Sanner and J-J E. Slotine, "Enhanced Algorithms for Adaptive Control Using Radial Gaussian Networks', *American Control Conference, 1992*

[4] A. U. Levin. and K. S. Narendra, "Control of Nonlinear Dynamical Systems Using Neural Networks: Controllability and Stabilization', *IEEE, Trans. on Neural Networks, vol. 4, no. 2, pp. 192-205, 1993.*

[5] K. S. Narendra and K. Parthasarathy, "Gradient Methods for the Optimisation of dynamical Systems Containing Neural Networks", *IEEE, Trans. on Neural Networks, vol. 2, no. 2, pp. 252-262, 1991.*

[6] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE, Trans. on Neural Networks, vol. 1, no. 1, pp. 4-27, 1990.*

[7] W. Messner, R. Horowitz, W. W. Kao and M. Boals, "A New Adaptive Learning Rule', *IEEE, Trans. on Automatic Control, vol. 36, no. 2, pp. 188-197, 1991.*

[8] L. X. Wang, "Stable Adaptive Fuzzy Control of Nonlinear Systems', *IEEE, Conf. on Decision and Control, 1992*

[9] L. X. Wang and J. M. Mendel, "Generating Fuzzy Rules by Learning from Examples', *IEEE, Trans. on Syst., Man, Cybern. vol. vol. 22, no. 6, pp. 1414-1427, 1992.*

[10] J. S. Jang, "Self-Learning Fuzzy Controllers Based on Temporal Back Propagation', *IEEE, Trans. on Neural Networks, vol. 3, no. 5, pp. 714-723, 1992.*

[11] C. T. Lin and C. S. G. Lee, "Neural-Network-Based Fuzzy Logic Control and Decision Systems", *IEEE, Trans. on Computers, vol. C-40, no. 12, pp. 1320-1336, 1991.*

[12] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*, Prentice Hall, pp.19 1989.

[13] J-J E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice Hall, pp. 278-284, 1991.

[14] F. Y. Hsu and L. C. Fu, " Adaptive Robust Fuzzy Control for Robot Manipulators", *IEEE Conference on Robotics and Automation, pp. 629-334, 1994*