ONLINE NEURAL NETWORK ADAPTIVE CONTROL OF A CLASS OF NONLINEAR SYSTEMS USING FUZZY INFERENCE REASONING

MOHAMED BAHITA[1] , KHALED BELARBI[2]

Key words: Fuzzy inference reasoning, Neural network adaptive control, Certainty equivalence feedback linearization, Nonlinear instable system.

This study addresses the proposition of neural network (NN) adaptive control for a class of nonlinear systems using fuzzy reasoning. In first step, an ideal control law is established based on feedback linearization technique and certainty equivalence. Then the NN system is introduced on line to approximate this ideal control law. The parameters of the NN system are on-line adapted and changed according to the gradient descent law, which will be approximated in part by a fuzzy inference system. In other words, instead of using the popular back propagation technique. we use an on-line simple fuzzy inference system to approximate part of the gradient descent resulting adaptation law.

1. INTRODUCTION

To model and control nonlinear systems with any desired accuracy, intelligent systems (IS) have been suggested as alternative approaches to conventional techniques in many cases. The most commonly applied methods are neural networks (NNs) and fuzzy logic systems (FLSs).

For the first method, active research has been carried out on NNs control for nonlinear systems [1–8]. In [1] for example, the paper studies the problem of learning from adaptive neural network (NN) control of a class of nonaffine nonlinear systems in uncertain dynamic environments. In [2], the authors use a NN as a controller for on board tracking platform. In [3–5], the authors use a radial basis function (RBF) NN type to estimate the model uncertainties, compensate external disturbances and emulate unknown parameters. In [6], the authors use an adaptive RBF network control algorithm to estimate the unknown nonlinear functions to overcome the necessity for the mathematical model. In [7], an RBF-NN is applied

[1] University of Constantine 3,Faculty of Chemical Engineering, Constantine, Algeria, E-mail: mbahita@yahoo.fr

[2] University of Constantine 1, Faculty of Engineering, Route de Ain el Bey, Constantine, Algeria

successively for online stable identification and control of nonlinear discrete-time systems. Concerning the second method, it has been shown that FLSs are a good tool for approximating any continuous nonlinear function to any desired accuracy over a compact set [8], and for a detail we can refer to a survey papers [9, 10].

These two methods (FLSs and NNs) play a role of function approximators in the adaptation mechanism of adaptive control in most cases. Adaptive control of dynamic systems has been an active area of research since the 1960s [11]. With the combination of adaptive control and feedback linearization, most of control problems have been resolved [12–14], by transforming a nonlinear system into a linear one, then linear control methods can be applied. But in most cases, the resulting control law contains unknown nonlinear functions. FLSs in particular have been widely used to approximate these unknown nonlinear functions [15, 16].

In this paper, NN adaptive control for a class of nonlinear systems is proposed on simulation study based on fuzzy reasoning. In first step, an ideal control law is established based on certainty equivalence approach and feedback linearization technique [17]. Then, the NN system is introduced on-line to approximate this ideal control law. The parameters of the NN system are on-line adapted and changed according to the gradient descent law. In general the NN parameters adaptation is based on the tracking error signal, and the back-propagation algorithm is used for solving this problem. In our work, instead of the tracking error, the control error is used, and instead of the back-propagation algorithm, we use a simple on-line FLS of Mamdani type [10] to approximate this control law. The used NN controller is of radial basis function (RBF) type with on-line adapted centres using the k-means algorithm [18, 19]. The advantages in using this type of network is to minimise the number of adapted parameters and then to avoid the slow convergence as in a multilayered Perceptron (MLP) mostly used in the literature, especially when it contains more than one hidden layer. As a consequence: minimising the computations time. The present work is an extended version of the work presented in the international symposium on robotics and intelligent sensors [20].

This work is organised as follows, in section 2, we describe the class of nonlinear systems under study and the feedback linearization control law based on the concept of certainty equivalence approach. In section 3, structure and properties of the direct adaptive neural controller DANC and the FLS approximator are explained. In section 4, the proposed method is used to control the nonlinear unstable inverted pendulum system.

2. PROBLEM FORMULATION

The system under study is an *n*th-order affine [17] nonlinear described by:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = x_3$$

$$\dot{x}_n = f(x_1, x_2, \ldots x_n) + g(x_1, x_2, \ldots x_n).u\,, \tag{1}$$

$$y = x_1\,, \tag{2}$$

where $\mathbf{\underline{x}} \in R^n$ is the state vector, $u \in R$ is the control input and $y \in R$ is the output of the system, $f(\underline{x})$ and $g(\underline{x})$ are unknown but bounded nonlinear functions. We assume that the state vector $\underline{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}} = (x, \dot{x}, \ldots x^{(n-1)})^{\mathrm{T}} \in R^n$ is available for measurement. In order for (1) to be controllable, it is required that the function $g(\underline{x}) \neq 0$ (an usual assumption [9]). Define the reference signal as $y_m(t)$, and we assume that its derivatives $y_m^{(1)}(t), \ldots, y_m^{(n)}$, exist and are bounded. The tracking error e and the error vector $\underline{e}$ will be defined as

$$e = y_m - y = y_m - x_1\,, \tag{3}$$

$$\underline{e} = (e, \dot{e}, \ldots, e^{(n-1)})^{\mathrm{T}} \in R^n\,. \tag{4}$$

Let the vector $\underline{k} = (k_0, k_1, \ldots, k_{n-1})^{\mathrm{T}} \in R^n$ be such that the polynomial $h(s) = s^n + k_{n-1}.s^{n-1} + \ldots\ldots + k_0 = 0$ has all its roots strictly in the left-half complex plane. If the functions $f(\underline{x})$ and $g(\underline{x})$ are known, then based on the certainty equivalence approach [17, pp. 207–275], the feedback linearization control law u noted $u^\bullet$ is derived as

$$u^\bullet = \frac{1}{g(\underline{x})}.(y_m^{(n)} + K^T.\underline{e} - f(\underline{x})). \tag{5}$$

Substituting (5) in (1), we obtain

$$e^{(n)} + k_{n-1}e^{(n-1)} + \ldots\ldots + k_0 e = 0\,, \tag{6}$$

where the main objective of control is $\lim\, e(t) \to 0$ when $t \to \infty$. However $f(\underline{x})$ and $g(\underline{x})$ are unknown, we have to design a NN of RBF type controller with output $u = u_c(\underline{x}, \underline{\theta})$ to approximate the feedback linearization control law of (5), with an adaptation law to adjust the NN-RBF parameters vector $\underline{\theta}$.

3. THE DIRECT ADAPTIVE NEURAL CONTROLLER (DANC) BASED ON FUZZY REASONING

In this section, we develop the DANC using an RBF-NN, and the adaptation law is derived. At the end of this part, the fuzzy logic system FLS (fuzzy reasoning) is explained.

3.1. THE DIRECT ADAPTIVE NEURAL CONTROLLER (DANC)

The design of an RBF-NN consists of three separate layers. The first layer is the input layer. The second layer is the hidden layer. The last layer gives the output of the network. The general RBF-NN model has a linear transformation from the hidden layer to the output layer. In other words, the output depends linearly on the weights. More explicitly, the output of an RBF-NN system can be put in the following form

$$u_c(\underline{x},\underline{\theta}) = \underline{\theta}^{\mathrm{T}}.\underline{\xi}(\underline{x}) \quad = \sum_{i=1}^{nr} \xi_i \theta_i \quad , \text{ with } \xi_i = \psi(\|\underline{x} - c_i\|_2), \tag{7}$$

$\underline{x}$ is the input vector, ψ is a non linear function called radial basis function (RBF), $\underline{\theta}$ are connections weights to be adapted (parameters) between the hidden layer and the output layer, so the vector $\underline{\theta}^{\mathrm{T}} = [\theta_1^{\mathrm{T}}\ \theta_2^{\mathrm{T}} \ldots \theta_{nr}^{\mathrm{T}}]$ contains all adjustable parameters and $\underline{\xi}(\underline{x})$ is a vector of radial basis functions (RBFs). c_i are centres of basis functions and *nr* is the number of basis functions. Gaussian RBF is employed frequently in neural networks, since it is bounded, strictly positive and smooth.

As cited above, the main objective of control is lim $e(t) \to 0$, when $t \to \infty$. The parameters update will be designed so as to minimize the control error e_u between the feedback linearization control law $u^{\bullet}$ (the expected ideal control law that ensures an ideal response) which is not available and the output $u_C(\underline{x},\underline{\theta}) = \underline{\theta}^{\mathrm{T}}.\underline{\xi}(\underline{x})$ of the RBF-NN controller (approximating this feedback linearization control law $u^{\bullet}$), then

$$e_u = u^{\bullet} - u_C(\underline{x},\underline{\theta}). \tag{8}$$

So, one corresponding form of the cost function minimizing the control error e_u is

$$\min J = \frac{e_u^2}{2}. \tag{9}$$

Based on the gradient descent law: $\dot{\underline{\theta}} = -\gamma \dfrac{\partial J}{\partial \underline{\theta}}$, (10)

$\gamma > 0$ is the learning rate, so, $\dfrac{\partial J}{\partial \underline{\theta}} = -e_u \dfrac{\partial u_C}{\partial \underline{\theta}}$. (11)

In (11), the term $\dfrac{\partial u_C}{\partial \underline{\theta}}$ can be obtained from (7). It remains just to determine e_u.

As mentioned above, in our work, instead of the tracking error, the control error is used, and instead of the back-propagation algorithm, we use a simple on-line FLS of Mamdani type [10] to approximate this control law $u^{\bullet}$ by approximating the control error $e_u = u^{\bullet} - u_c(\underline{x},\underline{\theta})$ as a hall. We replace e_u in (11) by its estimate $\alpha\hat{e}_u$ which is the output of a FLS of Mamdani type [10], with $\alpha > 0$. Then (10) can be written as:

$$\dot{\underline{\theta}} = \gamma\,\alpha\,\hat{e}_u \frac{\partial u_c}{\partial \underline{\theta}} = \gamma'\,\hat{e}_u\,\xi(\underline{z})\,, \tag{12}$$

with $\gamma' = \alpha\gamma > 0$ being the new step size. Under the condition that this step size is small enough and in the worst case, the cost function will converge to a local minimum of (9) [21, pp. 84–104], hence e_u will remain bounded throughout the search procedure. To ensure boundedness of the weights $\underline{\theta}$, the adaptation law (12) must be modified by the so-called e-modification [11, pp. 554–633]:

$$\dot{\underline{\theta}} = \gamma'\hat{e}_u.\xi(\underline{z}) - \gamma'\left|\hat{e}_u\right|.v_0\,\underline{\theta}\,, \tag{13}$$

$v_0 > 0$ is a design constant.

We give the overall scheme of the direct adaptive neural controller (DANC) with the fuzzy logic system (FLS) and the adaptation mechanism as shown in Fig. 1.

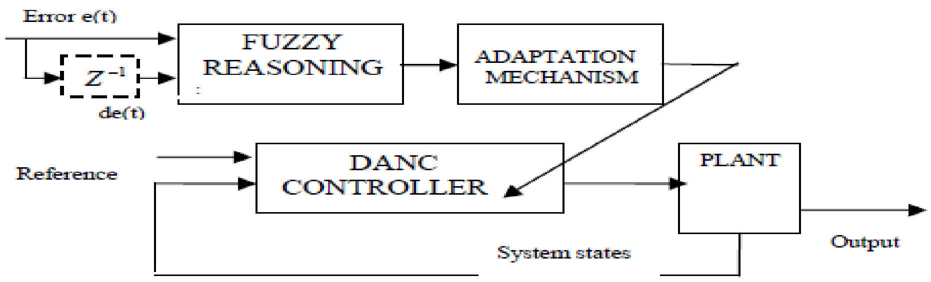

Fig. 1 – Structure of the DANC-FLS.

3.2. THE FUZZY LOGIC SYSTEM (FLS)

In this section, we explain the basic idea used to construct the FLS for obtaining an estimation $\hat{e}_u$ of the control error e_u. We have the following situations:

Situation 1: The control signal is “correct" in the sense that it is driving the output towards the reference: this occurs when the tracking error is maintained at zero or when it is decreasing.

Situation 2: The control signal is not “correct” when the output is drifting away from the reference. In this case, two cases may arise:

I – The output may be drifting away from the reference from above, in which case the magnitude of the applied control signal $u_C(\underline{x},\underline{\theta})$ is larger than $u^{\bullet}$, $u_C(\underline{z},\underline{\theta}) > u^{\bullet}$ and thus $u^{\bullet} - u_C(\underline{x},\underline{\theta}) < 0$.

II – The output may be drifting away from below, and the magnitude of the applied control signal is less than $u^{\bullet}$, $u_C(\underline{z},\underline{\theta}) < u^{\bullet}$ and thus $u^{\bullet} - u_C(\underline{z},\underline{\theta}) > 0$.

Using this reasoning, we are ready to create a rule base (RB) to estimate the control error e_u, where the output of the FIS is the crisp value of the estimated control error. The crisp input variables are the current tracking error *e(t)* and the change of the tracking error $de(t) = e(t) - e(t-1)$. Introducing the fuzzy variables: *ERROR(e), VARIATION OF ERROR(de)* and *CONTROL ERROR* ($\hat{e}_u$) each taking three fuzzy values: *ZERO(Z), NEGATIVE(N) and POSITIVE(P)*. We obtain the rule base (RB):

Case 1 – the control signal is correct when the tracking error is zero or decreasing; this implies the following rules:
If *ERROR is ZERO* AND *VARIATION OF ERROR is ZERO*
If *ERROR is POSITIVE* AND *VARIATION OF ERROR is NEGATIVE*
If *ERROR is NEGATIVE* AND *VARIATION OF ERROR is POSITIVE*
then *CONTROL ERROR is ZERO*

Case 2 – the output is drifting away from above; this induces the following rules:
If ERROR is ZERO AND VARIATION OF ERROR is NEGATIVE
If ERROR is NEGATIVE AND VARIATION OF ERROR is (NEGATIVE OR ZERO)
then CONTROL ERROR is NEGATIVE

Case 3 – the output is drifting away from below; this induces the following rules:
If ERROR is ZERO AND VARIATION OF ERROR is POSITIVE
If ERROR is POSITIVE AND VARIATION OF ERROR is (ZERO OR POSITIVE)
then CONTROL ERROR is POSITIVE.

For the detail, we can refer to [20]. We have also to chose the shape of all the membership functions and their distribution on the universe of discourse as shown in Fig. 2, where c_N, c_Z, c_P, are the points where the membership functions reach their maximum. After fuzzification and *(prod, max)* inference strategy, the crisp estimated control error $\hat{e}_u$ is computed through center of gravity defuzzification formula.

4. **SIMULATION RESULTS**

In this section, we test the performance of the proposed DANC-RBF with fuzzy reasoning on the inverted pendulum system [22] depicted in Fig. 3 and described by the following dynamical equations:

$$\begin{aligned}&\dot{x}_1 = x_2\\&\dot{x}_2 = f(\underline{x}) + g(\underline{x})\cdot u(t), \quad f(\underline{x}) = \frac{g\sin(x_1) - \dfrac{mlx_2^2\cos(x_1)\sin(x_1)}{M+m}}{l\left(\dfrac{4}{3} - \dfrac{m\cos^2(x_1)}{M+m}\right)},\\&y = x_1\end{aligned}$$

$$g(\underline{x}) = \frac{\dfrac{\cos(x_1)}{M+m}}{l\left(\dfrac{4}{3} - \dfrac{m\cos^2(x_1)}{M+m}\right)}, \tag{14}$$

$x_1 = \theta$ is the angular position of the pendulum (see Fig. 3), $x_2 = \dot{\theta}$ is the angular velocity of the pendulum. We use $g = 9.8\ \text{m/s}^2$, $M = 1\ \text{kg}$ is the mass of the cart, $m = 0.1\ \text{kg}$ is the mass of the pole and $l = 0.5\ \text{m}$ is the half length of the pole. The control objective is to make the pole of the pendulum track a sine wave trajectory $y_m = \theta_m = AM\sin(t)$ with different amplitudes AM, and we terminate by balancing the pole to the vertical position $(x_1, x_2) = (0,0)$, i.e., $AM = 0$. Clearly, the derivatives of the reference y_m exist and are bounded. The parameters are chosen as $\gamma = 9$, $v_0 =$ 0.005, step size $\mathrm{d}t = 0.01$, and $k = [k_0\ \ k_1]^{\mathrm{T}} = [5\ \ 5]^{\mathrm{T}}$ in order to have all roots of $s^2 + k_1.s + k_0 = 0$ in the open left-half plane.

The RBF controller has five radial basis functions. The parameters $\underline{\theta}$ are initialised to 0. The centres of the basis functions in the RBF network are uniformly distributed in the interval [–1; 2] and are adjusted using the k-means algorithm [18, 19]. The RBF network has two inputs $\underset{\rightarrow}{x} = \underline{z} = [z_1\ \ z_2] = [\theta\ \ \ (K^{\mathrm{T}}\underline{e} + \theta_m^{(2)})]$ with $\underline{e} = [\theta_m - \theta\ \ \ \dot{\theta}_m - \dot{\theta}]$. The used basis functions are Gaussian functions under the form

$$\psi(r) = \exp\left(\frac{-r^2}{2\sigma^2}\right), \tag{15}$$

with $r = \|\underline{x} - c_i\|_2$ and a width $\sigma = 0.5$. The following initial conditions for the inverted pendulum $(x_1(0),\ x_2(0))^{\mathrm{T}} = (-\ 0.2$ rad, 0 rad/s$)$ are used in the simulation. The fuzzy estimator of the control error has rule base (RB) as described in section 3.2. The membership functions of the fuzzy sets are shown in Fig. 2. The corresponding parameters defining these membership functions are set to: $c_N = -0.5$, $c_Z = 0$, $c_P = 0.5$ for the tracking error, $c_N = -3.5$, $c_Z = 0$, $c_P = 3.5$ for the variation of error and $c_N = -3.5$, $c_Z = 0$, $c_P = 3.5$ for the estimated control error. The simulation results for different amplitudes of the reference signal are shown in Figs. 4 to 7. The system output $y(t)$ (pole angle) is in dotted while the reference signal $y_m(t)$ is in continuous. Figure 4 shows the response curve of the pole angle from the initial position (– 0.2, 0) and the corresponding desired values with amplitudes $AM = \pi/30$ during the time interval $t \in [0,\ 12.5]$ s, and $AM = \pi/15$ during the time interval $t \in [12.5,\ 25]$ s, and finally with amplitude $AM = 0$ which represents a regulation case during the remaining time interval $t \in [25,\ 30]$ s. Figure 5 shows the tracking error converging rapidly to a value close to zero. Figure 6 represents the corresponding control input which peaks at $t = 12.5$ s (the first amplitude variation from $AM = \pi/30$ to $AM = \pi/15$), and at $t = 25$ s (the second amplitude variation from $AM = \pi/15$ to $AM = 0$). Figure 7 shows that the crisp estimated error provided by the fuzzy estimator is smooth, confirming the smoothing property of DANC-RBF-FLS system. It also remains bounded and converges quite rapidly to a value close to zero.

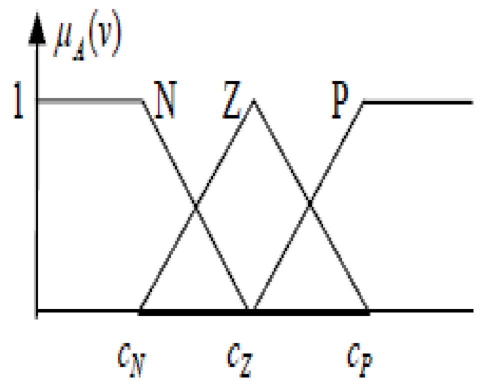

Fig. 2 – Distribution of membership functions. on the universe of discourse.

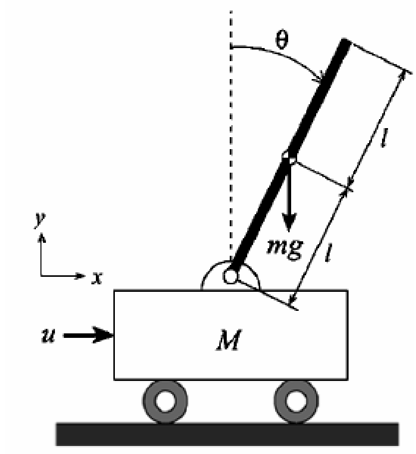

Fig. 3 – The inverted pendulum system.

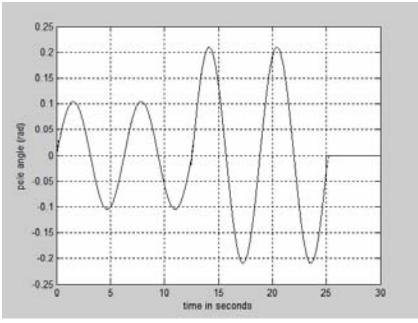

Fig. 4 – The pole angle $y = \theta$ (---)and the desired reference trajectory y_m (___).

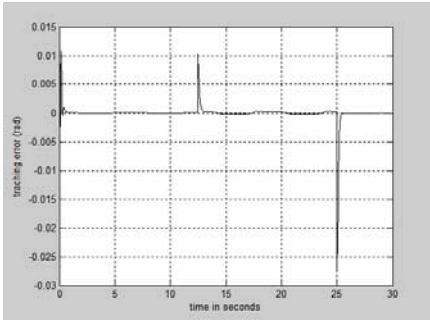

Fig. 5 – The corresponding tracking error e .

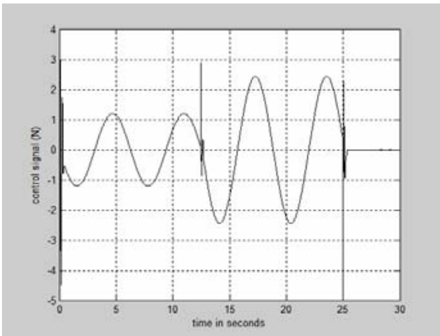

Fig. 6 – The control signal u .

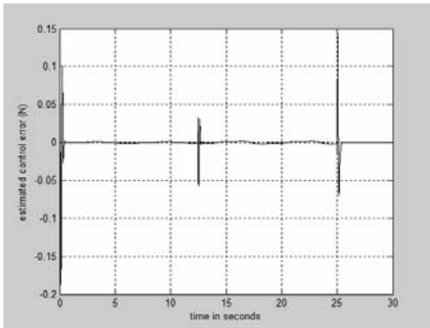

Fig. 7 – The corresponding estimated control error $\hat{e}_u$ using the RBF-Mamdani controller.

From these figures, we can see that the controlled system behaves well in all situations (tracking and regulation cases).

5. **CONCLUSION**

This paper introduces a direct adaptive neural controller (DANC) for an affine unknown nonlinear systems using fuzzy reasoning. An RBF-NN system is used on-line to approximate a feedback linearization control law based on the certainty equivalence approach. The parameters of the RBF-NN controller are on-line adjusted based on the gradient descent law minimizing the control error cost. Instead of using the back propagation technique, an FLS is used to estimate the control error appearing in the adaptation law. The algorithm was successfully tested with better performances to on-line control the inverted pendulum system.

Received on September 9, 2013

REFERENCES

1. S. L. Dai, C. Wang, M. Wang, *Dynamic Learning From Adaptive Neural Network Control of a Class of Nonaffine Nonlinear Systems,* IEEE Transactions on Neural Networks and Learning Systems., **25**, *1*, pp. 111–123, 2014.
2. O. G. Müler, *A Neural controller for on board tracking platform*, Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **54**, *2*, p. 213–222, 2009.
3. J. L. Tao, Y. Yang, D. H.Wang, C. Guo, *A robust adaptive neural networks controller for maritime dynamic positioning system*, *Neurocomputing.*, **110**, *1*, pp. 128–136, 2013.
4. J. Liu, Y. Lu, *Adaptive RBF neural network control of robot with actuator nonlinearities*, Journal of Control Theory and Applications., **8**, *2*, pp. 249–256, 2010.
5. J. Fei, H. Ding, *Adaptive sliding mode control of dynamic system using RBF neural network*, NonlinearDynamics, **70**, *2*, pp. 1563–1573, 2012.
6. G. Xia, X. Shao, A. Zhao, H. Wu, *Adaptive Neural Network Control with Backstepping for Surface Ships with Input Dead-Zone*., Mathematical Problems in Engineering., Article ID 530162, 2013.
7. N. S. Bhuvaneswari, G. Uma, T.R. Rangaswamy, *Adaptive and optimal control of a non-linear process using intelligent controllers*, Applied Soft Computing., **9**, pp. 182–190, 2009.
8. J. L. Castro, Fuzzy logical controllers are universal approximators, IEEE Trans. Syst., Man, Cybern., **25**, pp. 629–635, 1995.
9. G. Feng, *A Survey on Analysis and Design of Model-Based Fuzzy Control Systems*, Rev. IEEE Trans. Fuzzy Syst., **14,** *5*, pp. 676–697, 2006.
10. R. E. Precup, H. Hellendoorn, *A survey on industrial applications of fuzzy control,* Computers in Industry, **62**, pp. 213–226, 2011.
11. P. A. Ioannou, J. Sun, *Robust Adaptive Control*. Prentice-Hall., 1996.
12. S. S. Sastry, A. Isidori, *Adaptive control of linearization systems*, IEEE Trans. Automat. Contr., **34**, pp. 1123–1131, 1989.
13. R. Marino, P. Tomei, *Globally adaptive output-feedback control on nonlinear systems, part I: Linear parameterization*, IEEE Trans. Automat. Contr., **38**, pp. 17–32, 1993.
14. R. Marino and P. Tomei, *Globally adaptive output-feedback control on nonlinear systems, part II: Nonlinear parameterization*, IEEE Trans. Automat. Contr., **38**, pp. 33–48, 1993.
15. B. Chen, X. Liu, K. Liu, C. Lin, *Fuzzy-Approximation-Based Adaptive Control of Strict-Feedback Nonlinear Systems With Time Delays*, IEEE Trans. Fuzzy Syst., **18,** *5*, pp. 883–892, 2010.
16. Y. Chien, W. Wang, Y. Leuk, T. Lee, *Robust Adaptive Controller Design for a Class of Uncertain Nonlinear Systems Using Online T–S Fuzzy-Neural Modeling Approach*, IEEE Trans. SMC-B., **41,** pp. 542–552, 2011.
17. J. J. E. Slotine. L. Weiping, *Applied Nonlinear control*, Prentice Hall, 1991.
18. C. Darken, J. Moody, *Fast Adaptive k-means Clustering: Some empirical Results*, International Joint conf on Neural Networks., **2**, pp. 233–238, 1990.
19. S. Chen, S. A. Billings, P. M. Grant, *Recursive hybrid algorithm for nonlinear systems identification using Radial Basis Function Networks*, International journal of Control., **55**, pp. 1051–1070, 1992.
20. M. Bahita and K. Belarbi, *Radial Basis Function Controller of a Class of a Class of Nonlinear Systems Using Mamdani Type as a Fuzzy Estimator*, Procedia Engineering, Engineering., **41**, pp. 501–509, 2012.
21. M. Minoux, *Mathematical programming: theory and algorithms,* John Wiley.
22. Y. Lee and H. Zak, *Uniformly ultimately bounded fuzzy adaptive tracking controllers for uncertain systems*, IEEE Trans. Fuzzy Syst., **12**, *6*, pp. 797–811, 2004.