International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:9, 2010

Fuzzy rules emulated network adaptive controller with unfixed learning rate for a class of unknown discrete-time nonlinear systems

Chidentree Treesatayapun

Abstract—A direct adaptive controller for a class of unknown nonlinear discrete-time systems is presented in this article. The proposed controller is constructed by fuzzy rules emulated network (FREN). With its simple structure, the human knowledge about the plant is transferred to be if-then rules for setting the network. These adjustable parameters inside FREN are tuned by the learning mechanism with time varying step size or learning rate. The variation of learning rate is introduced by main theorem to improve the system performance and stabilization. Furthermore, the boundary of adjustable parameters is guaranteed through the on-line learning and membership functions properties. The validation of the theoretical findings is represented by some illustrated examples.

Keywords—Neuro-Fuzzy, learning algorithm, nonlinear discrete-time.

I. Introduction

Adaptive controllers for a class of nonlinear discrete-time have been received more efforts recently. Under the assumption that only linear parameters are all unknown, the successful adaptive schemes have been introduced in [1] and its companion article [2]. Recently, with some applications of artificial neural networks, the nonlinearities have been assumed to be unknown. In the case of indirect adaptive controllers, neural networks have been implemented as nonlinear system identifications to support control algorithms for the unknown constrains and plants. The linearization feedback controller has been developed in [3] for unknown nonlinear discrete-time systems. The control algorithm has been constructed by neural networks linearization models. Thus the system performance and the stability analysis have been related to the models accuracy. To ensure the system performance, in the case of NN controllers, the adaptation algorithm with a robust term has been attempted in [4]. This controller has been designed for systems which have output linear with the control effort. The stabilized tracking control based on NN has been introduced in [5]. Only unknown nonlinearities have been assumed to be unknown thus high-order neural networks have been employed to approximate.

In the adaptation, small learning rates or step sizes are often used in gradient search methods because of the system stability [6]. On the other hand, the small step size can slow down the reaching solution. Usually, these learning rates are all fixed as the suitable constants which can be obtained by some algorithms such as [7] for adaptive controllers based on neural networks.

In this article, however, a direct adaptive controller for a class of unknown nonlinear discrete-time systems is developed with the associate of the fuzzy rule emulated network or FREN. The initialization of parameters and if-then rules is given by human knowledge of controlled plants. Those parameters are automatically adjusted by an on-line learning mechanism. The time-varying learning is determined by the main theorem to guarantee the closed-loop system performance. Illustration examples are utilized to represent the system validation.

C. Treesatayapun is with the Department of Robotic and Advanced Manufactory, CINVESTAV-Saltillo, Ramos Arizpe, Coah., 25900 Mexico e-mail: (treesatayapun@gmail.com).

II. Control Algorithm

A. System configuration

In this work, we consider the system formulation which can be written in the format as

$$x(k+1) = f(\boldsymbol{x}(k)) + g(\boldsymbol{\mu}(k))u(k), \tag{1}$$

when $x(k)$ denotes the output, $u(k)$ is the controller effort and $\boldsymbol{\mu}(k) = [x(k)\, x(k-1) \cdots x(k-n)\, u(k-1)\, u(k-2) \cdots u(k-m)]^T$. These nonlinear functions $f(-)$ and $g(-)$ are assumed to be unknown. For the convergence and stability analysis, the function $g(-)$ the upper limit should be determined or possible for estimation. The control signal $u(k)$ is directly generated my FREN with this formulation

$$u(k) = \boldsymbol{\beta}^T(k)\boldsymbol{\phi}(k), \tag{2}$$

when $\boldsymbol{\beta}(k)$ is adjustable parameter vector defined by $\boldsymbol{\beta}(k) = [\beta_1(k)\, \beta_2(k) \cdots \beta_l(k)]^T$ and $\boldsymbol{\phi}(k)$ is basis function vector for FREN fuzzy mechanism or $\boldsymbol{\phi}(k) = [\phi_1(k)\, \phi_2(k) \cdots \phi_l(k)]^T$ where l denotes as the number of fuzzy rules.

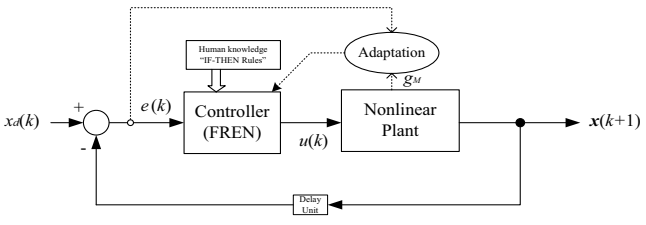

Fig. 1. System configuration.

Fig. 1 illustrates the system configuration. The controller unit is designed to generate the control effort u that forces the

system output x to track the desired trajectory x_d. It is clear that the knowledge about controlled plant is directly integrated to the adaptive controller FREN by if-then rules format. In this work, only on-line leaning mechanism is applied with the associate of an estimated variable call g_M which will be discussed next.

B. Parameters adaptation

To update the parameter β_i for $i = 1, 2, \cdots, l$, the gradient descent technique is implemented with time varying step size or learning rate. Let us define the cost function $E(k)$ at time index k as

$$E(k) = \frac{1}{2}e^2(k), \tag{3}$$

where

$$e(k) = x_d(k) - x(k). \tag{4}$$

The tunable parameter β_i can be obtained in the next time index by

$$\beta_i(k+1) = \beta_i(k) - \eta_i(k)\frac{\partial E(k+1)}{\partial \beta_i(k)}, \tag{5}$$

when $\eta_i(k)$ is a time varying learning rate which will be discussed next for system stability and convergence. Apply the chain rule through (3), (4) and (2), we obtain

$$\begin{aligned}\frac{\partial E(k+1)}{\partial \beta_i(k)} &= \frac{\partial E(k+1)}{\partial x(k+1)}\frac{\partial x(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial \beta_i(k)}, \\ &= -[x_d(k+1) - x(k+1)]y_p(k)\phi_i(k). \end{aligned}\tag{6}$$

Thus, the tuning law can be rewritten as

$$\beta_i(k+1) = \beta_i(k) + \eta_i(k)e(k+1)y_p(k)\phi_i(k), \tag{7}$$

where $y_p(k)$ denotes $\frac{\partial x(k+1)}{\partial u(k)}$. Let us consider the system formulation in (1), clearly, we have

$$y_p(k) = g(\boldsymbol{\mu}(k)). \tag{8}$$

As we mentioned in the previous, in this work, $g(\boldsymbol{\mu}(k))$ is assumed to be unknown but the tuning law needs this information to adjust the parameter. To overcome this problem, we need to design the new adaptation law which will be represented in the next subsection with the convergence analysis.

C. Convergence and stability analysis

The key of this work is to determine the learning rate $\eta_i(k)$ for every tunable parameter β_i for every time index k. Substitute the control effort $u(k)$ given by (2) into the system formulation (1), we have

$$x(k+1) = f(\boldsymbol{x}(k)) + g(\boldsymbol{\mu}(k))\boldsymbol{\beta}^T(k)\boldsymbol{\phi}(k). \tag{9}$$

Thus, the next time step error can be rewritten as

$$e(k+1) = x_d(k+1) - f(k) - g(k)\boldsymbol{\beta}^T(k)\boldsymbol{\phi}(k), \tag{10}$$

where $f(k)$ and $g(k)$ denote for $f(\boldsymbol{x}(k))$ and $g(\boldsymbol{\mu}(k))$, respectively. Substitute (10) into (7), the adaptation equation can be obtained as

$$\begin{aligned}\boldsymbol{\beta}(k+1) &= \beta(k) + \eta(k)[x_d(k+1) - f(k) \\ &\quad -g(k)\boldsymbol{\beta}^T(k)\boldsymbol{\phi}(k)]y_p(k)\boldsymbol{\phi}(k)], \\ &= [I - \eta(k)g^2(k)\boldsymbol{\phi}^T(k)\boldsymbol{\phi}(k)]\boldsymbol{\beta}(k) \\ &\quad +\eta(k)[x_d(k+1) - f(k)]y_p(k)\boldsymbol{\phi}(k). \end{aligned}\tag{11}$$

With this result, let us select learning be

$$\eta(k) = \frac{\gamma}{g_M^2\boldsymbol{\phi}^T(k)\boldsymbol{\phi}(k)}, \tag{12}$$

when g_M is the upper bound of $g(k)$ and $0 < \gamma < 2$ is the designed parameter. Consider (12), thus only the upper bound of $g(k)$ is needed to be known and the convergence of the tunable parameters can be demonstrated with the following lemma.

Lemma: *The convergence of tunable parameters*
With the system given by (1), let the control effort u generated by (2) and the adjustable parameters be tuned by (7). If the learning rate η is given by (12) where $0 < \gamma < 2$ and g_M is the upper bound of $|g(k)|$ then the convergence of adjustable parameters β_i is guaranteed.

Proof Substitute (12) into (7) and associate (1) for the next time index error (10), we have

$$\begin{aligned}\boldsymbol{\beta}(k+1) &= [I - \frac{\gamma}{g_M^2\boldsymbol{\phi}^T(k)\boldsymbol{\phi}(k)}g^2(k)\boldsymbol{\phi}^T(k)\boldsymbol{\phi}(k)]\boldsymbol{\beta}(k), \\ &\quad +\eta(k)[x_d(k+1) - f(k)]y_p(k)\boldsymbol{\phi}(k), \\ &= [I - \gamma\frac{g^2(k)}{g_M^2}]\boldsymbol{\beta}(k) + \eta(k)[x_d(k+1) \\ &\quad -f(k)]y_p(k)\boldsymbol{\phi}(k), \\ &= \xi_1(k)\boldsymbol{\beta}(k) + \xi_2(k), \end{aligned}\tag{13}$$

where $\xi_1(k) = I - \gamma\frac{g^2(k)}{g_M^2}$ and $\xi_2(k) = \eta(k)[x_d(k+1) - f(k)]y_p(k)\boldsymbol{\phi}(k)$. By setting the designed parameter γ as the above, $\xi_1(k)$ is always lest than 1 $(\xi_1(k) < 1), \forall k$.

□

The previous lemma shows about the convergence of the adjustable parameters but the system stability dose not discuss yet. The convergence of closed-loop systems will be introduced by this following theorem.

Theorem: *System stability(Closed loop system convergence)*
Let the desired trajectory $x_d(k)$ be bounded and the upper bound of $g(k)$ be known as g_M. Determine the control effort $u(k)$ by (2) and tune parameters by (7) with the varying learning rate given by (12) when $0 < \gamma < 2$. Then the tracking error $e(k)$ defined by (4) is bounded for the nonlinear system given by (1).

Proof Let define the Lyapunov function candidate as

$$V(k) = \frac{1}{2}e^2(k), \tag{14}$$

thus the change of Lyapunov function can be given as

$$\begin{aligned}\Delta V(k) &= V(k+1)-V(k),\\ &= \frac{1}{2}e^2(k+1)-\frac{1}{2}e^2(k),\\ &= \frac{1}{2}[e(k)-\Delta e(k)]^2-\frac{1}{2}e^2(k),\\ &= \Delta e(k)[e(k)+\frac{\Delta e(k)}{2}]. \end{aligned} \quad (15)$$

Let consider the next time index error can be written by

$$e(k+1)=e(k)+\Delta e(k), \quad (16)$$

where $\Delta e(k)$ can be estimated by

$$\Delta e(k)\approx \Delta^T_{e,\beta}(k)\Delta\beta(k), \quad (17)$$

when $\Delta_{e,\beta}(k)=[\frac{\partial e(k+1)}{\partial\beta_1(k)}\ \frac{\partial e(k+1)}{\partial\beta_2(k)}\ \cdots\ \frac{\partial e(k+1)}{\partial\beta_l(k)}]^T$. To simplify, we obtain

$$\Delta e(k)\approx -g(k)\phi^T(k)\Delta\beta(k). \quad (18)$$

From the tuning law obtained by (11), the change of adjustable parameters can be rewritten as

$$\begin{aligned}\Delta\beta(k) &= \beta(k+1)-\beta(k),\\ &= -\eta(k)g^2(k)||\phi(k)||^2\beta(k)+\eta(k)[x_d(k+1)\\ &\quad -f(k)]g(k)\phi(k). \end{aligned} \quad (19)$$

Substitute (19) into (18), we have

$$\begin{aligned}\Delta e(k) &\doteq \eta(k)g^3(k)||\phi(k)||^2\phi^T(k)\beta(k)-\eta(k)[x_d(k+1)\\ &\quad -f(k)]g^2(k)\phi^T(k)\phi(k),\\ &= \eta(k)g^2(k)||\phi(k)||^2[-x_d(k+1)+f(k)\\ &\quad +g(k)\phi^T(k)\beta(k)],\\ &= -\eta(k)g^2(k)||\phi(k)||^2[e(k+1)]. \end{aligned} \quad (20)$$

By using (16), we can rearrange (20) as

$$\Delta e(k)\doteq \frac{-\eta(k)g^2(k)||\phi(k)||^2e(k)}{1+\eta(k)g^2(k)||\phi(k)||^2}. \quad (21)$$

Substitute (21) into (15), we obtain

$$\begin{aligned}\Delta V(k) &= \frac{-\eta(k)g^2(k)||\phi(k)||^2e^2(k)}{1+\eta(k)g^2(k)||\phi(k)||^2}\\ &\quad \times\Big[1-\frac{\eta(k)g^2(k)||\phi(k)||^2}{2+2\eta(k)g^2(k)||\phi(k)||^2}\Big]. \end{aligned} \quad (22)$$

With the learning rate give by (12), the change of Lyapunov function candidate can be rearranged by

$$\begin{aligned}\Delta V(k) &= \frac{-\gamma\big[\frac{g(k)}{g_M}\big]^2e^2(k)}{1+\gamma\big[\frac{g(k)}{g_M}\big]^2}\Big[1-\frac{\gamma\big[\frac{g(k)}{g_M}\big]^2}{2(1+\gamma\big[\frac{g(k)}{g_M}\big]^2)}\Big]\\ &\le 0. \end{aligned} \quad (23)$$

□

The validation of the proposed controller and our claim about the closed-loop performance will be presented by simulation systems in the next section.

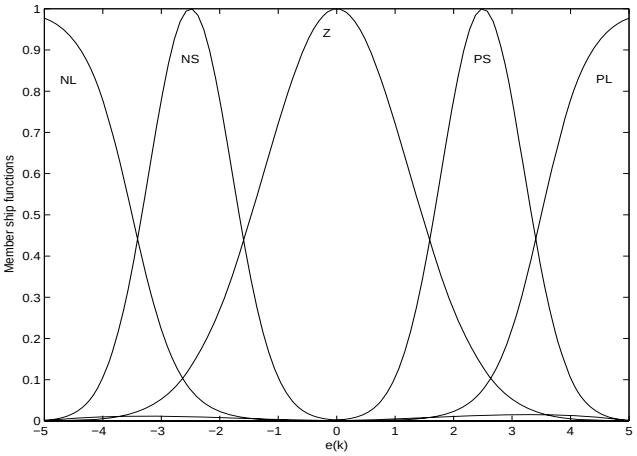

Fig. 2. Membership function for $e(k)$.

III. Illustrated Examples

A. Controller design

The nonlinear discrete-time system which is selected to demonstrate the performance is described as

$$x(k+1)=\sin(x(k))+u(k)[5+\cos(x(k)u(k))], \quad (24)$$

when $x(k)$ denotes the output and $u(k)$ stands for the control effort. The controller receives the error signal defined by (4) and generates the control effort $u(k)$ to track the desired trajectory $x_d(k)$ given by

$$x_d(k)=2\sin(\frac{2\pi k}{50})+2\sin(\frac{2\pi k}{100}). \quad (25)$$

For generality, the range of error should be designed to be ±5 and the membership functions are all given to cover this range as described in Fig. 2.

The next step is to define IF-THEN rules set which is suitable for this system. Consider to the fact that if the error signal is so large in the positive side then the system output should be increased to compensate this error. At the sense of human, the controller should give more the control effort to the plant. With this knowledge, the IF-THEN rule can be constructed as the followings:

If $e(k)$ is PL Then $u_1(k)=\beta_{PL}(k)\phi_1(k)$,
If $e(k)$ is PS Then $u_2(k)=\beta_{PS}(k)\phi_2(k)$,
If $e(k)$ is Z Then $u_3(k)=\beta_Z(k)\phi_3(k)$,
If $e(k)$ is NS Then $u_4(k)=\beta_{NS}(k)\phi_4(k)$,
If $e(k)$ is NL Then $u_5(k)=\beta_{NL}(k)\phi_5(k)$,

when N, Z and P denote "negative", "zero" and "positive" linguistic levels respectively, L stands for "large" and S intends for "small". These adjustable parameters $\beta_\square$ will be discussed next to follow those initial methods.

B. Human based initialization

Based on the knowledge of system, the control effort u has a suitable range that is ±1. At the first IF-THEN rule, the "PL" for β_{PL} should be the larger value at the positive side. On the other hand, those remaining parameters can be given

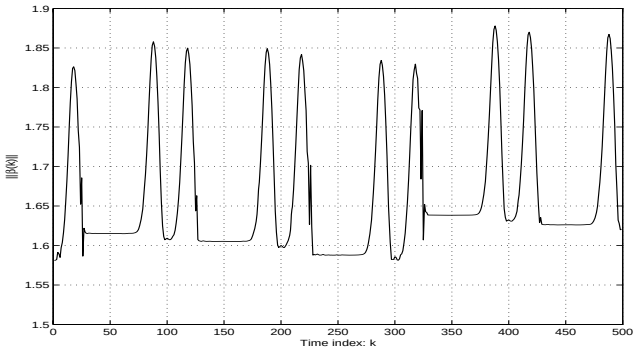

Fig. 3. $||\beta(k)||$:Human Initial setting.

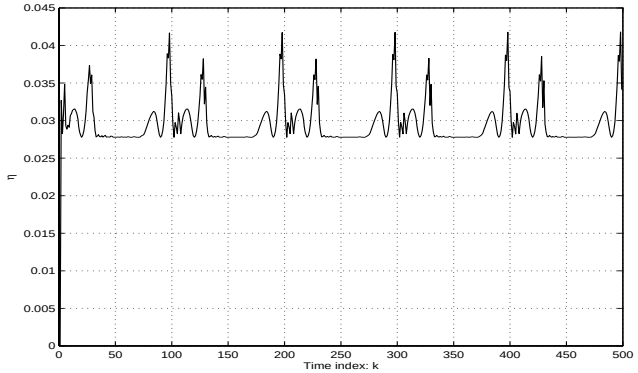

Fig. 4. $\eta(k)$:Human Initial setting.

as the same sense. Thus, for this initial setting, all parameters are defined by this following list:

Parameter	Value
$\beta_{PL}(1)$	1,
$\beta_{PS}(1)$	0.5,
$\beta_Z(1)$	0,
$\beta_{NS}(1)$	-0.5,
$\beta_{NL}(1)$	-1.

Let us define $||\boldsymbol{\beta}(k)|| = [\boldsymbol{\beta}^T(k)\boldsymbol{\beta}(k)]^{\frac{1}{2}}$, the time variation of $||\boldsymbol{\beta}(k)||$ is illustrated in Fig. 3. In Fig. 4, it is clearly attained that the learning rate $\eta(k)$ is varied by following the proposed algorithm. The tracking result is displayed in Fig. 5. Because of the initial setting based on the best knowledge of system, we can obtain the good result at the fist running state.

The effects of initial setting and the performance of the learning algorithm will be discussed in next subsections.

C. Zero initial setting

In this case, all parameters $\beta_{\square}(1)$ are given to be "zero" at the initial setting. With the parameter adaptation, Fig. 6 displays the time variation of $||\boldsymbol{\beta}(k)||$ along the simulation. The learning rate $\eta(k)$ can be shown in Fig. 7. Fig. 8 represents the tracking performance. It is clear that the learning algorithm can provide the better results at the final state with "zero" initial setting.

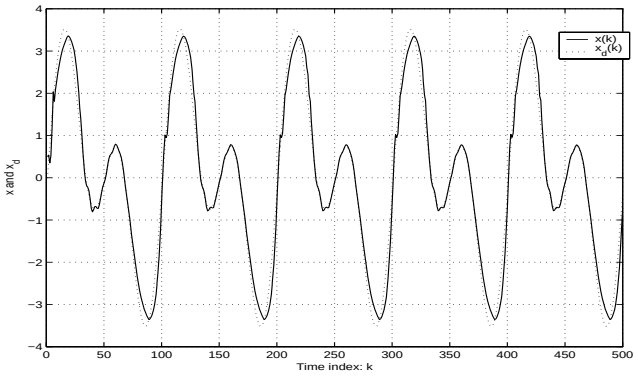

Fig. 5. $x(k)$:Human Initial setting.

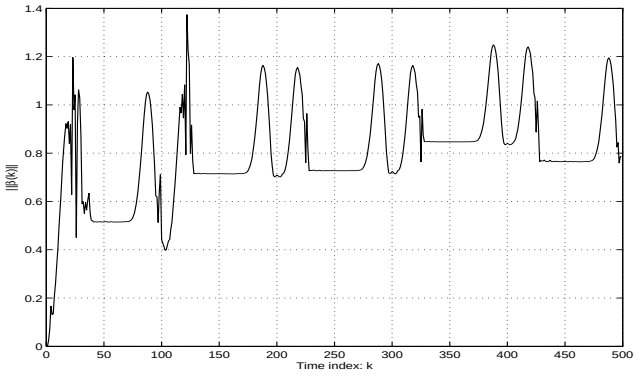

Fig. 6. $||\beta(k)||$:Zero Initial setting.

D. Random initial setting

The initial setting of those parameters is given by random values with in ± 1 and normal distribution. The colored tracking result is occurred at the beginning as depicted in Fig. 11. At the final state, we can obtain the satisfied result because of the on-line learning process. Fig. 9 and 10 show the time varying of $||\boldsymbol{\beta}(k)||$ and $\eta(k)$, respectively.

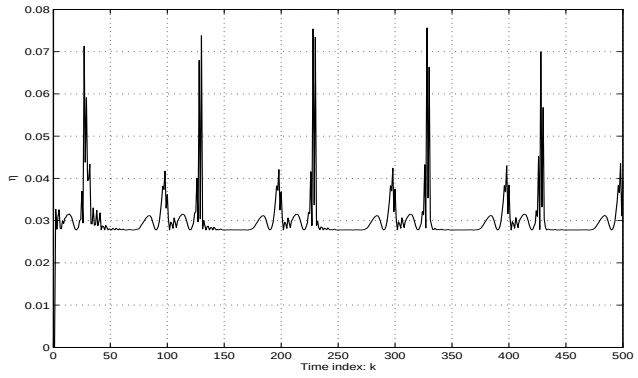

Fig. 7. $\eta(k)$:Zero Initial setting.

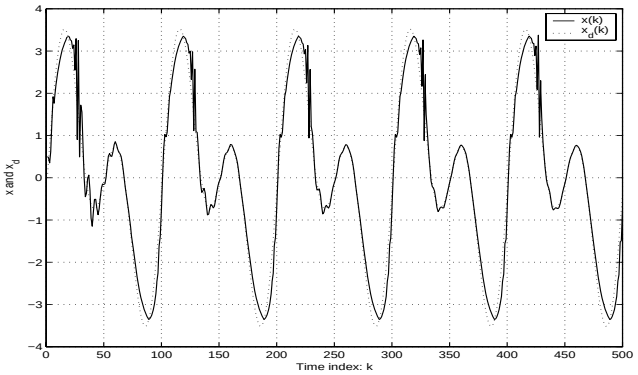

Fig. 8. $x(k)$:Zero Initial setting.

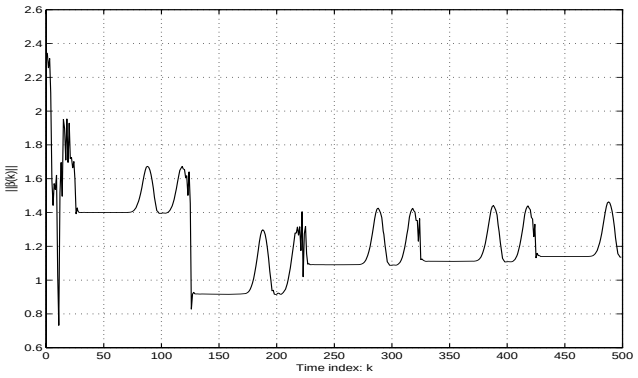

Fig. 9. $||\beta(k)||$:Random Initial setting.

IV. Conclusion

In this article, we have developed a direct adaptive controller for discrete-time nonlinear systems. These adjustable parameters inside an adaptive network called FREN are automatically tuned by an on-line algorithm. The step size of learning rate is determined on-line to guarantee the system stability. This proposed algorithm can be implemented even the plant mathematic model is unknown. With the simulation example, only the estimated g_M is needed to operate the controller. An advantage point of FREN is the integrated human knowledge through IF-THEN rules. The initial setting for membership functions and adjustable parameters can be designed by that knowledge. The simulation results confirm our claim with human sense setting as the better tracking performance than "zero" and "random" settings.

Eventually, if the good initial setting can not be contributed by the designer then the satisfied tracking results can be obtained by the on-line learning algorithm at the last state. Not only the tracking performance is guaranteed but also the convergence of adjustable parameters is verified by lemma and time variations of $||\beta(k)||$.

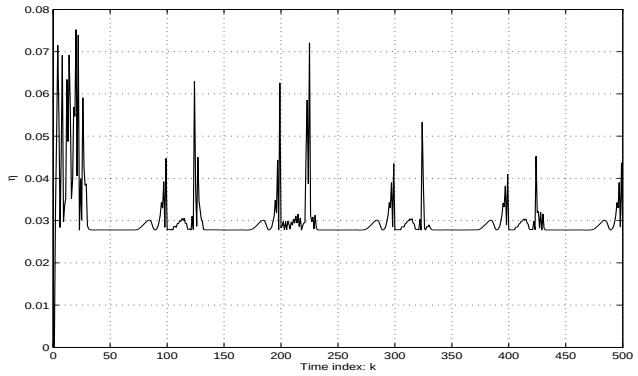

Fig. 10. $\eta(k)$:Random Initial setting.

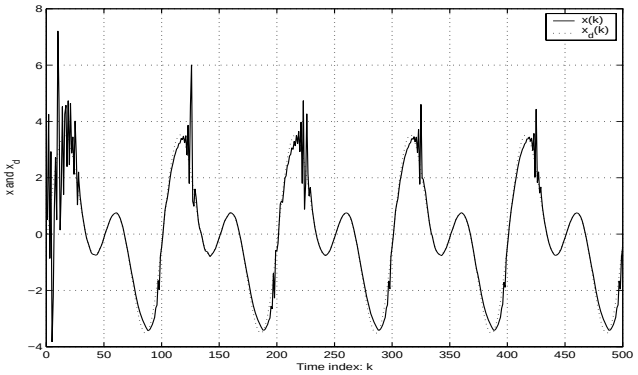

Fig. 11. $x(k)$:Random Initial setting.

Acknowledgment

The authors would like to thank CONACYT (Project # 84791) for the financial support through this work.

References

[1] J. Zhao and I. Kanellakopoulos, "Active Identification for discrete-time nonlinear control-part I: Output-Feedback systems," *IEEE Trans. Automat. Contr.*, vol. 47, no. 2, pp. 210-224, Feb. 2002.
[2] J. Zhao and I. Kanellakopoulos, "Active Identification for discrete-time nonlinear control-part II: Strict-Feedback systems," *IEEE Trans. Automat. Contr.*, vol. 47, no. 2, pp. 225-240, Feb. 2002.
[3] H. Deng, H.X. Li and Y.H. Wu, "Feedback-Linearization-Based neural adaptive control for unknown nonaffine nonlinear discrete-time systems," *IEEE Trans. Neural Networks*, vol. 19, no. 9, pp. 1615-1625, Sep. 2008.
[4] B.T. Thumati and S. Jagannathan, "Neural network control of a class of nonlinear discrete time systems with asymptotic stability guarantees," *American control conference 2009*, St. Louis, MO, USA, pp. 2934-2939, 10-12 Jun. 2009.
[5] H. E. Psillakis, "Sampled-Data adaptive NN tracking control of uncertain nonlinear systems," *IEEE Trans. Neural Networks*, vol. 20, no. 2, pp. 336-355, Feb. 2009.
[6] C.T. Lin, *Neural fuzzy systems*, Prentice-Hall, 1996.
[7] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-Time Systems*, Boca Raton, FL: Taylor & Francis, 2006.