# A generalized procedure in designing recurrent neural network identification and control of time-varying-delayed nonlinear dynamic systems

Xueli Wu [1* 2] Jianhua Zhang [1] Quanmin Zhu[3]

(1. YanShan University, Qinhuangdao, 066004,China)

(2. Hebei University of Science and Technology, Shijiazhuang, 050054,China)

(3. Bristol Institute of Technology, University of the West of England, Coldharbour Lane, Bristol BS161QY, UK)

* The corresponding author, wuxueli@yahoo.com.cn

**Abstract**

In this study, a generalized procedure in identification and control of a class of time-varying-delayed nonlinear dynamic systems is developed. Under the framework, recurrent neural network is developed to accommodate the on-line identification, which the weights of the neural network are iteratively and adaptively updated through the model errors. Then indirect adaptive controller is designed based on the dichotomy principles and neural networks, which the controller output is designed as a neuron rather than an explicit input term against system states. It should be noticed that including implicit control variable in design is more challenging, but more generic in theory and practical in applications. To guarantee the correctness, rigorousness, generality of the developed results, Lyapunov stability theory is referred to prove the neural network model identification and the designed closed-loop control systems uniformly ultimately bounded stable. A number of bench mark tests are simulated to demonstrate the effectiveness and efficiency of the procedure and furthermore these could be the show cases for potential users to apply to their demanded tasks.

**Keywords**

Time-varying-delayed nonlinear systems; on line identification; adaptive control; implicit controller design.

## 1. Introduction

Neurocomputing techniques and algorithms have been extensively developed and applied to many different fields. This study falls in a domain with the problem of identifying and controlling a class of complex nonlinear dynamic systems and the solution of being developed into an integrated procedure embedded with neurocomputing, iterative learning, online identification, and adaptive control. The rest of the introduction section includes the background of the study, the state of the art of the related research, the motivation and justification of the study, which will deliver a clear picture to the problems and solutions for the potential readers and users before going into detailed implementation of the solutions.

In the past decade, control system design for uncertain nonlinear systems (Wang et al., 2008, Noroozi et al. 2009) has received much more attention, particularly in the research of using universal function approximators to parameterize unknown nonlinearities. Neural network based inductive techniques, subject to their inherent approximation capabilities, have been found to be very supportive for controlling such class of complex dynamic systems(Liu, 2008, Zhang, 2009).Unavoidably online model identification is a kernel part of such control system design, neural networks have been very effective to approximate a wide class of complex nonlinear systems in case of no full model information achievable, or even in the case of those plants with black-box structures, which the sole info is the measured input and output data sequences.

In structure, neural networks can be classified as feedforward and recurrent ones. Feedforward networks are suitable for the approximation of complex static functions. The major drawback of such type of neural networks in describing dynamic functions is that the weights' updating does not utilize information on the local date structure and the function approximation is sensitive to the purity of training data. On the other hand, the recurrent networks incorporate feedback, not merely having concise structure, but importantly adaptive mechanism incorporated to fine-tune the approximation accuracy and convergent speed.

One of the major neural network based adaptive control approaches is based on the Lyapunov's stability theory which gives an adaptive control law with guaranteed stability of the closed-loop systems (Rubio et al. 2007a,2007b, Yu,2004,2006,2007,). In particular, Polycarpou (1996) developed an adaptive neural network control scheme for uncertain strict-feedback nonlinear systems using the backstepping technique. This approach relaxed the matching condition on the plant uncertainties. The design procedure made use of linearly parameterized neural networks such as radial basis function networks with fixed centers and widths. Adaptive backstepping design was proposed for parametric strict-feedback systems with overparameterization, which could guarantee globally stable and asymptotic tracking performance. By introducing tuning functions, the overparameterization can be effectively eliminated. The adaptive backstepping design has been further extended to parametric strict-feedback systems with unknown virtual control coefficients.

A notable contribution by Ge et al. (2003, 2005) is the construction of an integral Lyapunov function which is proved to be the key for the success of the approach. A function approximator has been used to describe the unknown nonlinear functions. Two types of artificial neural networks, the linearly parameterized neural networks (Ge et al, 2002, 2003, 2005) and the multilayer neural networks (Lewis, 1996), Wang & Huang, 2002, and Zhang, et al, 1999), have been predominantly used for approximating a wide range of unknown nonlinear functions in control system design.

Generally, all these studies referred above have presented a unified and general framework for nonlinear control system design in terms of either pure-feedback or strict-feedback, which a neural network is used to approximate unknown nonlinear functions and/or uncertain terms in the systems, and then based on the identified neural networks design controllers. However, from author's recent studies, there are still a number of issues should have been further studied and the corresponding solutions should have been developed. Here are the some major concerned issues.

1) Most of the published works have considered the system described as

$$\dot{x}(t) = f(x(t)) + g(x)u(t)$$

There have been some assumptions on function $g(x)$, such as $0 < g_1 \le g(x) \le g_2$, this is usually a form of known control directions(Zhang & Ge, 2007). The other assumption has been with $g(x) = g$, a constant and unknown, which is known as unknown control directions (Yang et al. 2009). Further study should include more practical and general form of nonlinear systems in the expression of

$$\dot{x}(t) = f(x(t), u(t))$$

The control input $u(t)$ should not be restricted as an explicit function of the states with known or unknown direction gains. Especially in many real systems, control $u(t)$ is an implicit function of system states, such as $\dot{x}(t) = f(x(t)) + xu(t) + \sin(u(t))$. Obviously identification and control of such class systems are more challenging, but more valuable in theory and practice. Simply the new system structure includes the previous one in one of its subsets.

2) With regarding to neural network enhanced adaptive control, a lot of papers have proposed the learning law for neural networks with ideal weight of the neural networks, such work can be found in Yu's publication (2006). Although the ideal weights of the neural networks are bound and exist, it is unknown. Accordingly estimated weights should be put in place in their updated process.

3) A milestone work by Rubio and Yu (2007b) has laid a foundation for using neural networks to approximate time delayed nonlinear dynamic systems. The first expansion could be to remove the pre-request of knowing some of the neural networks weight matrices to make neural networks more realistic in real applications and the second expansion is the integration of the identification with adaptive control to establish a comprehensive framework for online control of complex nonlinear dynamic systems, which has been widely demanded in many industrial domains. This has been a popular insight and approach in using neural networks to approximate complex nonlinear systems and then using the reference model plus linear design

methodologies to design nonlinear control systems. As it has not been completed yet, a lot of research and application tests actually are still remained in questions.

Motivated with the above issues, this study is devoted to a class more general time-varying-delayed nonlinear dynamic systems with implicit inputs. A new dynamic neural network fully connected with the neurons, is proposed to accommodate the utilization of Lyapunov stability theorem and adaptive laws in design of online identification and control. In technique details, the weights of the neural network are updated by the identification errors and the states of the neural network in terms of adaptation. The controller output is resolved in principle of dichotomy. The convergence of the neural network weight estimation and stability of the control system are analyzed by Lyapunov-Krasovskii approach to prove that the resulting closed-loop system is uniformly ultimately bounded stable. This study will provide a platform to make nonlinear control system design as straightforward as linear control system design.

The rest of the study is organized as follows. In Section 2, the problem formulation and preliminaries are presented, which a general nonlinear dynamic system model and its neural network approximator are presented to establish a basis for designing and analyzing the system identification and control. In Section 3, an identification algorithm is developed to approximate the nonlinear systems and the corresponding convergence is proved. In Section 4, an indirect adaptive controller is design first and then the closed-loop system stability is analyzed. Based on the neural network model and dichotomy principle, the controller output can be effectively determined. In Section 5, an online algorithm for the identification and control is presented to provide a step by step guide for potential users. In Section 6, three simulated case studies are conducted to initially demonstrate the efficiency and effectiveness of the procedure.

## 2. Preliminaries

A continuous-time-varying-delay nonlinear system can be generally described as:

$$\dot{x}(t) = h\left(x(t), x(t-\tau(t)), u(t)\right) \tag{1}$$

where $x(t) = \left[x_1(t), x_2(t), \ldots, x_n(t)\right]^T \in \Re^n$ is the state vector, $x(t-\tau(t))$ is state vector with bounded time-varying delays, $u(t) = \left[u_1(t), u_2(t), \ldots, u_n(t)\right]^T \in \Re^n$ is the bound input vector, and $h$ is an unknown continuous nonlinear smooth function. In this study it is assumed that each subsystem has one to one structure that is only one input to act on each subsystem. Mathematically this can be expressed as

$$\dot{x}_i(t) = h_i\left(x(t), x(t-\tau(t)), u_i(t)\right)$$
$$i = 1, 2, \ldots n$$

It should be mentioned that there is no need to keep the dimensions equalized (dim(u)=dim(x)), for example, if dim(u)<dim(x) (Yu, 2007), the new method allows some of $u_i = 0$. Secondly the general model can be used to describe nonlinear relationships between the inputs and states, instead of just linear input actions to the system states.

In this paper, a delayed recurrent neural network $h_{nn}(x)$ will be used to approximate the continuous function $h(x)$, so that $h(x) = h_{nn}(x) + \tilde{h}(x)$, where $\tilde{h}(t)$ is the modelling error. The neural network is formatted as

$$h_{nn} = C^* x(t) + A^* f\left(x(t)\right) + B^* f\left(x(t-\tau(t))\right) + S^*$$

where $C^*, A^*,$ and $B^*$ are the weights and $S^*$ is the threshold of the neural network, $C^* \in \Re^{n\times n}, A^* \in \Re^{n\times n}, B^* \in \Re^{n\times n}, S^* \in \Re^n$. The whole structure of the neural network is functional and concise to facilitate the utilization of Lyapunov stability theorem and adaptive laws in the following designs.

The nonlinear system (1) can be written in the form of following neural networks:

$$\dot{x}(t) = C\overline{x}(t) + Af(\overline{x}(t)) + Bf(x(t-\tau(t))) + S - \tilde{h}(t) \tag{2}$$

where $\forall t \in (0,\infty)$, $C = \begin{bmatrix} C_1^T & C_2^T \end{bmatrix}^T$, $\overline{x}(t) = \begin{bmatrix} x^T(t) & u^T(t) \end{bmatrix}^T$, $A = \begin{bmatrix} A_1^T & A_2^T \end{bmatrix}^T$, and $C_1 \in \Re^{n\times n}$, $C_2 = diag\left(c_{1,n+1}, c_{2,n+2}, \ldots, c_{n,2n}\right) \in \Re^n$, $A_1 \in \Re^{n\times n}$, $A_2 = diag\left(a_{1,n+1}, a_{2,n+2}, \ldots, a_{n,2n}\right) \in \Re^n$, $B \in \Re^{n\times n}, S \in \Re^n$, vector $x(t)$ is the state of the neural network, $x(t-\tau(t))$ is the delayed state of the neural network, with $x(\theta) = \phi(\theta)\ \forall \theta \in [-\tau, 0]$, $u(t)$ is the control vector, and the $\|u(t)\| \le \bar{u}$, $\tau(t)$ is the time-varying delay, the ideal weights $C, A, B$ are the bounded matrices, $f$ is the active function, $\tilde{h}(t)$ is the modelling error. Since the state and output variables are physically bounded, the modelling error $\tilde{h}(t)$ can be assumed to be bounded, and $\left|\tilde{h}(t)\right| < \varepsilon$.

For the known time-delay $\tau(t)$, it is bounded with

$$0 \le \tau(t) \le \tau, \left|\dot{\tau}(t)\right| \le \xi < 1$$

The active function is specified as a monotonically increased function, and bounded with

$$0 \le f(x) - f(y) \le k(x-y)$$

for any $x, y, k \in R$ and $x \le y, k > 0$, such as

$$f(x) = \tanh(x)$$

Selection of this kind of active function would guarantee the controller exist and obtainable by dichotomy principle.

The ideal weights $C, A, B, S$ are defined as follows:

$$(C, A, B, S) = \arg \min_{C,A,B,S} \left( \sup \left| h_{nn}(x) - h(x) \right| \right).$$

### 3. System identification with neural network

The nonlinear system (1) can be approximated by the following continuous-time delayed neural network:

$$\begin{aligned}\dot{y}(t) = {} & C(t)\bar{y}(t) + A(t)f(\bar{y}(t)) + B(t)f(y(t-\tau(t))) + S(t) \\ & + L(t)e(t) - diag(\operatorname{sgn}(e(t))) * \varepsilon(t)\end{aligned} \tag{3}$$

where $\bar{y}(t) = \begin{bmatrix} y^T(t) & u^T(t) \end{bmatrix}^T$ the neural network output vector, that is the estimate of the system state vector, $C(t), A(t), B(t)$ are interconnection weight matrices of delayed neural networks, and $S(t)$ is the threshold of neural networks. Identification error is defined as $e(t) = y(t) - x(t)$, and $L(t)$ is a diagonal constant matrix. The error derivative is obtained from (3), which is given below

$$
\begin{aligned}
\dot{e}(t) &= \dot{y}(t) - \dot{x}(t) \\
&= \tilde{C}\bar{y}(t) + C_1 e(t) + \tilde{A} f(\bar{y}(t)) + A_1 \tilde{f}(e(t)) + \tilde{B} f(y(t-\tau(t))) \\
&\quad + B\tilde{f}(e(t-\tau(t))) + (\tilde{S} + L(t))e(t) - diag(\operatorname{sgn}(e(t))) * \varepsilon(t) + \tilde{h}(t)
\end{aligned}
\tag{4}
$$

where $e(t) = 0$ for $t \in [-\tau, 0]$,

$$
\begin{gathered}
\tilde{C} = C(t) - C, \tilde{A} = A(t) - A, \tilde{B} = B(t) - B, \tilde{S} = S(t) - S, \tilde{L} = L(t) - L, \tilde{\varepsilon} = \varepsilon(t) - \varepsilon \\
\tilde{f}(e(t)) = f(y(t)) - f(x(t))
\end{gathered}
\tag{5}
$$

**Theorem** The identification error derivative (4) is uniformly ultimately bounded, where the weights of the delayed neural network are updated through following equations

$$
\begin{aligned}
\dot{C}(t) &= -\Gamma_1 (e(t)\bar{y}^T(t) + \sigma_1 C(t)) \\
\dot{A}(t) &= -\Gamma_2 (e(t) f^T(\bar{y}(t)) + \sigma_2 A(t)) \\
\dot{B}(t) &= -\Gamma_3 (e(t) f^T(y(t-\tau(t))) + \sigma_3 B(t)) \\
\dot{S}(t) &= -\Gamma_4 (e(t) + \sigma_4 S(t)) \\
\dot{L}(t) &= -\Gamma_5 (e(t)e^T(t) + \sigma_5 L(t)) \\
\dot{\varepsilon}(t) &= -\Gamma_6 (-|e(t)| + \sigma_6 \varepsilon(t))
\end{aligned}
\tag{6}
$$

where $\Gamma_i$ is a positive definite matrix, and $\sigma_i$ is a constant, such that $\Gamma_i = \Gamma_i^T > 0$, $\sigma_i > 0$, $i = 1, 2 \ldots 6$.

**Proof** Select a Lyapunov-Krasovskii candidate function as

$$
V_1 = \frac{1}{2} e^T(t) e(t)
\tag{7}
$$

whose derivative is

$$
\begin{aligned}
\dot{V}_1 &= e^T(t)\tilde{C}\bar{y}(t) + e^T(t)C_1 e(t) + e^T(t)\tilde{A} f(\bar{y}(t)) + e^T(t) A_1 \tilde{f}(e(t)) \\
&\quad + e^T(t)\tilde{B} f(y(t-\tau(t))) + e^T(t) B\tilde{f}(e(t-\tau(t))) \\
&\quad + e^T(t)\tilde{S} + e^T(t)L(t)e(t) - |e^T(t)|\varepsilon(t) + e^T(t)\tilde{h}(t)
\end{aligned}
\tag{8}
$$

With taking the active function into consideration, it gives

$$
\begin{aligned}
e^T(t) A_1 \tilde{f}(e(t)) &\le \frac{1}{2} e^T(t) A_1 A_1^T e(t) + \frac{k^2}{2} e^T(t) e(t) \\
e^T(t) B\tilde{f}(e(t-\tau(t))) &\le \frac{1}{2} e^T(t) BB^T e(t) + \frac{k^2}{2} e^T(t-\tau(t)) e(t-\tau(t))
\end{aligned}
\tag{9}
$$

Accordingly the following inequality is held

$$\dot{V}_1 \le -\eta_1 V_1 + e^T(t)\tilde{C}\bar{y}(t) + e^T(t)\tilde{A}f(\bar{y}(t)) + e^T(t)\tilde{B}f(y(t-\tau(t)))$$
$$+e^T(t)\tilde{S} + e^T(t)\tilde{L}e(t) - \left|e^T(t)\right|\tilde{\varepsilon}(t) + \frac{k^2}{2}e^T(t-\tau(t))e(t-\tau(t)) \tag{10}$$
$$+e^T(t)\left[C_1 + \frac{1}{2}A_1A_1^T + \frac{1}{2}BB^T + L + \frac{k^2+2}{2}\right]e(t)$$

where the $\eta_1 = 1$. Because $\tilde{C}, \tilde{A}, \tilde{B}, \tilde{S}$ contain the ideal weight matrices, so cancelled them in the next step.

$$V_2 = \frac{1}{2}tr\left(\tilde{C}^T\Gamma_1^{-1}\tilde{C} + \tilde{A}^T\Gamma_2^{-1}\tilde{A} + \tilde{B}^T\Gamma_3^{-1}\tilde{B} + \tilde{S}^T\Gamma_4^{-1}\tilde{S} + \tilde{L}^T\Gamma_5^{-1}\tilde{L} + \tilde{\varepsilon}^T\Gamma_6^{-1}\tilde{\varepsilon}\right) \tag{11}$$

Accordingly its derivative is derived as below

$$\dot{V}_2 \le -e^T(t)\tilde{C}\bar{y}(t) - e^T(t)\tilde{A}f(\bar{y}(t)) - e^T(t)\tilde{B}g(y(t-\tau(t))) - e^T(t)\tilde{S}$$
$$-e^T(t)\tilde{L}e(t) + \left|e(t)\right|\tilde{\varepsilon}(t) - \eta_2 V_2 + \frac{\sigma_1}{2}tr\left(C_1^TC_1\right) + \frac{\sigma_2}{2}tr\left(A_1^TA_1\right) \tag{12}$$
$$+\frac{\sigma_3}{2}tr\left(B^TB\right) + \frac{\sigma_4}{2}tr\left(S^TS\right) + \frac{\sigma_5}{2}tr\left(L^TL\right) + \frac{\sigma_6}{2}tr\left(\varepsilon^T\varepsilon\right)$$

where the $\eta_2 = \max\left(\frac{\sigma_i}{\lambda_{max}(\Gamma_i^{-1})}\right), i = 1, 2, \ldots, 6$.

Assign $V_3$ to cancel the sections associated with time delay in $\dot{V}_1$.

$$V_3 = \frac{k^2}{2(1-\xi)}\int_{t-\tau(t)}^{t}\exp\left(\frac{t-\tau(t)-\theta}{\xi-1}\right)e^T(\theta)e(\theta)d\theta \tag{13}$$

And its derivative is

$$\dot{V}_3 \le -\eta_3 V_3 + \frac{k^2}{2(1-\xi)}\exp\left(\frac{\tau}{1-\xi}\right)e^T(t)e(t) - \frac{k^2}{2}e^T(t-\tau(t))e(t-\tau(t)) \tag{14}$$

where $\eta_3 = \frac{1+\xi}{1-\xi}$. Above all choose the total Lyapunov function as

$$V = V_1 + V_2 + V_3$$

And its derivative is

$$\dot{V} \le -\eta V + e^T(t)\Xi e(t) + \delta$$

where

$$\Xi = C_1 + \frac{1}{2}A_1A_1^T + \frac{1}{2}BB^T + L + \frac{k^2+1}{2} + \frac{k^2}{2(1-\xi)}\exp\left(\frac{\tau}{1-\xi}\right) \tag{15}$$

$$\delta = \frac{\sigma_1}{2} tr\left(C_1^T C_1\right) + \frac{\sigma_2}{2} tr\left(A_1^T A_1\right) + \frac{\sigma_3}{2} tr\left(B^T B\right) + \frac{\sigma_4}{2} tr\left(S^T S\right) + \frac{\sigma_5}{2} tr\left(L^T L\right) + \frac{\sigma_6}{2} tr\left(\varepsilon^T \varepsilon\right)$$

$$\eta = \min\left(\eta_1, \eta_2, \eta_3\right)$$

To make $\Xi < 0$ to specify $L$ as follows

$$L < -\left(C_1 + \frac{1}{2} A_1 A_1^T + \frac{1}{2} BB^T + \frac{k^2+2}{2} + \frac{k^2}{2\left(1-\xi\right)} \exp\left(\frac{\tau}{1-\xi}\right)\right) \tag{18}$$

$\delta$ is bound from the characteristics provided from the ideal matrices and the modeling error.

Therefore it can be concluded

$$\dot{V} \le -\eta V + \delta \tag{19}$$

$$0 \le V \le \left(V\left(0\right) - \frac{\delta}{\eta}\right) \exp\left(-\eta t\right) + \frac{\delta}{\eta} \tag{20}$$

$$\left\| e\left(t\right) \right\| \le \sqrt{2\left(V\left(0\right) - \frac{\delta}{\eta}\right) \exp\left(-\eta t\right) + \frac{2\delta}{\eta}}$$

$$\lim_{t \to \infty} \left\| e\left(t\right) \right\| = \sqrt{\frac{2\delta}{\eta}}$$

With reference to boundness theorem (Ge et al. 2003), the error induced in using the neural network to approximate this class of systems is uniformly ultimately bounded. The boundary $\sqrt{\frac{2\delta}{\eta}}$ is a function of the ideal weights of the neural network and $\sigma$ in (6).

□

**Remark 1** In theory the gains Gamma and Sigma could be chosen arbitrarily within their closure sets. But limited by computation algorithms, a proper selection of Gamma and Sigma is necessary for identification and control. Similar work can be found in some papers, such as Ge, Hong, and Lee (2003).

## 4. Indirect adaptive controller design

The aim of the controller design is to drive the outputs (states) of the system properly following a pre-specified trajectory. Without losing generality, let the desired trajectory $x^*\left(t\right)$ be smooth. To describe the states of the system following the desired trajectory a nonlinear system in category of (1) is presented below

$$\dot{x}\left(t\right) = h_{nn} - \tilde{h}\left(t\right) \tag{21}$$

The parameters of neural network $h_{nn}$ could be gained through the identification algorithms.

The desired trajectory is notated as $x^*\left(t\right)$, through coordinate transformation

$$z\left(t\right) = x\left(t\right) - x^*\left(t\right) \tag{22}$$

Accordingly nonlinear system (21) can be alternatively expressed as

$$\dot{z}(t) = h_{nn} - \tilde{h}(t) - \dot{x}^*(t) \tag{23}$$

Because $x^*(t)$ is the pre-specified desired trajectory either by customers or designers, the objective is to drive the output of the plant $x(t)$ following the desired trajectory $x^*(t)$.

$$h_{nn} = -k^* z(t) - \varepsilon^T(t)\,\mathrm{sgn}(z(t)) + \dot{x}^*(t),\ k^* > 0 \tag{24}$$

$$\dot{z}(t) = -k^* z(t) - \varepsilon^T(t)\,\mathrm{sgn}(z(t)) - \tilde{h}(t) \tag{25}$$

$$V = \frac{1}{2} z^T(t) z(t) \tag{26}$$

$$\begin{aligned}\dot{V} &= -k^* z^T(t) z(t) - \left|z^T(t)\right|\varepsilon(t) + z^T(t)\tilde{h}(t) \\ &\le -k^* z^T(t) z(t)\end{aligned} \tag{27}$$

Till this stage, the controller design task can be clearly stated as: Use identified the neural network (model) to represent the system behaviour, with this reference model to design the control system, therefore the controlled system output (states) can be properly driven following a pre-specified desired trajectory.

If the dimension of the controller $\dim(u(t)) = n$, based on the identification method

$$\begin{aligned}h_{nn} &= -k^* z(t) - \varepsilon(t)\,\mathrm{sgn}(z(t)) + \dot{x}^*(t) \\ &= C_1(t)x(t) + C_2(t)u(t) + A_1(t)f(x(t)) \\ &\quad + A_2 f(u(t)) + B(t)f(x(t-\tau(t))) + S(t)\end{aligned}$$

It is requested to resolve the root of the nonlinear equations to obtain the controller output,

$$\xi(u^*) = D_1 + D_2 u^*(t) + D_3 f(u^*(t)) = 0 \tag{28}$$

where the parameters of the equation are specified by

$$\begin{aligned}D_1 &= k^* z(t) + \varepsilon(t)\,\mathrm{sgn}(z(t)) - \dot{x}^*(t) + C_1(t)x(t) \\ &\quad + A_1(t)f(x(t)) + B(t)f(x(t-\tau(t))) + S(t) \\ D_2 &= C_2(t) \\ D_3 &= A_2(t)\end{aligned}$$

Because the active function is bounded, it can be proved that the ideal controller is exist and can be resolved by dichotomy principle (Burden and Faires, 2004). To further explain the assertion, take off one of the nonlinear equations, say

$$\xi(u) = d_1 + d_2 u + d_3 f(u) \tag{29}$$

1) If $d_1 = 0$, then $u = 0$ is one of the roots for nonlinear equation (29);

2) If $d_1 \neq 0$ without losing generality, assume $d_1 > 0$, it would not effect the root of the nonlinear equation (24), then assume $d_2 \neq 0$, set up $a = 0, b = -\frac{\alpha(d_1 + abs(d_3))}{d_2}$, where $\alpha > 1$, then $\xi(a) > 0$ and $\xi(b) < 0$. Base on the

theorem of the numerical analysis, there exists an ideal controller output $u^*$ between $a$ and $b$. Then $\xi(u)=0$ can be resolved through the dichotomy principle which is presented in appendix.

3) If the $d_1 \neq 0, d_2 = 0$, then the $u = f^{-1}\left(-\frac{d_1}{d_3}\right)$ when the $\left|\frac{d_1}{d_3}\right| \leq 1$, else the root of the nonlinear equation (24) don't exist.

The whole online computational algorithm for the system identification and controller design will be described in next section.

In summary the indirect adaptive control addressed in this study is to guarantee all the signals in the closed-loop system remain bounded, and error system state $z$ converge to origin. From Lyapunov stability theorem, system (23) is stable, because the error of the system is ultimately uniformly bounded, the system state $x$ would converge to a small neighbourhood around desired trajectory $x^*$.

## 5. Algorithm for implement

In this section, a step by step procedure is listed to implement the control scheme.

Step1. Assign initial values of the gain matrices in adaptive law $\Gamma, \sigma, L$, the initial value of the to be estimated parameters and variable states of $C(1), A(1), B(1), S(1), L(1), \varepsilon(1), y(1)$.

Step2. Based on the initials of the state $x(1)$ and system input $u(1)$, calculate the state of the neural networks $y(2)$, start the algorithm at time instant i=1.

Step3. Calculate the new parameters and variables $C(i+1), A(i+1), B(i+1), S(i+1)$ by adaptive law (6), and determine the one step ahead variable $y(i+1)$ according to (3).

Step4. Assign the threshold $\varepsilon_0 = 0.01$ and system input $u(i+1)=0$, if $|d_1| \leq \varepsilon_0$, go to Step 9.

Step5. If $|d_2| \leq \varepsilon_0$, $|d_3| > \varepsilon_0$ and $\left|\frac{d_1}{d_3}\right| \leq 1$, then $u(i+1) = f^{-1}\left(-\frac{d_1}{d_3}\right)$, go to Step 9.

Step6. If $|d_2| > \varepsilon_0$, then $d_2 = d_2 * \operatorname{sgn}(d_1)$, $d_3 = d_3 * \operatorname{sgn}(d_1)$, $d_1 = abs(d_1)$, assign the initial values $a = 0$, $b = -\frac{\alpha(d_1 + abs(d_3))}{d_2}$, $\alpha = 1.1$, for dichotomy principle. Set up iterative learning step index j=1.

Step7. Assign $c = 0.5(a+b)$, then calculate $\xi(c)$ from (23). If $|\xi(c)| < \varepsilon_0$, then $u(i+1) = c$, go to Step 9.

Step8. If $\xi(c) > 0$, then $a = c$, else $b = c$, j=j+1, if j<3000 go to Step 7.

Step9. i=i+1 go to Step 3.

This is the online identification and control algorithm for nonlinear system with time delay. It should be mentioned that the algorithm can be solely used for identification in either open loop or closed loop.

## 6. Simulation studies

Two examples were selected to validate the performance of the proposed procedure and also to illustrate how to use the procedure in design.

**Example 1** The first system was given by

$$\dot{x}_1 = x_1(t) - x_1(t) * x_2(t) - x_1(t-\tau(t)) * x_2(t-\tau(t)) + u_1$$
$$\dot{x}_2 = x_1(t) + x_2(t) + x_2(t) * u_2 + u_2$$

The system was initially located at $x(0) = (1,1)^T$. Assumed the structure of the two inputs and two outputs system unknown that is a black box system except external input-output data available. In this experiment, the first test was the feasibility to use the proposed neural network to approximate the dynamic system, which was treated as a model identification case study. Initially set input $u_1(t) = \sin(t), u_2(t) = \cos(t)$ to identify the free response model. The cellular neural network was structured with 2 layers, 10 neurons for input layer, 2 neurons for output layer, and the activation functions were selected as hyperbolic tangent, $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

To design the controller, it was requested to find a model to approximate the system, therefore selected a recurrent neural network (model) as follows

$$\dot{y}(t) = C(t)\bar{y}(t) + A(t)f(\bar{y}(t)) + S(t) + L(t)e(t) - diag(\operatorname{sgn}(e(t))) * \varepsilon(t)$$

which was composed with 2 layers, 8 neurons for input layer, 2 neurons for output layer, activation functions were chosen as $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. Assigned the initials $C(0) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}, A(0), S(0), L(0), \varepsilon(0), y(0)$ as null matrices and null vectors and $\Gamma = \begin{bmatrix} 1 & 0.01 \\ 0.01 & 1 \end{bmatrix}, \sigma = 0.001$.

Then run the on-line identification and controller design procedure listed in the last section. The input and output were sampled every 0.01s and the whole process was run for 50 seconds.

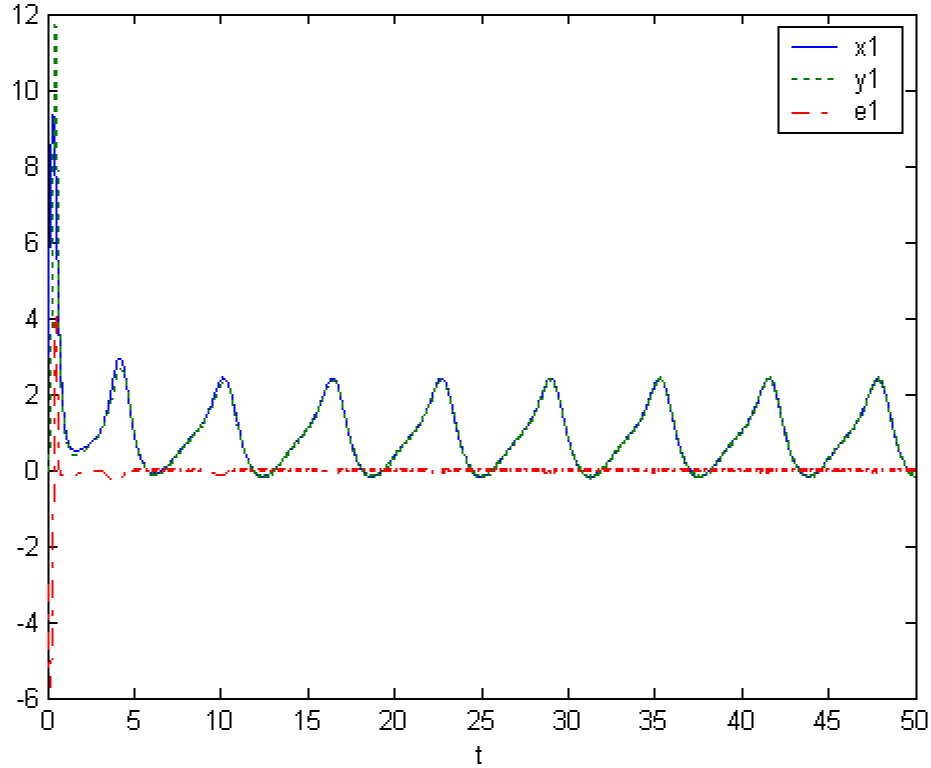


**Figure 1** System state $x_1$, NN state $y_1$ and error state $e_1$ with sinusoidal input

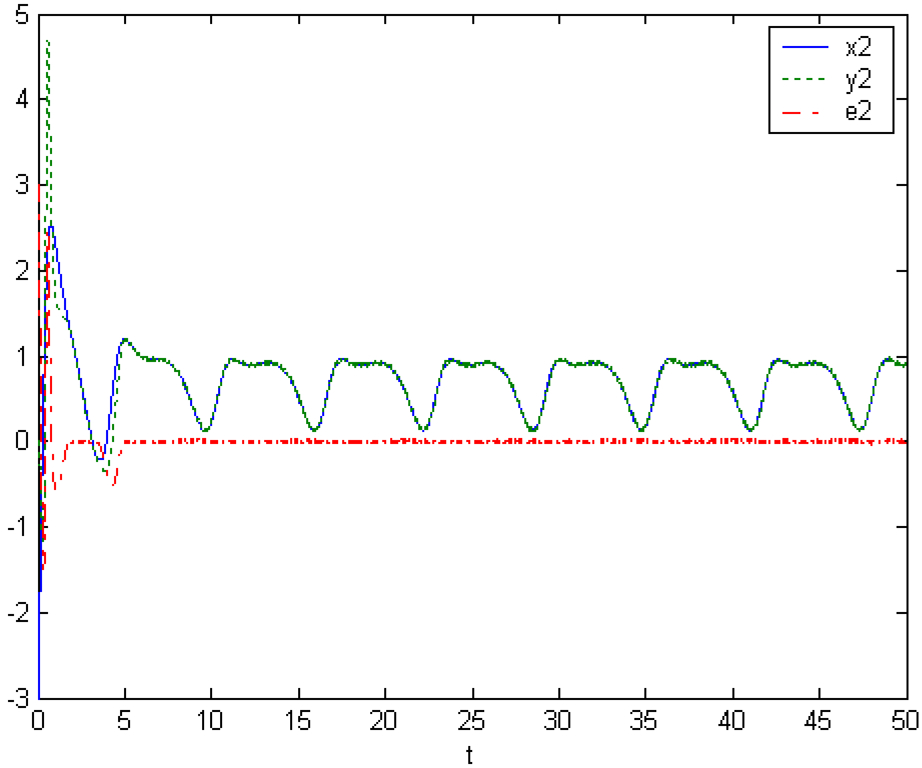


**Figure 2** System state $x_2$, NN state $y_2$ and error state $e_2$ with Cosine input

The control task was to drive the system state $x(t) = (x_1(t), x_2(t))^T$ to follow a pre-specified trajectory vector $x^*(t) = (\sin(t), \cos(t))^T$ with proper dynamic and static characteristics.

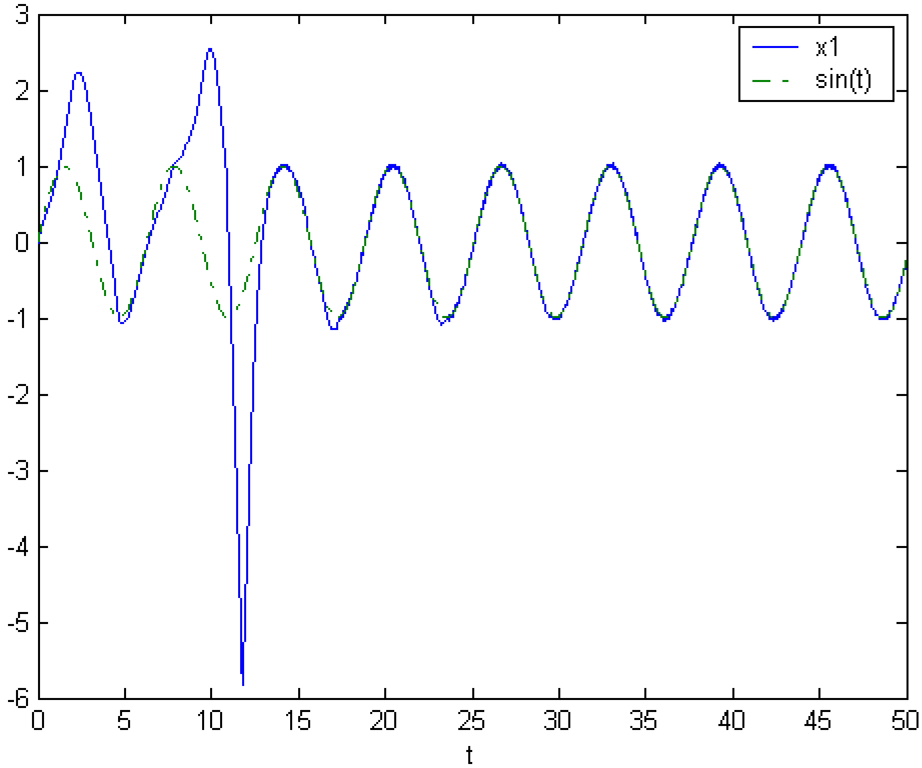


**Figure 3** System state $x_1$, and pre-specified trajectory $x_1^* = \sin(t)$

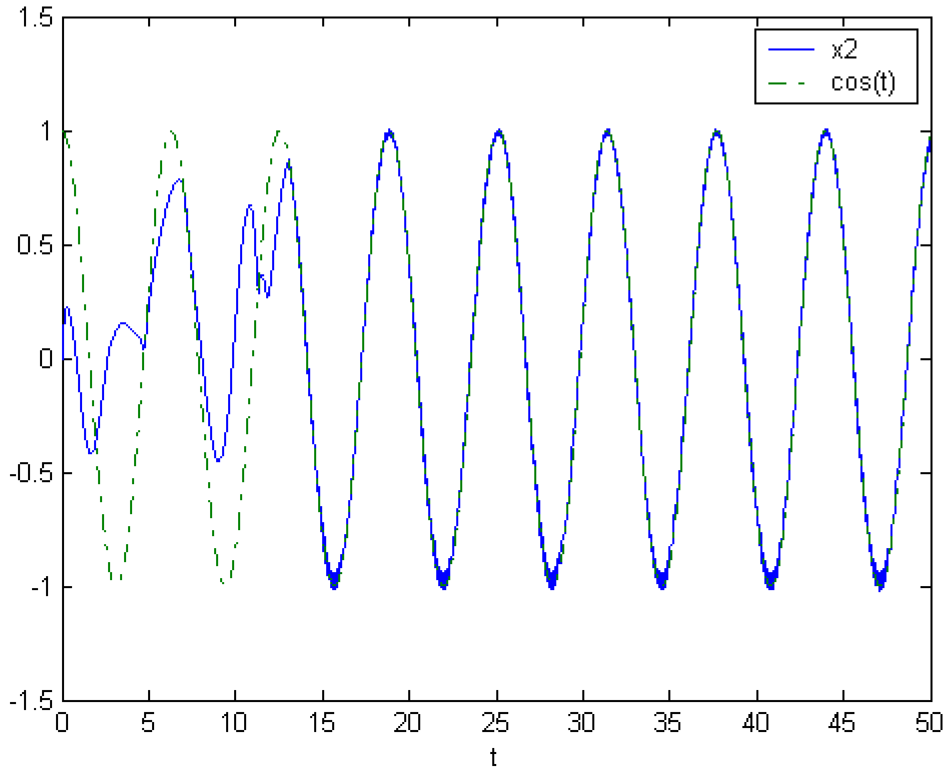


**Figure 4** System state $x_2$, and pre-specified trajectory $x_2^* = \cos(t)$

The simulated results are shown in figures 1 – 4 to indicate the excellent approximation of using the neural network for identification and control, which the errors were quickly converged to zero. Further it can be observed with the excellent performance of the system states following the pre-specified trajectory.

**Example 2** The complex nonlinear system was described by

$$\dot{x}_1 = x_1(t) - x_1(t) * x_2(t) - x_1(t-\tau(t)) * x_2(t-\tau(t))$$

$$\dot{x}_2 = x_1(t) + x_2(t) + x_2(t) * u_2 + u_2$$

Obviously the input has implicit relationship with state $x_2(t)$, that is the input cannot be explicitly separated from state $x_2(t)$. The recurrent neural network was structured with 2 layers, 10 neurons for input layer, 2 neurons for output layer, and the activation functions were hyperbolic tangent, $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$. Assigned the initials $\Gamma = 0.1 * I, \sigma = 1$, $C(0) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}, A(0), B(0), S(0)$ as null matrices and null vectors, initial conditions of plant were set $(x_1(t), x_2(t))^T = (3,3)^T$ respectively. Then run the on-line identification and controller design procedure listed in last section. The input and output were sampled every 0.01s and the whole process was run for 30 seconds.

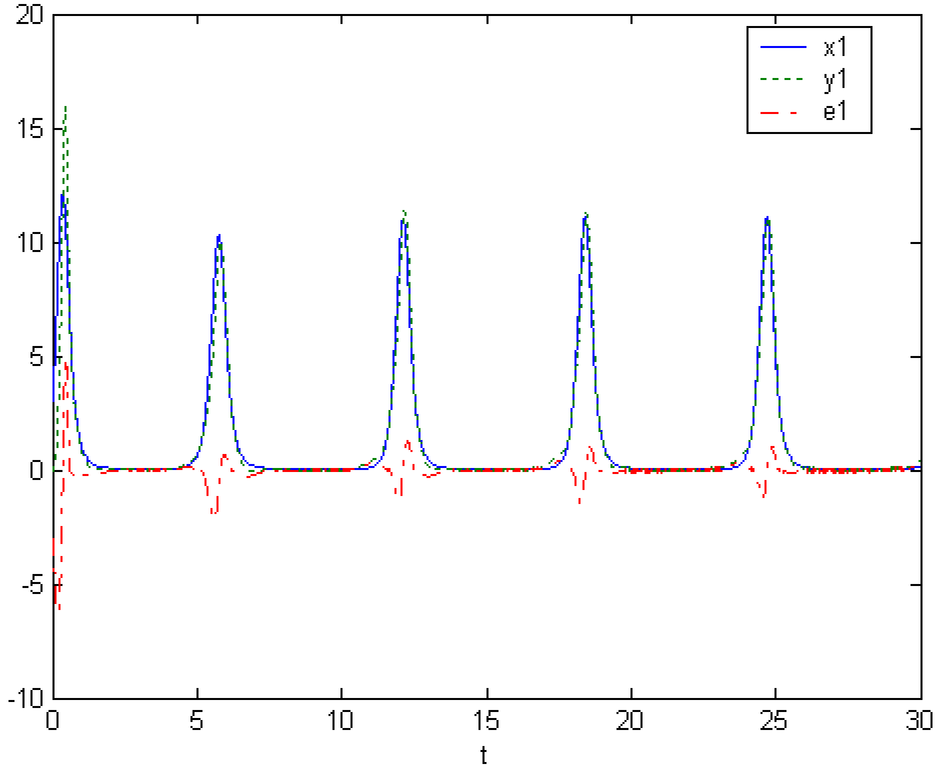


**Figure 5** System state $x_1$, NN state $y_1$ and error state $e_1$ with sinusoidal input

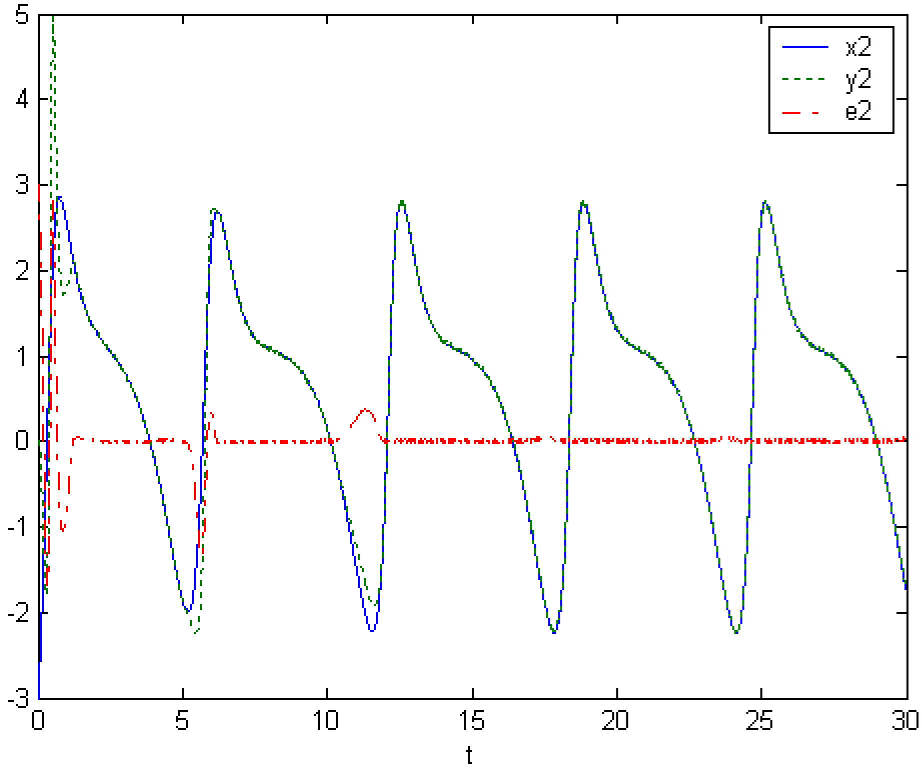


**Figure 6** System state $x_2$, NN state $y_2$ and error state $e_2$ with sinusoidal input

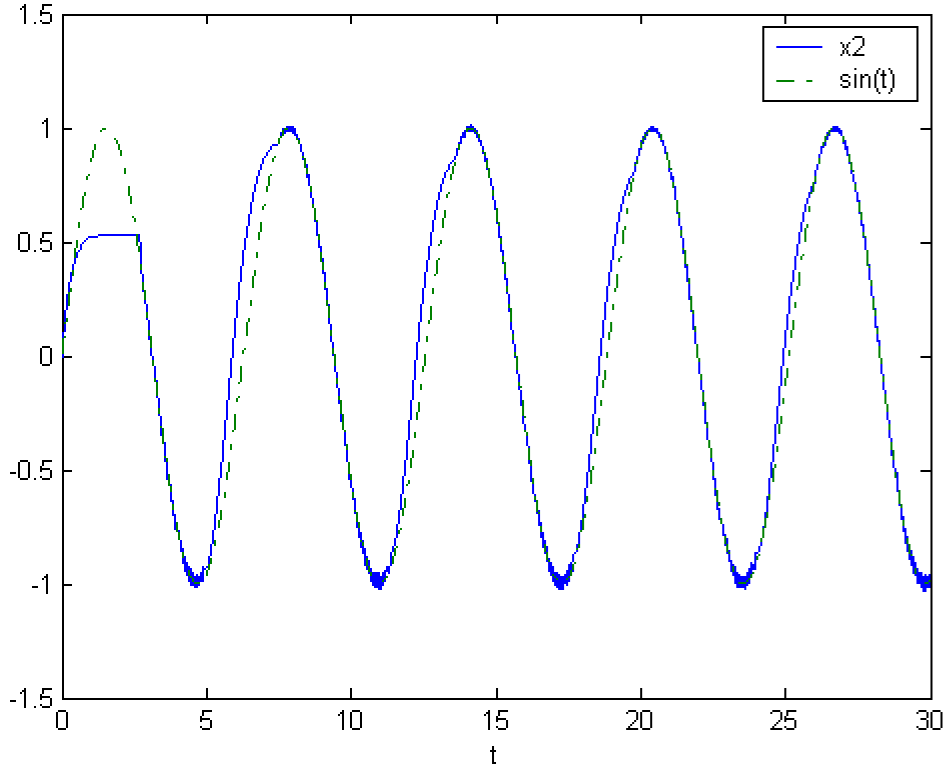


**Figure 7** System state $x_2$, and pre-specified trajectory $x_2^* = \sin(t)$

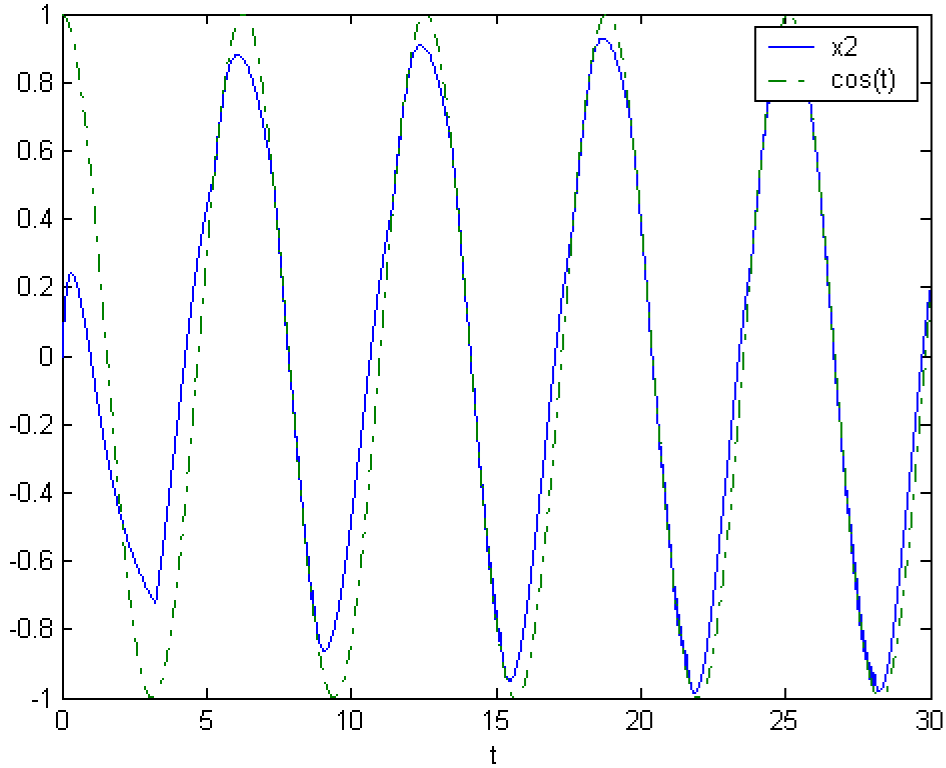


**Figure 8** System state $x_2$, and pre-specified trajectory $x_2^* = \cos(t)$

Once again as demonstrated in example 1, the second case study is shown in figures 5 – 8 to indicate the excellent approximation of using the neural network for identification and control, which the errors were quickly converged to zero. Further it can be observed with the excellent performance of the system states following the pre-specified trajectory.

## 7. Conclusion and future work

The main contributions of the study can be recognized as the integration and upgradation of neural network as a reference mode to relive the complexity in nonlinear control system design. It has been evidenced that there has been no unique approach to use analytical nonlinear model to design controllers. This approach has established a generic strategy using neural network to approximate nonlinear systems and then using the reference plus linear control system design methodology to design nonlinear control systems. The final controller output is formulated as resolving a polynomial equation root. Further the newly structured neural network and its corresponding adaptive laws have embedded more practical factors for applications. The simulation studies have indicated the success of controlling the state of the plants following desired trajectories for proper dynamic response and to origin point for steady state settlement.

There still exist a number of open studies in the follow up research. In regarding of system complexity, the potential studies will be oriented to control those systems with unknown time delay, non-smooth nonlinear nonlinearities (dead-zone, say). In regarding of neural network enhancement to approximate complex systems, the potential studies will cover to investigate new structure of NNs and new adaptive laws to adjust the weights of the NNs in terms of more efficient and effective. In regarding of the nonlinear polynomial root solver for obtaining controller output, the potential studies will cover to revise dichotomy searching mechanism to accommodate online computational accuracy and time.

## 8. Acknowledgement

This work was supported by NSFC Grant 60604004, 60843004, the Hebei Province Nature Fund Grant (F2007000637). Further the authors are grateful to the editor and the anonymous reviewers for their helpful comments and constructive suggestions with regarding to the revision of the paper.

## 9. References

Arik, S. (2003). Global asymptotic stability of a larger class of delayed neural networks. IEEE Circuits and Systems, 5: 721-724

Burden, R. L. and Faires, J. D. (2004). Numerical Analysis. Brooks Cole. Pacific Grove, California, United States

Ge, S. S., and Wang C. (2002). Direct Adaptive NN control of a Class of Nonlinear Systems. IEEE on Neural Networks, 13(1) 214-220

Ge, S. S., Hong, F. and Lee, T. H. (2003). Adaptive neural network control of nonlinear systems with unknown time delays. IEEE on TRANS. Automatic Control, 48 (11), 2004-2010

Ge, S.S., Hong, F. and Lee, T.H. (2005). Robust adaptive control of nonlinear systems with unknown time delays. Automatica,41,1181-1190

Hua, C. C., Long, C.N. & Guan, X. P. (2006). New results on stability analysis of neural networks with time-varying delays Physics Letter A, 352: 335-240

Lewis, F. L., Aydin Y. and Liu, K. (1996). Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance, IEEE TRAN. On Neural Networks, 7 (2) 388-399

Liao, X. F., Yu, X. B. & Chen, G. R. (2002). Delay-dependent exponential stability analysis of delayed cellular neural networks: IEEE Communications, Circuits and Systems and West Sino Expositions, 2:1657-1661

Liu, S., Wang, Y. J. & Zhu, Q.M. (2008). Development of a new EDRNN procedure in control of arm trajectories, Neurocompting, 72, 490-499

Noroozi, N., Roopaei, M., and Jahromi,M. Z. (2009). Adaptive fuzzy sliding model control scheme for uncertain systems, Commun Nonlinear Sci Numer Simulat 14, 3978-3992

Poznyak, A.S., Sanchez, E.N. and Yu, W. (2001). Differential Neural Networks for Robust Nonlinear Control: Identification, State Estimation and Trajectory Tracking, World Scientific.

Polycarpou , M. M. (1996). Stable adaptive neural scheme for nonlinear systems. IEEE Trans. on Neural Networks, 7(3),447-451

Roska, T., Boros, T., Thiran, P. & Chua, L.O. (1999). Detecting simple motion using cellular neural networks: Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications, 127-138

Rubio, J. D. J. & Yu, W. (2007a). Nonlinear system identification with recurrent neural networks and dead-zone Kalman filter algorithm. Neurocomputing, 70:2460-2466

Rubio, J. D. J. & Yu, W. (2007b). Stability Analysis of Nonlinear System Identification via Delayed Neural Networks. IEEE Trans. Circuits System, 2(54):161-165

Singh, V. (2004). A Generalized LMI-Based approach to the global asymptotic stability of cellular neural networks: IEEE Trans. on Neural Networks, 15:223-225

Wang, D. & Huang, J. (2002). Adaptive neural network control for a class of uncertain nonlinear systems in pure-feedback form. Automatica, 38,1365-1372

Wang, Y. Q., Jiang, C. H., Zhou, D.H. and Gao, F. R. (2008). Variable structure control for a class of nonlinear systems with mismatched uncertainties. Applied Mathematics and Computation, 200:387-400.

Yang, C. G., Ge, S. S. & Lee, T. H. (2009). Output feedback adaptive control of a class of nonlinear discrete-time systems with unknown control directions. Automatica, 45,270-276

Yu, W. (2004). Nonlinear system identification using discrete-time recurrent neural networks with stable learning algorithms. Information sciences, 158, 131-147

Yu, W. (2006). Multiple recurrent neural networks for stable adaptive control. Neurocomputing, 70,430-444

Zhang, L.F., Zhu Q.M. & Longden, A. (2009). A correlation-Test-Based validation procedure for identified neural networks. IEEE Trans. on Neural Networks, 20(1), 1-13

Zhang, T., Ge, S.S. & Hang, C.C. (1999). Design and performance analysis of a direct adaptive controller for nonlinear systems. Automatica, 35, 1809-1817

Zhang, T.P. & Ge, S. S. (2007). Adaptive neural control of MIMO nonlinear state time-varying delay systems with unknown dead-zones and gain signs. Automatica , 43, 1021-1033

Zhou, D. M., Zhang, L.M. & Zhao, D.F. (2003). Global exponential stability for re-current neural networks with a general class of activation functions and variable delays: IEEE Int. Conf. Neural Networks & Signal Processing, 108-111

**Appendix**

To find a numerical solution of a nonlinear function by iterative computation in dichotomy principle (Burden and Faires 2004), consider a nonlinear function below

$$f(x) = 0$$

where the function $f(\cdot)$ is smooth. There exist an interval, to contain one of the solutions (roots) of the function, with limits $a, b$, where $a < b$ and $f(a) < 0$, $f(b) > 0$, and $\xi \in (a, b)$.

The solution $\xi$ of the function $f(\cdot)$ by the dichotomy principle can be obtained by the following iterative algorithm

Step 1. Search the root by calculating $c = \frac{a+b}{2}$ and $f(c)$

Step 2. Check if $c$ is a proper approximate to the root by checking

If $|f(c)| \leq \varepsilon$ (a pre-specified error threshold, normally a very small constant), then $c$ is accepted as the approximate to the root.

Otherwise

If $f(c) > 0$ to indicate the root lying in the interval $(a, c)$, then $b = c$, go to step 1.

If $f(c) < 0$ to indicate the root lying in the interval $(c, b)$, then $a = c$, go to step 1.

The algorithm can be stopped by either a pre-specified error threshold, say 0.01, or a pre-specified iteration time, say 100.