Robust Adaptive Backstepping Controller for a Class of Nonlinear Cascade Systems via Fuzzy Neural Networks

Ching-Hung Lee*, Bo-Ren Chung and Jen-Chieh Chien
Department of Electrical Engineering, Yuan Ze University, Chungli, Taoyuan 320 Taiwan

Abstract: In this paper, a robust adaptive backstepping control scheme using fuzzy neural networks, called FNN-ABC, is proposed for a class of nonlinear uncertain systems with cascade structure. Each subsystem is in the form of lower triangular and non-affine systems which contains of external disturbance, uncertainties, or parameters variations. By the backstepping approach, a fuzzy neural network (FNN) based robust adaptive controller is designed in a step by step manner for each subsystem. Two kinds of FNN systems are used to estimate the subsystems’ unknown functions. According to the FNNs’ estimations, the FNN-ABC control input can be generated by Lyapunov approach such that system output follows the desired trajectory. To enhance the control performance (or FNNs’ approximation accuracy), a Taylor expansion method are adopted to derive the update laws of FNNs’ antecedent-part parameters. Based on the Lyapunov approach, the adaptive laws of FNNs’ parameters and stability analysis of closed-loop system are obtained. Finally, the proposed FNN-ABC is applied to the tracking control of a single-link flexible-joint robot. A simulation study is proposed to illustrate the performances of our approach.

Keywords: Nonlinear cascade system, backstepping, fuzzy neural network, adaptive control, Lyapunov

1. INTRODUCTION

Over the past decade, both fuzzy logic systems and neural networks have found extensive applications for plants that are complex and ill-defined. In most of these cases, they have been used as universal approximators [1-6]. Recently, they have been merged to obtain the fuzzy neural network (FNN) systems (or neuro-fuzzy systems) and have the advantages of both fuzzy logic and neural networks. Based on the approximation ability, many adaptive control techniques are accompanied with them for approximation of system functions or controllers [4, 7-20].

The backstepping design provides a systematic framework and recursive design methodology for nonlinear systems [7, 8, 11, 16, 21]. The design procedure treats the state variables as virtual control inputs, to design step by step to design the virtual controllers and prove the stability by Lyapunov stability theorem. The actual control input can be obtained in the final step. However, the major constraint is that the system functions must be known. If the internal uncertainty and external disturbance exist, then the system maybe become unstable. Therefore, the FNNs are used to approximate the unknown functions to solve this problem.

Recently, newly backstepping control schemes that combine the backstepping technique and neural networks were proposed for nonlinear systems with the lower triangular form (or the nested lower triangular form) [6, 11, 15, 19]. However, most of these approaches still have some constraints, e.g., the gain functions of control input were assumed to be constants (or exactly known). Besides, they are also limited to the feedback linearizable nonlinear systems, which mean that the unknown nonlinearities must satisfy a matching condition. However, many real-world systems do not satisfy such a condition, especial non-affine system. The problem of controlling non-affine system is a difficult one [14, 17-20]. To treat the problem, widely used in practice, is that based or linearization of the nonlinear plant model around an operating point. While the linearization may result in the design of sufficiently accurate controllers in the case of stabilization, in the case of tracking of desired trajectories the problem becomes much more difficult. Hence there is a clear need for the development of systematic control design techniques for nonlinear models that are non-affine in $u$, and are suitable for the case of tracking of desired trajectories.

In this study, we proposed an FNN-based robust adaptive backstepping control scheme (FNN-ABC) for a class of nonlinear cascade systems with lower triangular non-affine form, i.e., it is not in feedback linearizable form. This controller is designed in the concept of the backstepping control procedure, and the FNNs are used to estimate the unknown system functions online. The adaptive laws of FNNs’ parameter are obtained by the Lyapunov approach; therefore the tracking error can be guaranteed uniformly ultimately bounded (UUB).

This paper is organized as follows. Section 2 introduces the problem formulation, design procedure of backstepping controller, and FNN systems. The FNN-ABC robust control
scheme is introduced in Section 3. The stability analysis is
guaranteed by the Lyapunov theorem. Section 4 shows the
simulation results of a single-link flexible-joint robot. Finally,
conclusion is given in Section 5.

2. PRELIMINARIES

2.1. Problem Formulation and Backstepping Controller

Consider the following $n$th-order nonlinear cascade system as

\[ \begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
& \vdots \\
\dot{x}_p &= F_1(\xi, x_{p+1}, d_1) \\
\dot{x}_{p+1} &= x_{p+2} \\
\dot{x}_{p+2} &= x_{p+3} \\
& \vdots \\
\dot{x}_m &= F_2(\xi, u, d_2) \\
y &= x_1
\end{align*} \]

where $\xi = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$, $i = 1, 2, \ldots, n$ denote the
system state vectors; $u \in \mathbb{R}$ and $y \in \mathbb{R}$ are the input and output,
respectively. $F_1$ and $d_1$ ($i = 1, 2$) are the nonlinear smooth
functions and bounded external disturbance, respectively.
System (1) cannot be stabilized by feedback linearization
approach. Note that system (1) can be viewed
as two connected subsystems.

**Assumption 1:** The inequality
\[
\frac{\partial F_1(\xi, x_{p+1}, d_1)}{\partial x_{p+1}} \neq 0
\]
holds for all $\xi \in \mathbb{R}^{p+1}$.

**Assumption 2:** The inequality
\[
\frac{\partial F_2(\xi, u, d_2)}{\partial u} \neq 0
\]
holds for all $(\xi, u) \in \Omega_x \times \mathbb{R}$ with a controllability region
$\Omega_x$.

The above assumptions imply that the
\[
\frac{\partial F_1(\xi, x_{p+1}, d_1)}{\partial x_{p+1}}, \quad \frac{\partial F_2(\xi, u, d_2)}{\partial u}
\]
are strictly either positive or negative.

We define the state transformation $z_i = x_i, q_j = x_{pj}, i = 1, \ldots, p, j = 1, \ldots, m, p + m = n$. System (1) can be viewed as
two connected subsystems

\[ \begin{align*}
\dot{z}_i &= z_2 \\
\dot{z}_2 &= z_3 \\
& \vdots \\
\dot{z}_p &= F_1(\xi, q_1, d_1) \\
\dot{q}_1 &= q_2 \\
\dot{q}_2 &= q_3 \\
& \vdots \\
\dot{q}_m &= F_2(\xi, q_m, u, d_2) \\
y &= x_1
\end{align*} \]

According to the concept of literature [14], subsystems
(2) and (3) can be written as

\[ \begin{align*}
\dot{z}_i &= z_2 \\
\dot{z}_2 &= z_3 \\
& \vdots \\
\dot{z}_p &= h_1(\xi, q_1, d_1) + c_1 q_1 \\
\dot{q}_1 &= q_2 \\
\dot{q}_2 &= q_3 \\
& \vdots \\
\dot{q}_m &= h_2(\xi, q_m, u, d_2) + c_2 u
\end{align*} \]

where
\[ h_i = F_1(\xi, q_1, d_1) - c_1 q_1, \]
\[ h_2 = F_2(\xi, q_m, u, d_2) - c_2 u, \]
and $c_1, c_2$ are design nonzero constants. Then subsystems (4) and (5) are lower triangular
and affine-like. Let $z = [z_1, z_2, \ldots, z_{p+1}]^T = [z_1, z_2, \ldots, z_n]^T$
\[ \in \mathbb{R}^{p+1}, \quad q = [q_1, q_2, \ldots, q_{m+1}]^T = [q_1, q_2, \ldots, q_m]^T \in \mathbb{R}^m. \]
The desired output $z_d$ of $z$-subsystem and its derivative are assumed to be bounded. In this paper, we use the
backstepping design procedure to design the stabilizing controller for system (4)-(5).

**Step 1:** Define the tracking error $e = y - y_d = z - z_d$ and the corresponding error vector as
\[ e = [e_1, \dot{e}_1, \ldots, e_1^{(p-1)}]^T \in \mathbb{R}^p, \]
\[ e^{(p)} = z^{(p)}_2 - z^{(p)}_d = h_1 + c_1 q_1 - z^{(p)}_d. \]

There exists a stabilizing virtual controller for (6)
\[ q_d = \frac{1}{c_1} \left(-k_1^T e + e^{(p)} - h_1\right) \]
where $k_i = [k_{i,2}, \ldots, k_{i,1}]^T$ is determined such that $s^p + k_{i,1}s^{p-1} + \ldots + k_{i,p}$ is Hurwitz. Let $e_q = q_i - q_{i,m}$, thus

$$e_q^{(p)} = -k_{i}^{(p)} e_q + c_i e_q$$

(8)

or

$$\dot{e}_q = A_i e_q + b_i c_i e_q$$

(9)

where

$$A_i = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
-k_{i,p} & -k_{i,p-1} & \cdots & -k_{i,2} & -k_{i,1}
\end{bmatrix}$$

and $b_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}$.

Since $A_i$ is stable (Hurwitz), there exists a positive definite symmetric matrix $P_i$ satisfies

$$A_i^T P_i + P_i A_i = -Q_i$$

(10)

for an arbitrary positive definite matrix $Q_i$. The Lyapunov candidate function is chosen as

$$V_i = \frac{1}{2} e^T P_i e_i$$

(11)

such that

$$\dot{V}_i = e_i^T P_i \dot{e}_i = -\frac{1}{2} e_i^T Q_i e_i + e_i^T P_i b_i c_i e_i^T e_i.$$

(12)

Step 2: As the definition in step 1, $e_q = q_i - q_{i,m}$ and the corresponding error vector as $e_q = [e_q^T \dot{e}_q \cdots e_q^{(m-1)}]^T \in \mathbb{R}^n$.

The derivative of $e_q^{(m-1)}$ is

$$e_q^{(m)} = q_i^{(m)} - q_{i,m}^{(m)} = h_2 + c_i u - q_{i,m}^{(m)}.$$  

(13)

Actual control input $u$ is designed as

$$u = \frac{1}{c_2} \left( -k_2^T e_q - e_1^T P_i b_i c_i e_q + q_{i,1}^{(m)} - h_2 \right)$$

(14)

where $k_2 = [k_{2,n}, \ldots, k_{2,1}]^T$ is determined such that $s^m + k_{2,1}s^{m-1} + \ldots + k_{2,m}$ is Hurwitz. Substitute (14) into (13)

$$e_q^{(m)} = -k_2^T e_q - \frac{e_1^T P_i b_i c_i e_q}{e_1^T P_i b_i}$$

(15)

or

$$\dot{e}_q = A_2 e_q + b_2 \left( -\frac{e_1^T P_i b_i c_i e_q}{e_1^T P_i b_i} \right)$$

(16)

where

$$A_2 = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
-k_{2,m} & -k_{2,m-1} & \cdots & -k_{2,2} & -k_{2,1}
\end{bmatrix}$$

and $b_2 = \begin{bmatrix} 0 \\ \vdots \\ 1 \end{bmatrix}$.

Therefore, the asymptotically stability of system (1) is guaranteed. However, since $F_i (i=1, 2)$ are not exactly known that include both internal uncertainties and external disturbance, i.e., $h_i (i=1, 2)$ are also unknown and controller (14) maybe results unstable. Therefore, fuzzy neural network (FNN) systems are used to estimate the unknown functions $h_i$ and $h_2$. In addition, $h_d$ depends on $n$ and $u$ appears in both the left- and right-hand sides of (14). Therefore, a dynamic neural network (DNN) is introduced to estimate $h_2$.

![Figure 1: Schematic Diagram of Recurrent Fuzzy Neural Network](image)

(R Denotes the Amount of Rule Numbers)

### 2.2. Fuzzy Neural Network Systems

Herein, we introduce the used recurrent fuzzy neural network (RFNN) systems [9]. The schematic diagram of RFNN is shown in Fig. 1. There are four layers. Layer 1 is input layer and each node represents an input linguistic variable. The nodes in this layer only transmit input variables to the next layer, i.e., $O_i^{(1)}(k) = x_i(k)$. Layer 2- membership layer, is used to calculate Gaussian membership grade, i.e.,

$$O_i^{(2)}(k) = \exp \left[ \frac{\left( (O_i^{(1)}(k-1) - \theta_i) + O_i^{(1)}(k) \right) - m_i}{\sigma_i^2} \right]^2$$

(20)
where \( m_i \) and \( \sigma_i \) are the center and the width of the Gaussian function, \( \theta_j \) is the adjustable feedback parameter. It is worth to mention, if we set \( \theta = 0 \), the RFNN can be reduced as the FNN system [4, 9]. Nodes in layer 2 represent the terms of the respective linguistic variables. Nodes in layer 3 represent fuzzy rules, i.e., layer 3 forms the fuzzy rule base. Links before layer 3 represent the antecedent part of fuzzy rules, and the links after layer 3 represent the consequent part. The product operation is used, i.e.,

\[
O_{\theta}^{(j)}(k) = \prod_{i=1}^{J} O_{\theta}^{(j)}(k).
\]

Layer 4 is the output layer. Each node is for actual output to be pumped out this system. The links between layer 3 and layer 4 are connected by the weighting value \( w_j \), i.e.,

\[
y_p = O_{\psi}^{(4)}(k) = \sum_{j=1}^{G} w_j O_{\theta}^{(3)}(k) = w^T \psi = \frac{\sum_{j=1}^{G} w_j O_{\theta}^{(3)}(k)}{\sum_{j=1}^{G} O_{\theta}^{(3)}(k)}
\]

where \( w = [w_1, w_2, \ldots, w_G]^T \) is the weighting vector; \( \psi = [\psi^1, \psi^2, \ldots, \psi^G] \), \( \psi^i = O_{\theta}^{(3)}(k) / \sum_{j=1}^{G} O_{\theta}^{(3)}(k) \) represents the normalized value. By the results of [9], the RFNN is suitable for on-line estimation. That is, for any given real function \( h_i : \Phi \rightarrow \Phi \), there exists optimal parameters, the function \( h_i \) can be described by the output of the RFNN with a reconstruction error. From (22), the FNN’s and RFNN’s outputs are

\[
\hat{h}_i(x) = y_{pi} = \hat{w}_i^T \hat{\psi}_i = \hat{w}_i^T \psi \left( m_i, \sigma^i, \theta^i \right)
\]

therefore, the unknown functions \( h_i \) are

\[
h_i = w_i^T \psi_i + \Delta_i = w_i^T \psi \left( m_i^o, \sigma_i^o, \theta_i^o \right) + \Delta_i
\]

where \( \Delta (i=1, 2) \) are the approximation errors; \( w_i^*, m_i^o, \sigma_i^o, \theta_i^o \) are the optimal vectors of \( w_i, m_i, \sigma_i, \theta_i \) of the \( i \)th FNN, respectively. From (23), the function estimated error \( \hat{h}_i \) satisfies

\[
\begin{align*}
\hat{h}_i &= h_i - \hat{h}_i \\
&= w_i^T \psi_i + \Delta_i - \hat{w}_i^T \hat{\psi}_i \\
&= w_i^T \psi_i - \hat{w}_i^T \hat{\psi}_i + \hat{w}_i^T \hat{\psi}_i + \Delta_i \\
&= w_i^T \psi_i + \hat{w}_i^T \hat{\psi}_i + \Delta_i, \quad i = 1, 2.
\end{align*}
\]

The linearization technique is employed to partially linear form [12], we have

\[
\Phi_i = \begin{bmatrix}
\psi_i^{(0)} \\
\psi_i^{(1)} \\
\vdots \\
\psi_i^{(g)}
\end{bmatrix}
= \begin{bmatrix}
\frac{\partial \psi_i^{(0)}}{\partial m_i^{(0)}} \\
\frac{\partial \psi_i^{(0)}}{\partial m_i^{(1)}} \\
\frac{\partial \psi_i^{(0)}}{\partial m_i^{(2)}} \\
\vdots \\
\frac{\partial \psi_i^{(0)}}{\partial m_i^{(g)}} \\
\frac{\partial \psi_i^{(1)}}{\partial m_i^{(0)}} \\
\frac{\partial \psi_i^{(1)}}{\partial m_i^{(1)}} \\
\frac{\partial \psi_i^{(1)}}{\partial m_i^{(2)}} \\
\vdots \\
\frac{\partial \psi_i^{(1)}}{\partial m_i^{(g)}} \\
\frac{\partial \psi_i^{(g)}}{\partial m_i^{(0)}} \\
\frac{\partial \psi_i^{(g)}}{\partial m_i^{(1)}} \\
\frac{\partial \psi_i^{(g)}}{\partial m_i^{(2)}} \\
\vdots \\
\frac{\partial \psi_i^{(g)}}{\partial m_i^{(g)}}
\end{bmatrix}
\begin{bmatrix}
m_i^{(0)} - \hat{m}_i \ \\
\sigma_i^{(0)} - \hat{\sigma}_i \\
\sigma_i^{(1)} - \hat{\sigma}_i \\
\vdots \\
\sigma_i^{(g)} - \hat{\sigma}_i
\end{bmatrix}
+ H_i
\]

(26a)

\[
\Phi_i = \begin{bmatrix}
\psi_i^{(0)} \\
\psi_i^{(1)} \\
\vdots \\
\psi_i^{(g)}
\end{bmatrix}
= \begin{bmatrix}
\frac{\partial \psi_i^{(0)}}{\partial \theta_i^{(0)}} \\
\frac{\partial \psi_i^{(0)}}{\partial \theta_i^{(1)}} \\
\frac{\partial \psi_i^{(0)}}{\partial \theta_i^{(2)}} \\
\vdots \\
\frac{\partial \psi_i^{(0)}}{\partial \theta_i^{(g)}} \\
\frac{\partial \psi_i^{(1)}}{\partial \theta_i^{(0)}} \\
\frac{\partial \psi_i^{(1)}}{\partial \theta_i^{(1)}} \\
\frac{\partial \psi_i^{(1)}}{\partial \theta_i^{(2)}} \\
\vdots \\
\frac{\partial \psi_i^{(1)}}{\partial \theta_i^{(g)}} \\
\frac{\partial \psi_i^{(g)}}{\partial \theta_i^{(0)}} \\
\frac{\partial \psi_i^{(g)}}{\partial \theta_i^{(1)}} \\
\frac{\partial \psi_i^{(g)}}{\partial \theta_i^{(2)}} \\
\vdots \\
\frac{\partial \psi_i^{(g)}}{\partial \theta_i^{(g)}}
\end{bmatrix}
\begin{bmatrix}
\sigma_i^{(0)} - \hat{\sigma}_i \\
\sigma_i^{(1)} - \hat{\sigma}_i \\
\sigma_i^{(2)} - \hat{\sigma}_i \\
\vdots \\
\sigma_i^{(g)} - \hat{\sigma}_i
\end{bmatrix}
+ H_i
\]

(26b)

or

\[
\Phi_i = O_{m_i} m_i^{(i)} + O_{\sigma_i} \sigma_i^{(i)} + H_i
\]

(27a)

\[
\Phi_i = O_{\sigma_i} m_i^{(i)} + O_{\sigma_i} \sigma_i^{(i)} + H_i
\]

(27b)

where \( O_{m_i}, O_{\sigma_i}, \) and \( O_{\theta_i} \) are derivatives of \( \Phi_i \), with respective to \( m_i, \sigma_i, \theta_i \), and \( \sigma_i, \theta_i \) at \( m_i^{(i)}, \sigma_i^{(i)}, \theta_i^{(i)} \) is \( \left[ m_i^{(i)}, \sigma_i^{(i)}, \theta_i^{(i)} \right] \); \( \hat{m}_i = \hat{w}_i^T \left( \hat{m}_i - m_i \right) \), \( \sigma_i = \hat{\sigma}_i - \hat{\sigma}_i \), \( \hat{\sigma}_i = \sigma_i^{(n)} - \hat{\sigma}_i^{(n)} \); \( H_i, i = 1, 2 \) are higher-order terms. Substitute (27) into (25), we have

\[
\begin{align*}
\hat{h}_i &= \hat{w}_i^T \Phi_i + \hat{w}_i^T \hat{\psi}_i + \Delta_i \\
&= \hat{w}_i^T \left( O_{m_i} \hat{m}_i + O_{\sigma_i} \hat{\sigma}_i + H_i \right) + \hat{w}_i^T \hat{\psi}_i + \Delta_i \\
&= \hat{w}_i^T \left( \hat{\psi}_i - O_{m_i} \hat{m}_i - O_{\sigma_i} \hat{\sigma}_i \right) + \hat{w}_i^T \left( O_{m_i} \hat{m}_i + O_{\sigma_i} \hat{\sigma}_i \right) + \Delta_i \\
&= \hat{w}_i^T \left( \hat{\psi}_i - O_{m_i} \hat{m}_i - O_{\sigma_i} \hat{\sigma}_i \right) + \hat{w}_i^T \left( O_{m_i} \hat{m}_i + O_{\sigma_i} \hat{\sigma}_i \right) + \Delta_i,
\end{align*}
\]

(28a)

International Journal of Computational Intelligence in Control
are the FNNs’ approximation of function \( h_i \) \((i = 1, 2)\), \( \hat{\delta}_i \) are the estimations of \( \delta_i \) \((i=1, 2)\), the estimated errors is defined as \( \hat{\delta}_i = \delta_i - \hat{\delta}_i \). The following corresponding parameter update laws of FNNs are chosen
\[
\dot{\theta}_i = r_{w_i} e_i^T P_i b_i \left( \tilde{\psi}_i - O_i^T \tilde{m}_i - O_i^T \tilde{\sigma}_i \right)
\]
\[
\dot{m}_i = r_{m_i} e_i^T P_i b_i O_{m_i} \hat{w}_i
\]
\[
\dot{\sigma}_i = r_{\sigma_i} e_i^T P_i b_i O_{\sigma_i} \hat{w}_i
\]
\[
\dot{\hat{\delta}}_i = a_i \left| e_i^T P_i b_i \right|
\]
where \( r_{w_i}, r_{m_i}, r_{\sigma_j} \) \((i = 1, 2, \sigma_j = 1, 2)\) are the parameter adaptive rates. Then, the tracking error is guaranteed uniformly ultimately bounded (UUB) if \( J_1 > 0 \) and \( J_2 > 0 \) keep sufficiently small.

**Proof:** Herein, we prove it by the concept of backstepping approach.

**Step 1:** According to equations (4) and (29), the error dynamics is
\[
e^{(p)} = \hat{q}^{(p)} - \hat{q}^{(p)} = -k_1 \hat{e} + c_i \hat{e} + \tilde{h} + u_i
\]

or in matrix representation
\[
el = A_e e + b_1 \left( c_i \hat{e} + \tilde{h} + u_i \right)
\]

Design the Lyapunov candidate function
\[
V_1 = \frac{1}{2} e_i^T P_i e_i + \frac{1}{2 r_{m_i}} \hat{w}_i^T \hat{w}_i + \frac{1}{2 r_{\sigma_i}} m_i^T m_i
\]

such that
\[
V_1 = e_i^T P_i \dot{e}_i - \frac{1}{r_{m_i}} \hat{w}_i^T \hat{w}_i - \frac{1}{r_{\sigma_i}} m_i^T m_i - \frac{1}{r_{\sigma_i}} \sigma_i^T \sigma_i - \frac{1}{r_{\sigma_i}} \tilde{\delta}_i^T \tilde{\delta}_i
\]
Choose the Lyapunov candidate function

\[ V_1 = -\frac{1}{2}e_1^TQ_1e_1 + e_1^T P_1 b_1 \left[ c_1 e_1 + \dot{w}_1 (\text{sat}_2 - \dot{\hat{\delta}}_1) \right] \]

Substituting (33)-(35) into (45), we obtain

\[ V_2 = -\frac{1}{2}e_2^TQ_2e_2 + e_2^T P_2 b_2 \left[ (\omega_0 + \dot{u}_1) - \frac{1}{a_1} \hat{\delta}_3 \right] \]

By substituting (36)-(39) and (46) into (49), we obtain

\[ V_2 = -\frac{1}{2}e_2^TQ_2e_2 + e_2^T P_2 b_2 \left[ (\omega_0 + \dot{u}_1) - \frac{1}{a_2} \hat{\delta}_2 \right] \]

The robust controllers and \( \hat{\delta}_2 \)'s adaptive laws are designed as (31), (32), (40) and (41). Therefore, we obtain

\[ V_2 = -\frac{1}{2}e_2^TQ_2e_2 + e_2^T P_2 b_2 \left[ \frac{1}{a_2} \hat{\delta}_2 \right] \]

Choose the Lyapunov candidate function

\[ V_2 = V_1 + \frac{1}{2}e_2^T P_2 e_2 + \frac{1}{2} \dot{\omega}_2^T \dot{\omega}_2 + \frac{1}{2} \dot{b}_2^T \dot{b}_2 \]

such that

\[ V_2 = V_1 + e_1^T P_1 e_1 + \frac{1}{2} \dot{\omega}_1^T \dot{\omega}_1 + \frac{1}{2} \dot{b}_1^T \dot{b}_1 \]

implies \( V_2 < 0 \), the errors \( e_2 \) and \( \dot{e}_2 \) are guaranteed UUB within a small region if \( J_1 \) and \( J_2 \) are positive and keep arbitrarily small. This completes the proof. 

4. Simulation Result

In this section, a single-link flexible-joint robot is presented to show the performance of our proposed adaptive backstepping control scheme. The single-link flexible-joint
robot arm is shown in Fig. 3 [22] and the corresponding system model is

\begin{equation}
\begin{aligned}
I_1 \ddot{\theta}_1 + mgl \sin \theta_1 + k (\theta_1 - \theta_2) = 0 \\
I_2 \ddot{\theta}_2 - k (\theta_1 - \theta_2) = u
\end{aligned}
\end{equation}

where \( \theta_1 \) and \( \theta_2 \) are the angles of the link and of the motor shaft, respectively; \( m \) is the total mass of the link; \( I_1 \) and \( I_2 \) are rotor inertias of the link and the motor, respectively. \( l \) denote the distance from the motor shaft to the center of mass of the link, \( g \) is the acceleration constant due to gravity, \( k \) is the torsional spring constant, and \( u \) is the torque applied to the motor shaft. System (54) can be expressed in the form of (1) by \( x_1 = \theta_1, x_2 = \dot{\theta}_1, x_3 = \theta_2, x_4 = \dot{\theta}_2, \dot{x}_1 = x_2, \dot{x}_2 = -\frac{mgl}{I_1} \sin \theta_1 - \frac{k}{I_1} (\theta_1 - \theta_2), \dot{x}_3 = x_4, \dot{x}_4 = \frac{k}{I_2} (x_1 - x_3) + \frac{1}{I_2} u. \) (55)

The desired trajectory is \( x_{id} = \sin(t) \). The parameters values are given by \( mgl=10, k=100, I_1=100, I_2=10 \). The design parameters are \( k_1 = k_2 = [64, 16] \); \( c_1 = 1, c_2 = 0.1; P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \alpha_1 = \alpha_2 = 0.01; \hat{\delta}_1 (0) = \hat{\delta}_2 (0) = 2 \); \( J_1 = J_2 = 0.1 \). And the parameters of RFNN and FNN are chosen as shown in Table 1.

![Figure 2](image1)

**Figure 2:** The Robust FNN-based Adaptive Backstepping Control Scheme for Nonlinear Cascade System (1)

![Figure 3](image2)

**Figure 3:** Single-link Flexible-joint Robot Arm [22]

| Network Structure and Initializations of the FNN and RFNN |
|-------------|----------------|
| **FNN** | **RFNN** |
| Network structure | [3-15-5-1] | [4-20-5-1] |
| \( m_{ij} \) | [-2, -1, 0, 1, 2] | [-2, -1, 0, 1, 2] |
| \( \sigma_j \) | 1 | 1 |
| \( \phi_j \) | None | 0 |

**Simulation 1:** Stability illustration.

The comparison of these two approaches (FNN-ABC\(_1\) and FNN-ABC\(_2\)) is introduced in Table 2. The initial condition is set to be \( \bar{x}_0 (0) = [\pi/12, 0, \pi/6, 0]^T \). Figure 4 shows the simulation results with tracking error- RMSE=
4.1525\times 10^{-4}$. Figure 4(a) shows reference and output trajectories $x_d$ and $x_1$, respectively (dashed-line: desired trajectory; solid-line: FNN-ABC result). Figures 4(b) and 4(c) show the tracking error and control effort, respectively. These indicate that the closed-loop system is stable and tracking error approaching to zero. The tracking control can be achieved and performs well by our approaches.

**Simulation 2: Robustness of FNN-ABC**

Herein, the robustness of FNN-ABC is discussed. Parameters $I_1$ and $I_2$ of nominal plant have 20\% modeling error, i.e., nominal plant- $I_1=100$, $I_2=10$; actual plant- $I_1=80$, $I_2=8$. In addition, parameter $k$ has 50\% error, i.e., nominal plant- $k=100$; actual plant- $k=150$. Figure 8 shows the simulation results of FNN-ABC (state trajectory, tracking error, and control effort). The stabilizing time is larger than the simulation-1. The FNN-ABC performs well even if the system modeling error occurs.

5. CONCLUSION

This paper has presented an FNN-based adaptive backstepping control scheme (FNN-ABC) for a class of nonlinear uncertain systems with cascade and lower triangular non-affine form. Two kinds of FNN systems (FNN and RFNN) are used to estimate the unknown functions. By the Lyapunov stability approach, the stability of the control system is guaranteed and the adaptive laws of FNNs’ parameters have been obtained. Besides, Taylor expansion also has used to derive another kind of parameter adaptive laws for reducing the effect of initialization and improving the control performance. A single-link flexible-joint robot has been presented to illustrate the effectiveness and performance of FNN-ABC. By the backstepping design procedure, FNN-ABC approach can be extended to the nonlinear cascade system more than two subsystems.

**ACKNOWLEDGEMENT**

The authors would like to thank the Chief Editor and anonymous reviewers for their insightful comments and valuable suggestions. This work was partially supported by the National Science Council, Taiwan, R.O.C., under contract No: NSC-97-2221-E-155-033-MY3.

**REFERENCES**


Robust Adaptive Backstepping Controller for a Class of Nonlinear Cascade Systems via Fuzzy Neural Networks


[37] Peter Warr., How Do Infrastructure Projects Benefit The Poor?, Indian Development Review, Vol. 4, No. 2, pp. 225

[38] Abhimanyu Singh, Relevance of Marriage in Contemporary Times, International Review of Comparative Sociology


[40] M. Osintsev and V. Sobolev, Global Invariant Manifolds in a Problem of Kalman-Bucy Filtering for Gyroscopic Systems, Global and Stochastic Analysis

[41] Chun-Fei Hsu, Synchronization of Coupled Neurons Using a TSK-Type Recurrent Fuzzy Neural Network with Asymmetric Gaussian Membership Functions, International Journal of Computational Intelligence in Control


[46] V. Lokesha, Shruti R and Sunilkumar M. Hosamani, Operations on Gear Graph With Topological Indices, Indian Journal of Mathematics and Mathematical Sciences
