INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 12, NO. 3, SEPTEMBER 2007, 245-253

# A Composite Approach to Adaptive Neural Network Control of Unknown Flexible Joint Robots

Han YAO, Wen-Fang XIE and Cang YE

*Abstract-* **In this paper, a composite approach to the adaptive neural network (NN) controller is proposed for a rigid link flexible joint (RLFJ) robot manipulator with unknown nonlinearities. Based on a singular perturbation formulation of robot motion dynamics, the RLFJ robot is described by a reduced-order flexible-joint model. The concept of integral manifold is used to decompose the model into fast and slow dynamics. A composite controller is proposed to deal with the uncertainties in both fast and slow subsystems and to make the link position of the robot follow the desired trajectory. Two NNs are used to approximate two explicit nonlinear functions in two control components to alleviate the symbolic computational burden. By using Lyapunov theorem extension, the stability of the whole system has been proved. The simulation results are presented to show the effectiveness of the approach.**

*Index Terms—* **Neural Networks, Flexible Joint, Composite Approach, Robot Control, Lyapunov Stability**

## 1. INTRODUCTION

Many control strategies have been developed for the control of n-link robot manipulators, such as exact compensation of nonlinearities, robust adaptive algorithms, and variable structure theory [1]- [4]. These control methods share the common feature that the robot dynamics are modeled by the rigid link rigid joint (RLRJ) equations of motion. Unfortunately, experimental evidence indicates that the assumption of perfect rigidity is never satisfied exactly in the real world. The joint flexibility should be taken into account in both modeling and control [5]. From the modeling point of view, a flexible-joint manipulator can be treated as rigid-links interconnected by elastic joints [6]. Normally the joints of robots are made of the harmonic drives that are gear boxes with high-ratio and compact torque-transmission. However, the harmonic drive is plagued with friction and its unique mechanical design and assembly cannot deliver sufficient high stiffness. These characteristics pose challenges to the controller design since the joint flexibility may cause instability or resonant behavior in the system [5]. To deal with these problems, a number of control schemes based on the flexible models have been developed to control flexible-joint robots. These methods include feedback linearization [5, 7, 8, 27], singular perturbation techniques [9, 10], sliding mode [11], and robust adaptive controller approaches [12, 13].

In the category of singular perturbation techniques, the integral manifold scheme in the context of composite control has been investigated in [14, 15, 16]. The approaches start with strategies dealing with the flexible joint robot with known parameters and are extended to the integration of composite control and corrective control methods to cope with flexible joint robots with unknown parameters—the so-called "adaptive integral manifold" approach. Research efforts have been focused on dealing with the effects of un-modeled dynamics and system parameter variations using the reduced order model of the flexible-joint manipulators. The seeming drawbacks of the integral manifold method are its complexity in deriving the expression of the slow control and the computational cost of implementation. These problems are more pronounced in the adaptive integral manifold method [9]. Although the current advances in symbolic software and parallel computing technologies have facilitated the computationally intensive control algorithm, the symbolic computation remains intractable as it hinges on the robot's nonlinear model that is hard to be identified and verified. Moreover, the symbolic computation of symbolic has to be carried out again whenever the RJFL robot is changed.

Recently, many NN controllers with closed-loop stability [17, 18, 19, 20, 21] have been proposed for various control applications. Due to its ability of universal function approximation, NN has been successfully used to design controllers for flexible-joint robots [11, 22, 23, 24]. In [11, 22], NNs are used to approximate the inverse nonlinear function to compensate the flexible nonlinearities. In the above work, off-line training is used to obtain the preliminary weights. Kwan *et al.* [18] proposed a robust NN controller for the motion control of Rigid-Link Electrically Driven (RLED) robots which are modeled as $n^{th}$ order rigid robot dynamics and the $n^{th}$ order electrical dynamics of joint drivers. Hence, the problem of joint flexibility has not been addressed specifically in the controller design. In [24], adaptive neural network controller had been proposed for flexible joint robots using singular perturbation technique. However, the authors only designed an adaptive NN controller for slow subsystem while the uncertainties are ignored in the fast subsystems.

In this paper, an NN-based controller is developed for a RLFJ robot manipulator using the concept of integral manifold and the idea of universal approximation of NN. Based on a singular perturbation formulation of robot motion dynamics, the RLFJ robot is described by a

Manuscript received March 18, 2007; Revised August 20, 2007. This work was supported by Natural Sciences and Engineering Research Council (NSERC).
Han Yao and Wen-Fang Xie are with Dept. of Mechanical & Industrial Engineering, Concordia University, 1455 De Maisonneuve W. Canada (e-mail: han_yao@alcor.concordia.ca wfxie@encs.concordia.ca); Cang Ye is with Department of Applied Science, University of Arkansas at Little Rock 2801 S. University Ave, ETAS 575, Little Rock, AR, USA (e-mail: cxye@ualr.edu).

reduced-order flexible-joint model. The concept of integral manifold is used to decompose the model into fast and slow dynamics. A composite controller is proposed to comprise a slow rigid-based component and additional fast corrective terms so that it can deal with the uncertainties in both fast and slow subsystems. Motivated by the universal approximation of NN, we use two NNs in fast and slow controllers to approximate two explicit nonlinear functions to alleviate the symbolic computational burden. For the flexible-joint based fast controller, a fictitious variable is introduced in the design of an NN controller to provide sufficient damping for the fast dynamics. The NNs' weight matrix update rules are designed using the Lyapunov theorem extension [25] to ensure the unknown RJFL robot's stability. It has been proven that the proposed NN controller guarantees the boundedness of tracking errors and makes the link position of the robot follow the desired trajectory.

This paper is organized as follows. In Section 2, the model of an RLFJ robot manipulator in a singular perturbation form is introduced. Some properties of the robot manipulator and the basic idea of the NN model are presented. In Section 3, the developments of the adaptive NN based controllers for both rigid and flexible joint robots are detailed and the system stability is proved. In Section 4, the numerical implementation of the controller for a two-link flexible-joint manipulator is given and the results are compared with those of the adaptive integral manifold controller. Section 5 concludes the paper.

## 2. SYSTEM MODELS

### 2.1 RLFJ Robot Model

A real industrial manipulator is driven by the actuators in the joints and its behavior cannot be fully captured by a rigid model. Normally an n-link RLFJ robot is modeled by a chain of rigid links interconnected by elastic joints [26]:

$$M(q)\cdot\ddot{q}+V_m(q,\dot{q})\cdot\dot{q}+G(q)+F(\dot{q})+T_L+K\cdot(q-q_f)=0 \quad (1)$$

$$J\cdot\ddot{q}_f+B\cdot\dot{q}_f-K\cdot(q-q_f)=\tau \quad (2)$$

with $q,\dot{q},\ddot{q}\in R^n$ referring to the link position, velocity and acceleration, $q_f,\dot{q}_f,\ddot{q}_f\in R^n$, the motor shaft angle, angular velocity and angular acceleration, respectively, $M(q)\in R^{n\times n}$ the inertia matrix, $V_m(q,\dot{q})\in R^{n\times n}$ the coriolis and centripetal term, $G(q)\in R^n$ the gravity vector, $T_L\in R^n$ the load disturbance, diagonal matrix $K=K^T\in R^{n\times n}$ the stiffness coefficients matrix , diagonal matrix $J=J^T\in R^{n\times n}$ motor inertia diagonal matrix $B=B^T\in R^{n\times n}$ the joint damping term, $\tau$ the control torque and $F(\dot{q})\in R^n$ the friction with the form:

$$F(\dot{q})=[\alpha_0+\alpha_1\cdot e^{-\beta_1|\dot{q}|}+\alpha_2(1-e^{-\beta_2|\dot{q}|})]\cdot\mathrm{sgn}(\dot{q}) \quad (3)$$

where $\alpha_0+\alpha_1$ represents static friction; $\alpha_2$ represents the viscous friction.

*Property 1:* $M(q)$ is symmetric, positive-definite and a nonlinear function of $q$ . It is bounded by

$$m_1\cdot I\le M(q)\le m_2\cdot I \quad (4)$$

with $m_1$ and $m_2$ being known positive constants.

*Property 2:* $V_m(q,\dot{q})$ is bounded by $v_d(q)\|\dot{q}\|$ , with continuous function $v_d(q)$ .

*Property 3:* The matrix $\dot{M}-2V_m$ is skew-symmetric.

*Property 4:* The unknown load disturbance $T_L$ is bounded by a known positive constant $a_d$ .

When taking the joint flexibility into account, one needs to double the state variables in the dynamic model. This causes that the dynamic model of a flexible joint robot is of order four instead of order two for a rigid robot. Thus, the control problem becomes more complicated when the joint flexibility is considered. Since joint stiffness is large compared with other parameters, one assumes

$$K=K_1/\gamma^2 \quad (5)$$

where $\gamma$ is a small parameter representing the inverse of stiffness and $K_1$ is on the order of 1. Suppose that $J$ and $B$ are very small and on the same order of $\gamma$ . The rigid model can be derived from (1) and (2) by assuming no elasticity at the joints (i.e. $\gamma=0$) and is given by:

$$(M(q)+J)\cdot\ddot{q}+(V_m(q,\dot{q})+B)\cdot\dot{q}+G(q)+F(\dot{q})+T_L=\tau \quad (6)$$

### 2.2 Neural Network Model

In control engineering, NN is used to approximate a nonlinear function for modeling and control purposes. In this work, we use a three-layer Multi-Layer Perceptron (MLP) neural network whose input layer, hidden-layer and output layer are denoted by, $X=[x_1\ \ x_2\ \ ...\ \ x_m]^T$ , $a=[a_1\ \ a_2\ \ ...\ \ a_s]^T$ , and $Y=[y_1\ \ y_2\ \ ...\ \ y_n]^T$ , respectively. The second-to-third layer interconnection weight matrix is defined as $W^T=w_{kj}$ , for $j=1,2,...s$ , $k=1,\ 2,...n$ ; and the associated bias vector is defined as ${\theta_w}^T=[\theta_1\ \ \theta_2\ \ ...\ \ \theta_n]^T$ . The first-to-second layer interconnection weight matrix is defined as $V^T=v_{ji}$ , for $i=1,\ 2,...m$ , $j=1,2,...s$ ; and the associated bias vector is defined as ${d_v}^T=[d_1\ \ d_2\ \ ...\ \ d_s]^T$ . The output of NN is computed by:

$$y_k=\sum_{j=1}^{s}\left[w_{kj}\cdot\sigma\left[\sum_{i=1}^{m}v_{ji}\cdot x_i+d_j\right]+\theta_k\right] \quad (7)$$

where $k=1,2,...n$ , $\sigma(\cdot)$ is the activation function. It can also be expressed in the following matrix form

$$Y = W^T \cdot \sigma(V^T \cdot X + d_v{}^T) + \theta_w{}^T \quad (8)$$

The bias $d_j$ can be added to the weight matrix $V^T = v_{ij}$ as the first column and the input vector is set as $X = [x_0 \quad x_1 \quad \ldots \quad x_{m+1}]^T$ with $x_0 \equiv 1$. Handling $\theta_k$ and $W^T = w_{jk}$ in the same way, one has

$$Y = W^T \cdot \sigma(V^T \cdot X). \quad (9)$$

Let $S$ be a compact set of $R^n$. Define $C^n(S)$ as the space such that the map $f(x): S \to R^n$ is continuous. The NN (9) can approximate function $f(x) \in C^n(S)$, $x \in R^n$ as

$$f(x) = W^T \cdot \sigma(V^T \cdot x) + \varepsilon(x) \quad (10)$$

with $\varepsilon(x)$ is a functional restructure error vector and $W^T$ is an ideal constant weight matrix.

In this paper, it is assumed that there exists such a weight matrix that $\|\varepsilon(x)\| \le \varepsilon_N$ with constant $\varepsilon_N > 0$, for all $x \in R^n$, and the norm of the matrix is bounded by a known constant $\|W\| \le W_N$ with $W_N > 0$.

## 3. CONTROL STRATEGY

### 3.1 Control Objective

The control objective is to develop a position tracking controller for an unknown RLFJ robot dynamics (Eq. 1) so that the link position follows a desired trajectory. The tracking error of link position is defined as

$$e(t) = q_d(t) - q(t) \quad (11)$$

where $q_d(t) \in R^n$ is the given desired trajectory which is continuous and its derivatives up to higher order are bounded.

A filtered error is defined as

$$r = \dot{e} + \Lambda \cdot e \quad (12)$$

where $\Lambda = \Lambda^T > 0$.

The elasticity at the joints is large enough such that the system can be decomposed into a "slow" subsystem and a "fast" subsystem. From [6], the control signal $\tau$ for the whole system has the form as

$$\tau = \tau_s + \tau_f \quad (13)$$

where $\tau_s$ is the slow part and $\tau_f$ is the fast part, which is defined as:

$$\tau_f = K_f(\dot{q} - \dot{q}_f). \quad (14)$$

Usually, we choose

$$K_f = K_2/\gamma \quad (15)$$

with $K_2$ on the order of 1.

Define $z$ as the difference between the link and motor position

$$z = q_f - q. \quad (16)$$

The RLFJ robot manipulator dynamics becomes

$$M(q)\cdot\ddot{q} + V_m(q,\dot{q})\cdot\dot{q} + G(q) + F(\dot{q}) + T_L - K\cdot z = 0 \quad (17)$$

$$J\cdot\ddot{q}_f + B\cdot\dot{q}_f + K\cdot z = \tau. \quad (18)$$

Substituting (13), (14) and (16) into (18), one obtains:

$$J\cdot\ddot{z} + (B + K_f)\cdot\dot{z} + K\cdot z = \tau_s - J\cdot\ddot{q} - B\cdot\dot{q}. \quad (19)$$

Define an integral manifold as $h = K\cdot z$ and rewrite (19) as

$$\begin{aligned} &K\cdot J\cdot K^{-1}\cdot\ddot{h} + K\cdot(B + K_f)\cdot K^{-1}\cdot\dot{h} + K\cdot h \\ &= K(\tau_s - J\cdot\ddot{q} - B\cdot\dot{q}) \end{aligned} \quad (20)$$

where $h = h(t,\gamma,q,q_f)$.

Under the assumptions (5) and (15), one obtains

$$\begin{aligned} &\gamma^2\cdot K\cdot J\cdot K^{-1}\cdot\ddot{h} + \gamma\cdot K\cdot(\gamma\cdot B + K_2)\cdot K^{-1}\cdot\dot{h} + K_1\cdot h \\ &= K_1(\tau_s - J\cdot\ddot{q} - B\cdot\dot{q}) \end{aligned} \quad (21)$$

An approximate reduced-order flexible model can be derived by using a power series expansion of the integral manifold $h$ and control $\tau_s$ around $\gamma = 0$. It is found that the slow control component $\tau_s$ is independent of fast control component $\tau_f$. Let us denote:

$$\begin{aligned} h &= h_0 + \gamma h_1 + O(\gamma^2) \\ \tau_s &= \tau_0 + \gamma\,\tau_1 + O(\gamma^2) \end{aligned} \quad (22)$$

where $\tau_0$ is the control input to the rigid model, $\tau_1$ is the corrective torque term for compensating the effects of $\gamma$, vector $h_0$ represents a zero-order approximation of $h$ and $h_1$ represents the first order correction to $h_0$.

Substituting (22) into (21) yields

$$\begin{aligned} &\gamma^2\cdot K\cdot J\cdot K^{-1}\cdot\ddot{h}_0 + O(\gamma^3) + \gamma\cdot K\cdot(\gamma\cdot B + K_2)\cdot K^{-1}\cdot\dot{h}_0 \\ &+ \gamma^2\cdot K\cdot(\gamma B + K_2)\cdot K^{-1}\cdot\dot{h}_1 + O(\gamma^3) + K_1\cdot h_0 + \gamma K_1 h_1 \\ &+ O(\gamma^2) = K_1\cdot\tau_0 + \gamma K_1\cdot\tau_1 + O(\gamma^2) - K_1 J\cdot\ddot{q} - K_1 B\cdot\dot{q}. \end{aligned} \quad (23)$$

By equating terms of the same powers of $\gamma$ on both sides of (23), one obtains

$$\gamma^1 \text{ terms:} \quad K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0 + K_1 h_1 = K_1\cdot\tau_1 \quad (24)$$

$$\gamma^0 \text{ terms:} \quad K_1 h_0 = K_1\cdot\tau_0 - K_1\cdot J\cdot\ddot{q} - K_1 B\cdot\dot{q}. \quad (25)$$

From (24), one may find:

$$h_1 = K_1^{-1}(K_1\cdot\tau_1 - K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0) \quad (26)$$

and (25) is written as :

$$h_0 = K_1^{-1}(K_1\cdot\tau_0 - K_1\cdot J\cdot\ddot{q} - K_1 B\cdot\dot{q}). \quad (27)$$

Equation (26) means that the corrective control $\tau_1$ relates to $h_1$ and rigid control $\tau_0$ relates to $h_0$.

From (21), the integral manifold $h$ becomes:

$$h = (\tau_s - J\cdot\ddot{q} - B\cdot\dot{q}) - \gamma\cdot K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0 + O(\gamma^2) \quad (28)$$

After substitution for $K\cdot z$ in (17) and usage of (28), the system (17) is rewritten as

$$\begin{aligned} &M(q)\cdot\ddot{q} + V_m(q,\dot{q})\cdot\dot{q} + G(q) + F(\dot{q}) + T_L = \tau_s \\ &- J\cdot\ddot{q} - B\cdot\dot{q} - \gamma\cdot K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0 + O(\gamma^2). \end{aligned} \quad (29)$$

The variable $\dot{h}_0$ is "fast" variables; the link variables $q$ and $\dot{q}$ are "slow" variables. Moreover, the rigid model (6) is obtained by setting $\gamma = 0$.

The control task is to design $\tau_0$ and $\tau_1$ such that the link position of robot follows the desired trajectory. In [14], both $\tau_0$ and $\tau_1$ are derived with complicate expressions especially in the adaptive integral manifold method. In control application, NN is usually used as a tool for modeling nonlinear function due to their universal function approximation capability. In order to alleviate the symbolic computational burden in calculating $\tau_0$ and $\tau_1$, two three-layer MLP neural networks are utilized to approximate two complicate nonlinear functions to form the control signals $\tau_0$ and $\tau_1$.

### 3.2 Rigid Joint Case

The control signal $\tau_0$ is designed by considering the rigid joint model (6). Combining the filtered error (12) and system dynamics (1), one may obtain:

$$[M+J]\cdot\dot{r} = F_0 - [V_m + B]\cdot r - \tau_0 + T_L \tag{30}$$

where $F_0$ is a complicated nonlinear function defined as

$$\begin{aligned} F_0 &= M(q)\cdot(\ddot{q}_d + \Lambda\dot{e}) + (V_m + B)\cdot(\dot{q}_d + \Lambda e) \\ &+ G(q) + F(\dot{q}) + J\ddot{q}_d + J\Lambda\dot{e} \end{aligned}. \tag{31}$$

Motivated by the universal approximate capability of NN, we utilize a first-layer-fixed MLP neural network to approximate the nonlinear function $F_0$ (31).

$$F_0 = f_0(x) = W_0^T\cdot\sigma(V^T\cdot x) + \varepsilon_0(x) \tag{32}$$

$$\hat{F}_0 = \hat{f}_0(x) = \hat{W}_0^T\cdot\sigma(V^T\cdot x) \tag{33}$$

where $x = \begin{bmatrix} \ddot{q}_d^T & \dot{q}_d^T & \dot{q}^T & q^T & \mathrm{sgn}(\dot{q})^T & 1\end{bmatrix}^T$, input-layer weight matrix $V^T$ is selected randomly and is not tuned, and $\hat{W}_0^T$ is the estimated output-layer weight matrix $W_0^T$.

Define the weight estimation error as

$$\tilde{W}_0^T = W_0^T - \hat{W}_0^T. \tag{34}$$

The RLRJ controller is designed as

$$\tau_0 = \hat{F}_0 + K_v\cdot r \tag{35}$$

where $r$ is the filtered error and $K_v$ is a gain matrix.

The update rule of the NN is designed as

$$\dot{\hat{W}}_0 = \Gamma\cdot\sigma(V^T\cdot x)\cdot r^T - k\cdot\Gamma\cdot\|r\|\cdot\hat{W}_0 \tag{36}$$

where $\Gamma = \Gamma^T > 0$ and $k > 0$.

The stability of controller is proved in the following theorem.

***Theorem 3.1:*** For an RJRL robot (6), the NN controller (35) and update rule (36) are applied. For a desired trajectory $q_d(t)$, it is assumed that its time derivatives up to the third order are continuous and bounded. The controlled system's filtered error $r(t)$ is bounded and the tracking error $e(t)$ converges to a small neighborhood around zero by appropriately choosing suitable gain matrix $K_v$.

***Proof:*** Define the Lyapunov function as

$$L_1 = \frac{1}{2}r^T\cdot[M+J]\cdot r + \frac{1}{2}tr(\tilde{W}_0^T\cdot\Gamma^{-1}\cdot\tilde{W}_0) \geq 0 \tag{37}$$

where $M$ and $J$ are defined in ( 1), (2), and (4) and $\Gamma = \Gamma^T > 0$.

Differentiating (37) yields

$$\dot{L}_1 = r^T\cdot[M+J]\cdot\dot{r} + \frac{1}{2}r^T\cdot\dot{M}\cdot r + tr(\tilde{W}_0^T\cdot\Gamma^{-1}\cdot\dot{\tilde{W}}_0) \tag{38}$$

Introducing (30), (35) and (36), one has:

$$\begin{aligned} \dot{L}_1 &= -r^T\cdot(K_v + B)\cdot r + r^T\cdot(T_L + \varepsilon_0) \\ &+ k\cdot\|r\|\cdot tr(\tilde{W}_0^T\cdot W_0 - \|\tilde{W}_0^T\|^2) \end{aligned} \tag{39}$$

Since $B$ is very small compared with $K_v$, its influence can be omitted. The minimum eigenvalue of gain matrix $K_v$ is $\lambda_{v\min}$. Thus we have

$$\dot{L}_1 \leq -\|r\|\left[\lambda_{v\min}\cdot\|r\| - (a_d + \varepsilon_N) - k\cdot\frac{W_N^2}{4}\right]$$

where $a_d + \varepsilon_N$ is the upper bound of $T_L + \varepsilon_0$.

Assume that one have

$$\|r\| > \frac{k\cdot W_N^2/4 + (a_d + \varepsilon_N)}{\lambda_{v\min}} \quad \text{or} \tag{40}$$

$$\|\tilde{W}_0\| > W_N/2 + \sqrt{W_N^2/4 + (a_d + \varepsilon_N)/k}\ . \tag{41}$$

One can prove that $\dot{L}_1$ is negative. In Inequality (40), if the minimum eigenvalue $\lambda_{v\min}$ of gain matrix $K_v$ is chosen large enough, the following inequality is held

$$\frac{k\cdot W_N^2/4 + (a_d + \varepsilon_N)}{\lambda_{v\min}} < b_r \tag{42}$$

where $b_r > 0$ represents the radius of a ball inside the compact set $C_r$ of filtered error $r(t)$.

Thus, any trajectory $r(t)$ starting in compact set $C_r = \{r \mid \|r\| \leq b_r\}$ converges within $C_r$ and is bounded. Then tracking error $e(t)$ converges to a small neighborhood around zero. According to the standard Lyapunov theorem extension [25], this demonstrates the UUB (uniformly ultimately bounded) of both $r(t)$ and $\tilde{W}_0$.

Q.E.D.

### 3.3 Flexible Joint Case

Introducing the filtered error (12) into system (29), one has

$$\begin{aligned} M\cdot\dot{r} &= -V_m\cdot r + F_0 - \tau_s + T_L + J\cdot\ddot{q} + B\cdot\dot{q} \\ &+ \gamma\cdot K^{-1}\cdot K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0 + O(\gamma^2) \end{aligned}. \tag{43}$$

The above equation is rewritten as:

$$M\cdot\dot{r}=-V_m\cdot r+F_0-\tau_0-\gamma\cdot\tau_1-O(\gamma^2)+T_L+ \gamma\cdot\left\{K^{-1}\cdot K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0+K_2\cdot K_f^{-1}\left[J\cdot\ddot{q}+B\cdot\dot{q}\right]\right\} \tag{44}$$

The nonlinear function $F_1$ is defined as

$$F_1=K^{-1}\cdot K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0+K_2\cdot K_f^{-1}\left[J\cdot\ddot{q}+B\cdot\dot{q}\right]. \tag{45}$$

The error dynamics is derived as:

$$M\cdot\dot{r}=-V_m\cdot r+F_0-\tau_0-\gamma\cdot\tau_1-O(\gamma^2)+\gamma\cdot F_1+T_L \tag{46}$$

Again, a second first-layer-fixed MLP neural network is used to approximate the nonlinear function $F_1$ (46).

$$F_1=f_1(y)=W_1^T\cdot\sigma(V^T\cdot y)+\varepsilon_1(y) \tag{47}$$

where $y=\begin{bmatrix}\tau_0^T & q^T & \dot{q}^T\,\hat{W}_{1s}^T & 1\end{bmatrix}^T$, $\hat{W}_{1s}$ represents all of the entries of $\hat{W}_1$ stacked into a single column vector.

To implement $F_1$ (45), one needs to compute $\dot{h}_0$. It can be obtained by differentiating (27) and using $\ddot{q}$ from the rigid model (6). Since $h_0$ is a nonlinear function related to rigid control $\tau_0$, $F_1$ is highly complex nonlinear function of $\tau_0$, $q$, $\dot{q}$ and $\hat{W}_1$.

The corrective term is designed as:

$$\tau_1=\hat{F}_1+K_u\cdot\varphi \tag{48}$$

where $\varphi$ is a fictitious variable, which will be designed later. Substituting the control strategy $\tau_0$ (35) and $\tau_1$ (48) into the error dynamics (46), one obtains:

$$M\cdot\dot{r}=-(V_m+K_v)\cdot r-\gamma\cdot K_u\cdot\varphi+\tilde{F}_0+\gamma\cdot\tilde{F}_1 +T_L-O(\gamma^2) \tag{49}$$

Design a fictitious variable as

$$\dot{\varphi}=\tau_0-K\cdot z\,. \tag{50}$$

Using (22) and (28), one has

$$\dot{\varphi}=-\gamma\cdot\tau_1-O(\gamma^2)+\gamma\cdot[K_1^{-1}\cdot K\cdot K_2\cdot K^{-1}\cdot\dot{h}_0 +K_2\cdot K_f^{-1}(J\cdot\ddot{q}+B\dot{q})]. \tag{51}$$

Using (45), one has

$$\dot{\varphi}=\gamma\cdot F_1-\gamma\cdot\tau_1-O(\gamma^2)\,. \tag{52}$$

Similarly, with the control strategy (48), equation (52) becomes

$$\dot{\varphi}=\gamma\cdot\tilde{F}_1-\gamma\cdot K_u\cdot\varphi-O(\gamma^2)\,. \tag{53}$$

### 3.3 Overall Controller

Two different NN based controllers have been designed—one is the first slow part $\tau_0$ based on a rigid NN approximating $F_0$ function and the other is the second slow part $\tau_1$ based on the corrective NN approximating $F_1$ function. The composite control scheme is shown as:

$$\tau=\tau_s+\tau_f=\tau_0+\gamma\cdot\tau_1+O(\gamma^2)-K_f\dot{z}\,. \tag{54}$$

With (35) and (48), the overall control scheme is derived as

$$\tau=(\hat{F}_0+K_v\cdot r)+\gamma\cdot(\hat{F}_1+K_u\cdot\varphi)-K_f\dot{z}\,. \tag{55}$$

Choose the update rule for those weight matrices respectively as

$$\dot{\hat{W}}_0=\Gamma_0\cdot\sigma_0(V^T\cdot x)\cdot r^T-k\cdot\Gamma_0\cdot\|\xi\|\cdot\hat{W}_0 \tag{56}$$

$$\dot{\hat{W}}_1=\Gamma_1\cdot\sigma_1(V^T\cdot y)\cdot(\varphi^T+r^T)-k\cdot\Gamma_1\cdot\|\xi\|\cdot\hat{W}_1 \tag{57}$$

where $\xi=\begin{bmatrix}r^T & \varphi^T\end{bmatrix}^T$, $\Gamma=\Gamma^T>0$ and $k>0$, and the weight matrix errors are derived as:

$$\dot{\tilde{W}}_0=\dot{\hat{W}}_0=-\Gamma_0\cdot\sigma_0(V^T\cdot x)\cdot r^T+k\cdot\Gamma_0\cdot\|\xi\|\cdot\hat{W}_0 \tag{58}$$

$$\dot{\tilde{W}}_1=-\dot{\hat{W}}_1=-\Gamma_1\cdot\sigma_1(V^T\cdot y)\cdot(\varphi^T+r^T)+k\cdot\Gamma_1\cdot\|\xi\|\cdot\hat{W}_1\,. \tag{59}$$

*Remark 1:* The NN controller design algorithm was motivated by the integral manifold method. In integral manifold procedure, an iterative algorithm has been proposed to solve the manifold to which the slow dynamics converges by using Taylor series expansion around the zero of the inverse of stiffness. The procedure becomes very tedious and time-consuming. For different robots with different nonlinear models, the procedure has to be repeated. While the proposed NN controller does not need such computations and are applicable to different robots with different parameters due to the on-line tuning.

*Remark 2:* No off-line weight tuning is needed. The initial estimation values of the weight matrix $\hat{W}_0^T$ and $\hat{W}_1^T$ are set to zero. At the beginning, the controller becomes a PD controller. The control scheme does not guarantee that $\hat{W}_0^T$ and $\hat{W}_1^T$ converge to the true values of $W_0^T$ and $W_1^T$. From the above theorem, one may claim that the boundedness of $W_0^T$, $W_1^T$ and $r(t)$ are guaranteed. Thus, the tracking error $e(t)$ is guaranteed to approach to zero.

***Theorem 3.2:*** For an RJFL robot (1), the NN controller (55) and update rules (58) and (59) are applied. For a desired trajectory $q_d(t)$, it is assumed that its time derivatives up to the third order are continuous and bounded. The controlled system's filtered error $r(t)$ and fictitious variable $\varphi(t)$ are bounded and the tracking error $e(t)$ converges to a small neighborhood around zero by appropriately choosing suitable gain matrices $K_f$, $K_v$, and $K_u$.

***Proof :*** Define the Lyapunov function as

$$L_2=\frac{1}{2}r^T\cdot M\cdot r+\frac{\gamma}{2}\varphi^T\cdot\varphi+\frac{1}{2}tr(\tilde{W}_0^T\cdot\Gamma_0^{-1}\cdot\tilde{W}_0) +\frac{\gamma}{2}tr(\tilde{W}_1^T\cdot\Gamma_1^{-1}\cdot\tilde{W}_1)\geq 0 \tag{60}$$

Differentiating the above function yields

$$\dot{L}_2=r^T\cdot M\cdot\dot{r}+\frac{1}{2}r^T\cdot\dot{M}\cdot r+\gamma\cdot\varphi^T\cdot\dot{\varphi}+tr(\tilde{W}_0^T\cdot\Gamma_0^{-1}\cdot\dot{\tilde{W}}_0) +\gamma\cdot tr(\tilde{W}_1^T\cdot\Gamma_1^{-1}\cdot\dot{\tilde{W}}_1)\,. \tag{61}$$

Introducing (49), (53), (58) and (59), one obtains:

$$\dot{L}_2 = -r^T \cdot K_v \cdot r - \gamma \cdot r^T \cdot K_u \cdot \varphi + r^T \cdot (\varepsilon_0 + \varepsilon_1 + T_L)$$
$$-\gamma^2 \cdot \varphi^T \cdot K_u \cdot \varphi + \gamma \cdot \varphi^T \cdot \varepsilon_1$$
$$+ tr(\tilde{W}_0^T \cdot k \cdot \|\xi\| \cdot \hat{W}_0) + \gamma \cdot tr(\tilde{W}_1^T \cdot k \cdot \|\xi\| \cdot \hat{W}_1) .$$

Defining $Q = \begin{bmatrix} K_v & \gamma \cdot K_u \\ 0 & \gamma^2 \cdot K_u \end{bmatrix}$ and $W = diag\{W_0 \quad \gamma \cdot W_1\}, \tilde{W} = W - \hat{W}$ , one acquires:

$$\dot{L}_2 = -\xi^T \cdot Q \cdot \xi + r^T \cdot (\varepsilon_0 + \varepsilon_1 + T_L) + \gamma \cdot \varphi^T \cdot \varepsilon_1$$
$$+ k \cdot \|\xi\| \cdot tr(\tilde{W}^T \cdot W - \tilde{W}^T \cdot \tilde{W}) . \quad (62)$$

Thus one has

$$\dot{L}_2 \le -\lambda \cdot \|\xi\|^2 + \|\xi\| \cdot \varepsilon_N + k \cdot \|\xi\| \cdot tr(\|\tilde{W}\|_F \cdot W_N - \|\tilde{W}\|_F{}^2) \quad (63)$$

where $\lambda_{Q\min}$ is the minimum eigenvalue of $Q$ and $\varepsilon_N = \max(|\varepsilon_0 + \varepsilon_1 + T_L| \quad \gamma^2 \cdot |\varepsilon_1|)$, $W_N$ is the bound of the ideal weight matrix $W$ , and

$$tr(\|\tilde{W}\|_F \cdot W_N - \|\tilde{W}\|_F{}^2) \le \|\tilde{W}\|_F \cdot W_N - \|\tilde{W}\|_F{}^2 .$$

Inequality (63) becomes

$$\dot{L}_2 \le -\|\xi\| \cdot [\lambda_{Q\min} \cdot \|\xi\| - \varepsilon_N - k \cdot (\|\tilde{W}\|_F \cdot W_N - \|\tilde{W}\|_F{}^2)] ,$$

and one has

$$\lambda_{Q\min} \cdot \|\xi\| - \varepsilon_N - k \cdot (\|\tilde{W}\|_F \cdot W_N - \|\tilde{W}\|_F{}^2)$$
$$= \lambda_{Q\min} \cdot \|\xi\| - \varepsilon_N - \frac{k \cdot W_N{}^2}{4} + k \cdot (\|\tilde{W}\|_F - \frac{W_N}{2})^2 .$$

Suppose one have

$$\|\xi\| > \frac{k \cdot W_N{}^2 / 4 + \varepsilon_N}{\lambda_{Q\min}} \quad (64)$$

or

$$\|\tilde{W}\|_F > W_N / 2 + \sqrt{W_N{}^2 / 4 + \varepsilon_N / k} . \quad (65)$$

One can prove that $\dot{L}_2$ is negative. Inequality (41) shows that if control gains $K_f$ , $K_v$ , and $K_u$ are chosen large enough, the following inequality is held

$$\frac{k \cdot W_N{}^2 / 4 + \varepsilon_N}{\lambda_{Q\min}} < b_\xi$$

where $b_\xi > 0$ represents the radius of a ball inside the compact set $C_\xi$ of filtered error $\xi(t)$ .

Thus, any trajectory $\xi(t)$ starting in compact set $C_\xi = \{\xi | \|\xi\| \le b_\xi\}$ converges within $C_\xi$ and is bounded. According to the standard Lyapunov theorem extension [25], it demonstrates the UUB (uniformly ultimately bounded) of both $\xi(t)$ and $\tilde{W}$ .

Q.E.D.

The overall NN controller structure is shown in Fig. 1. The control algorithm is summarized as the following steps.

**Step i** With (11) and (12), the filtered error $r(t)$ is obtained.

**Step ii** Following the control strategy (35), the control signal $\tau_0(t)$ is calculated.

**Step iii** The fictitious variable $\varphi(t)$ is obtained by (50).

**Step iv** Following the control strategy (48), control signal $\tau_1(t)$ is computed.

**Step v** The overall control signal $\tau(t)$ is calculated using the control scheme represented by (13) and (14).

## 4. SIMULATION

The effectiveness of the proposed control scheme is demonstrated on a two-link manipulator [26], which can be described in the form of (1) and (2)

$$M(q) = \begin{bmatrix} a + b \cdot \cos(q_2) & c + \frac{b}{2} \cdot \cos(q_2) \\ c + \frac{b}{2} \cdot \cos(q_2) & c \end{bmatrix}$$

$$V_m(q, \dot{q}) = \begin{bmatrix} -\frac{b}{2} \cdot \dot{q}_2 \cdot \sin(q_2) & -\frac{b}{2} \cdot (\dot{q}_1 + \dot{q}_2) \cdot \sin(q_2) \\ \frac{b}{2} \cdot \dot{q}_1 \cdot \sin(q_2) & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} d \cdot \cos(q_1) + e \cdot \cos(q_2) \\ e \cdot \cos(q_1 + q_2) \end{bmatrix}$$

$$F(\dot{q}) = \begin{bmatrix} \{35 + 1.1 \cdot e^{-50|\dot{q}_1|} + 0.9(1 - e^{-65|\dot{q}_1|})\} \cdot \mathrm{sgn}(\dot{q}_1) \\ \{38 + 1.0 \cdot e^{-55|\dot{q}_2|} + 0.95(1 - e^{-60|\dot{q}_2|})\} \cdot \mathrm{sgn}(\dot{q}_2) \end{bmatrix}$$

$a = l_2{}^2 \cdot m_2 + l_1{}^2 \cdot (m_1 + m_2)$ , $b = 2 \cdot l_1 \cdot l_2 \cdot m_2$ , $c = l_2{}^2 \cdot m_2$
$d = (m_1 + m_2) \cdot l_1 \cdot g_0$ , $e = m_2 \cdot l_2 \cdot g_0$ .

The parameter values are shown in Table 1. Two input reference signal are chosen as desired two joints positions: $q_{1d} = 2 \cdot \sin(0.1\pi \cdot t)$ and $q_{2d} = 3 \cdot \sin(0.1\pi \cdot t)$ . The control objective is to make the flexible-joint robot joint angle $q = [q_1 \quad q_2]^T$ follow the given desired joint trajectory $q_d = [q_{1d} \quad q_{2d}]^T$ . The gains are selected as: $\Lambda = [20 \quad 1]^T$ , $K_V = diag\{50 \quad 50\}$ , $K_u = diag\{5 \quad 5\}$ , $K_f = diag\{3 \quad 3\}$ , $\Gamma_0 = diag\{10 \quad 10\}$ , $\Gamma_1 = diag\{20 \quad 20\}$ , and $k = 0.1$ .The system responses under the control of the proposed NN-based controller are shown in Fig. 2.

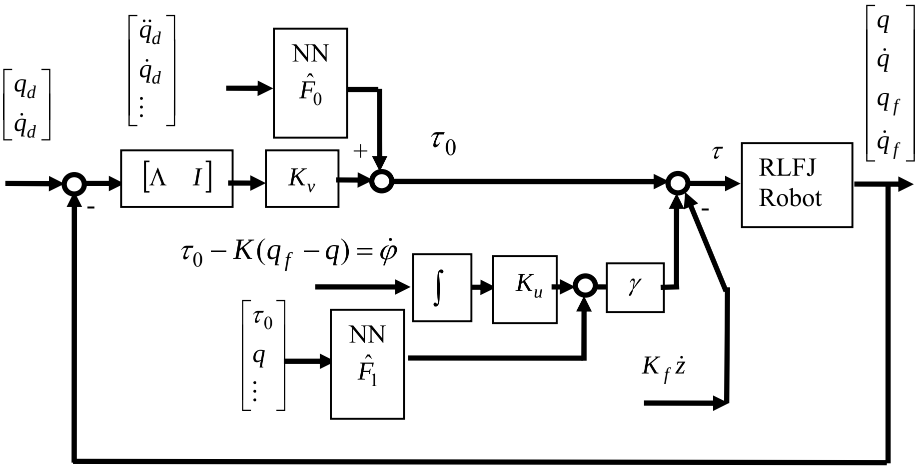
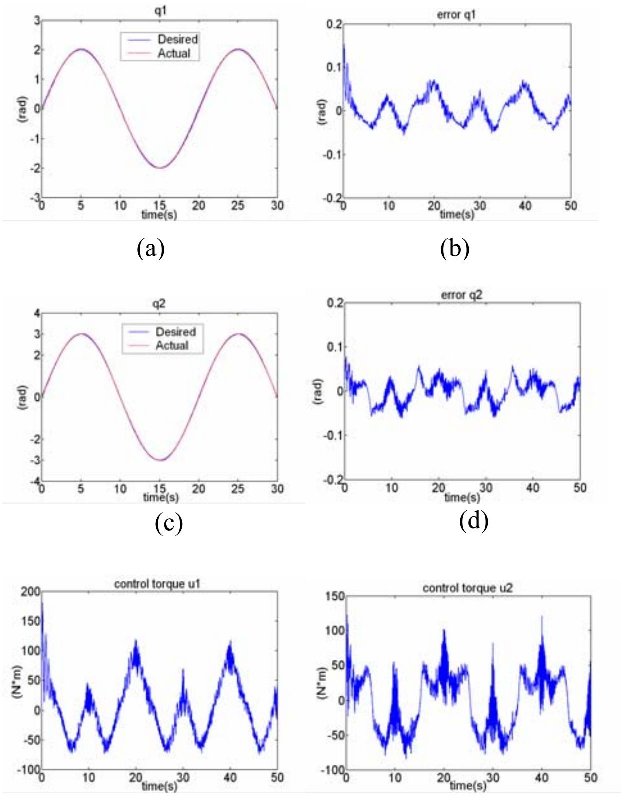



Fig. 1 NN Controller structure

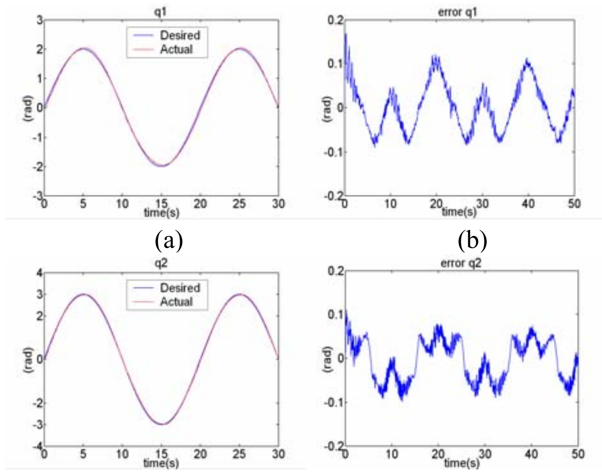



(a) (b) (c) (d) (e) (f)

Fig. 2 Performance of the NN controller with $K = diag\{100 \quad 100\}$ (a) Actual (dashed line) and desired (solid) joint $q_1$ angles (b) Error $q_1 - q_{1d}$ (c) Actual (dashed line) and desired (solid) joint $q_2$ angles. (d) Error $q_2 - q_{2d}$, (e) Bounded control torque of joint $u_1$ (f) Bounded control torque $u_2$

**Table 1 System Parameters Values**

| System Parameters | Values |
|---|---|
| Link 1 length (m) | 1 |
| Link 2 length (m) | 1 |
| Mass of link 1 (kg) | 0.8 |
| Mass of link 2(kg) | 2.3 |
| Gravity acceleration(m/s$^2$) | 9.8 |




(a) (b) (c) (d)

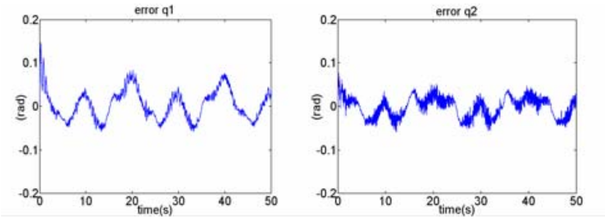
Fig. 3 Performance of an adaptive integral manifold scheme [16] (a) Actual (dashed line) and desired (solid line) joint $q_1$ angles (b) Error $q_1 - q_{1d}$ (c) Actual (dashed line) and desired (solid line) joint $q_2$ angles (d) Error $q_2 - q_{2d}$

For a comparison of performance, an adaptive manifold scheme [16] has been implemented by using the same control parameters. The simulation results are shown in Fig. 3. Please note that an unknown friction term is added in the robot model which makes the control problem more challenging than that in [16].

The simulation results show that the proposed NN-based controller outperforms the adaptive manifold approach with simpler implementation. The NN is tuned on-line without any preliminary off-line training.

### 4.2 Robustness Test

In order to test the robustness of the controller, one changes the system parameters to $l_1 = 1.2m, l_2 = 0.8m$, $m_1 = 1kg$, and $m_2 = 2kg$ and then apply the same NN-based controller to the system. The system responses are shown in Fig. 4.



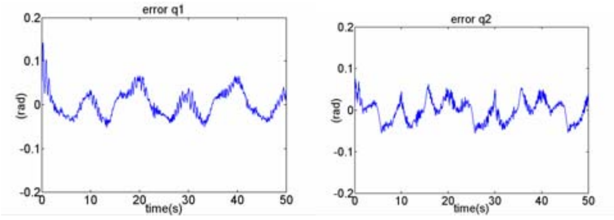

(a) (b)

Fig. 4 Performance of NN controller with 20% change of system parameters (a) Error $q_1 - q_{1d}$ and (b) Error $q_2 - q_{2d}$

From the above results, one observes that the proposed NN controlled system gives a good response when the system parameters are changed within 20% percent range. The test results show that the NN controller owns the ability to deal with the system uncertainties.

### 4.3 Stiffness Parameter

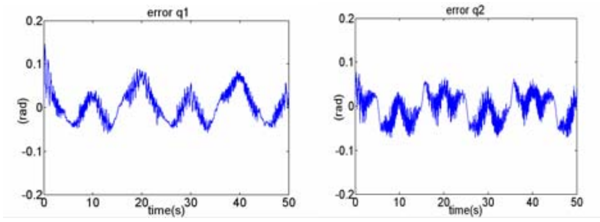
Further simulation runs have been carried out to test the effect of stiffness variation to the controller. The same controller is run by using two stiffness parameters: $K = diag\{300 \quad 300\}$ and $K = diag\{30 \quad 30\}$. The system responses are shown in Figs. 5 and 6. The results demonstrate that the proposed controller can deal with relative large range stiffness change.




(a) (b)

Fig. 5 Performance of NN controller with stiffness parameters $K = diag\{300 \quad 300\}$ (a) Error $q_1 - q_{1d}$ (b) Error $q_2 - q_{2d}$

Figs. 2-6 show the simulation results of applying the NN controller to the RLFJ system for tracking desired signal. One can see that a very good tracking performance is obtained. The NN controller can indeed improve the tracking performance without resorting to high-gain feedback. In addition, we do not even need to know the explicit parameters of a system. Moreover, the NN controller can be implemented in a wide stiffness parameter range. This is a significant advantage since the NN controller can be applied to any type of flexible or rigid robots with little modification to gain parameters.




(a) (b)

Fig. 6 Performance of NN controller with stiffness parameters $K = diag\{30 \quad 30\}$

(a) Error $q_1 - q_{1d}$ (b) Error $q_2 - q_{2d}$

## 5. Conclusion

In this paper, an adaptive NN controller is designed for a rigid link flexible joint (RLFJ) robot manipulator with unknown nonlinearities by using a composite control approach. Two NNs are used to approximate two complicated unknown nonlinear functions in both fast and slow control components. No off-line training is required for NNs. The control algorithm and weight matrix update rule are derived from Lyapunov theorem extension. The stability and boundedness of the tracking error of this unknown RLFJ robot manipulator have been proved. Simulation results show that the proposed NN controller outperforms the adaptive composite control method and can be applicable to unknown flexible robots with a larger range of stiffness. Future work includes the development of on-line self-constructing NN-based controller for unknown RLFJ robot manipulators.

## Acknowledgement

The authors would like to thank the comments provided by the anonymous reviewers and editor, which help the authors improve this paper significantly.

## References

[1] R. Horowitz and M. Tomizuka, "An adaptive control scheme for mechanical manipulators – compensation of nonlinearity and decoupling control," *ASME Journal of Dynamic Systems Measurement and Control*, Vol. 108, no. 2, pp.127-135, 1986.

[2] S. Nicosia and P. Tomei, "Robot Control by using only Joint Position Measurements," *IEEE Transactions on Automatic Control*, Vol. 35, no. 9, pp.1058-1061, 1990.

[3] J. J. Craig, P. Hsu, and S. S. Sastry, "Adaptive Control of Robot Manipulators," *International Journal of Robotics and Research,* Vol.6, No. 3, pp.16-28, 1987.

[4] J. J. E. Slotine and W. Li "Adaptive Manipulator Control: A Case Study," *IEEE Transactions on Automatic Control*, Vol. 33, no. 11, pp. 995-1003, Nov. 1988.

[5] M.W. Spong, "Modeling and Control of Elastic Joint Robots", *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 109, pp.310-319, 1987.

[6] F. Ghorbel, J. H. Hung, and M. W. Spong, "Adaptive Control of Flexible-Joint Manipulators," *IEEE Control System Magazine*, Vol. 9, No. 7, pp. 9-13, 1989.

[7] A. De Luca, "Decoupling and Feedback Linearization of Robots with Mixed Rigid/Elasitic Joints", *International Journal of Robust Nonlinear Control*, Vol.8, pp.965-977, 1998.

[8] S.S. Ge, T.H. Lee and E.G. Tan, "`Adaptive Neural Network Control of Flexible Joint Robots based on Feedback Linearization," *International Journal of Systems Science*, Vol.29, no.6, pp.623-635, 1998.

[9] M. W. Spong "Adaptive Control of Flexible Joint Robots," *Systems and Controls Letters*, Vol. 13, no. 1, pp. 15-21, 1989.

[10] D.M. Dawson, Z. Qu, and M.M. Bridges, "Hybrid Adaptive Control for Tracking of Rigid-Link Flexible-Joint Robots," *IEE Proceedings Control Theory and Applications,* Vol. 140, No. 3, pp. 155-159, 1993.

[11] O. Aboulshamat and P. Sicard "Position Control of a Flexible Joint with Friction Using Neural Network Feedforward Inverse Models," *Proceeding of Canadian Conference on Electrical and Computer Engineering*, Toronto, Canada, pp. 283-388, 2001.

[12] F. Abdollahi, H. A. Talebi, and R.V. Patel, "State Estimation for Flexible-Joint Manipulators using Stable Neural Networks," *Proceeding of \ IEEE International Symposium on Computation Intelligence in Robotics and Automation*, Kobe, Japan, pp. 25-29, July 2003.

[13] W.H. Zhu and J. De Schutter, "Adaptive Control of Mixed Rigid/Flexible Joint Robot Manipulators Based on Virtual Decomposition", *IEEE Transactions on Robotics and Automation*, Vol. 15, No.2, pp 310-317, April 1999.

[14] M. Moallem, K. Khorasani, and R. V. Patel, "An Integral Manifold Approach for Tip-Position Tracking of Flexible Multi-Link Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 823-837, 1997.

[15] R. A. Al-Ashoor, R. V. Patel, and K. Khorasani, "Robust Adaptive Controller Design and Stability Analysis for Flexible-Joint Manipulators," *IEEE transactions on Systems, Man and Cybernectics*, Vol. 23, no. 2, pp.589-602, 1993.

[16] F. Ghorbel and M. W. Spong, "Adaptive Integral Manifold Control of Flexible Joint Robot Manipulators," *Proceedings of IEEE International Conference on Robotics and Automation*, Nice, France, pp.707-714, 1992.

[17] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural Net Robot Controller with Guaranteed Tracking Performance," *IEEE Transactions on Neural Networks*, Vol. 6, no. 3, pp. 703-715, 1995.

[18] C. Kwan, F. L. Lewis, and D. M. Dawson, "Robust Neural-Network Control of Rigid-Link Electrically Driven Robots," *IEEE Transactions on Neural Networks*, Vol. 9, no. 4, pp.581-588, 1998.

[19] W. Gao, R. R. Selmic, and F. L. Lewis, "Robust Composite Saturation Compensation for a Single Flexible Link Using Neural Networks," *Proc. 2005 IEEE International Symposium on Intelligent Control,* Limassol, Cyprus, pp. 280-285, June 2005.

[20] C. M. Kwan, F. L. Lewis, and Y. H. Kim, "Robust Neural Network Control of Rigid Link Flexible-joint Robots," *Asian Journal of Control*, Vol. 1, no. 3, pp. 188-197, 1999.

[21] S. S. Ge and C. Wang, "Direct Adaptive NN Control of a Class of Nonlinear Systems," *IEEE Transactions on Neural Networks*, Vol. 13, No. 1, pp.214-221, 2002.

[22] V. Zeman, R.V. Patel, and K. Khorasani, "Control of a Flexible-Joint Robot Using Neural Networks," *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 4, pp.453-462, 1997

[23] W. Chatlatanagulchai and P. H. Mechl, "Motion Control of Two-Link Flexible-Joint Robot, Using Backstepping, Neural Networks and Indirect Method," *Proceeding of the IEEE International Conference on Control Applications*, Toronto, Canada, pp. 601-605, 2005.

[24] S.S. Ge and I. Postlethwaite, "Adaptive neural network controller design for flexible joint robots using singular perturbation technique," *Transactions of the Institute of Measurement and Control*, Vol.17, No.3, pp.120-131, 1995.

[25] K. S. Narendra and A. M. Annaswamy, "A New Adaptive Law for Robust Adaptation without Persistent Excitation", *IEEE transactions on Automatic Control*, Vol. 32, No.2, pp.134-145, 1987.

[26] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators,* New York, New York: Macmillan Publishing Company, 1993.

[27] S.S. Ge, T.H. Lee and E.G. Tan, "`Adaptive neural network control of flexible joint robots based on feedback linearization," *International Journal of Systems Science*, Vol.29, no.6, pp.623-635, 1998.


**Yao Han** received his Master degree from Concordia University, Montreal, Canada in 2006. He was an assistant engineer in China Shipbuilding Industry Corporation (CSIC), China from 2001 to 2003. He received Bachelor from Beijing University of Aeronautics and Astronautics in 2001. His research interests include Neural Network, robotic control, adaptive control and integral manifold.

**Wen-Fang Xie** is an assistant professor with Department of Mechanical and Industrial Engineering at Concordia University, Montreal, Canada. She was in Industrial Research Fellowship holder from Natural Sciences and Engineering Research Council of Canada (NSERC) and served as a senior research engineer in InCoreTec, Inc. Canada before she joined Concordia University in 2003. She received her Ph.D from The Hong Kong Polytechnic University in 1999 and Master Degree from Beijing University of Aeronautics and Astronautics in 1991. Her research interests include nonlinear control in Mechatronics, Neural Network, system identification and advanced process control.

**Cang Ye** received B.E. and M.E. degree from the University of Science and Technology of China, P. R. China in 1988 and 1991, respectively. He received his Ph.D. degree from the University of Hong Kong in 1999.

He is currently an Assistant Professor in Department of Applied Science, University of Arkansas at Little Rock, AR. He was a research investigator at University of Michigan, Ann Arbor, MI from 2003 to 2005. His research interests are in Robotics, Neural Network, and Reinforcement Learning.

Dr. Ye is a senior member of IEEE and a member of Technical Committee on Robotics and Intelligent Sensing, IEEE Systems, Man, and Cybernetics Society.