Robust Neural-fuzzy-network Control for Rigid-link Electrically Driven Robot Manipulator

Rong-Jong Wai, Member, IEEE, Chun-Yen Tu, and Po-Chen Chen

1,2,3 Department of Electrical Engineering, Yuan Ze University, Chung Li 32026, Taiwan, R.O.C.
1 e-mail: rjwai@saturn.yzu.edu.tw 2 e-mail: andy.due@delta.com.tw 3 e-mail: imbola@ms93.url.com.tw

Abstract—This study addresses the design and analysis of an intelligent control system for an n-link robot manipulator to achieve the high-precision position tracking. According to the concepts of mechanical geometry and motion dynamics, the dynamic model of an n-link robot manipulator including actuator dynamics is introduced initially. However, it is difficult to design a suitable model-based control scheme due to the uncertainties in practical applications, such as friction forces, external disturbances and parameter variations. In order to deal with the mentioned difficulties, a robust neural-fuzzy-network control (RFNC) system is investigated to control the joint position control of an n-link robot manipulator for periodic motion. In this control scheme, a four-layer neural-fuzzy-network (NFN) is utilized for the major control role, and the adaptive tuning laws of network parameters are derived in the sense of projection algorithm and Lyapunov stability theorem to ensure the network convergence as well as stable control performance. The merits of this model-free control scheme are that not only the stable position tracking performance can be guaranteed, but also no prior system information and auxiliary control design are required in the control process. In addition, numerical simulations of a two-link robot manipulator actuated by DC servomotors are provided to verify the effectiveness and robustness of the proposed RFNC methodology.

I. INTRODUCTION

In the past decade, the applications of intelligent control techniques (fuzzy control or neural-network control) to the motion control of robotic manipulators have received considerable attention [1], [2]. In general, robotic manipulators have to face various uncertainties in their dynamics, such as payload parameter, friction and disturbance. It is difficult to establish an appropriate mathematical model for the design of a model-based control system. Thus, the general claim of these intelligent control approaches is that they can attenuate the effects of structured parametric uncertainty and unstructured disturbance by using their powerful learning ability without detailed knowledge of the controlled plant in the design processes. For the most part of the robot manipulator control in the published literature [1], [2], actuator dynamics are typically excluded from the robot dynamic behavior to simplify the control design. However, actuator dynamics perform an important part of the complete robotic dynamics, especially in the factors of high-velocity moment, highly varying loads, friction and actuator saturation [3]–[5]. Thus, there exist some interactions between robot and actuator dynamics that can not be neglected.

Recently, the concept of incorporating fuzzy logic into a neural network has grown into a popular research topic [6]–[9]. The integrated fuzzy-neural-network system possesses the merits of both fuzzy systems [10] (e.g., humanlike IF-THEN rules thinking and ease of incorporating expert knowledge) and neural networks [11] (e.g., learning and optimization abilities, and connectionist structures). In this way, one can bring the low-level learning and computational power of neural networks into fuzzy systems and also high-level, humanlike IF-THEN rule thinking and reasoning of fuzzy systems into neural networks. The aim of this study is to design an intelligent control scheme for the position control of an n-link robot manipulator including actuator dynamics without the requirement of system information and constraint conditions and the compensation of auxiliary control design.

This study is organized as follows. Section II presents the dynamic model of an n-link robot manipulator including actuator dynamics briefly [12], [13], and exhibits the design process of a robust feedback linearization control (RFC) system. To relax the requirement of system parameters and uncertainty information, a robust neural-fuzzy-network control (RFNC) system is investigated to control an n-link robot manipulator for periodic motion in Sec. III. The design procedures of the proposed RFNC system are described in detail. The adaptive learning rules in the RFNC system are derived in the sense of projection algorithm [10], and Lyapunov stability theorem [14], so that the network convergence and system-tracking stability can be guaranteed in the closed-loop control system. Numerical simulations of a two-link robot manipulator under the possible occurrence of uncertainties are provided to demonstrate the robust control performance of the proposed RFNC system in Sec. IV. Conclusions are drawn in Sec. V.

II. ROBUST FEEDBACK LINEARIZATION CONTROL

In general, the dynamic model of armature-controlled DC servomotors on an n-link robot manipulator can be expressed in the following form [5], [13]:

\[ \tau_e = K_i i_e \]  
\[ \tau_e = J_e \ddot{\theta}_e + B_e \dot{\theta}_e + \tau_n \]  
\[ v_e = R_m i_e + L_m \dot{i}_e + K_m \dot{\theta}_e \]

where \( \tau_e \in \mathbb{R}^n \) is the vector of electromagnetic torque; \( K_i \in \mathbb{R}^{n \times n} \) is the diagonal matrix of motor torque constants.
\( i_e \in \mathbb{R}^n \) is the vector of armature currents; \( J_\alpha \in \mathbb{R}^{n \times n} \) is the diagonal matrix of the moment inertia; \( B_\alpha \in \mathbb{R}^{n \times n} \) is the diagonal matrix of torsional damping coefficients; \( \theta_e, \theta, \dot{\theta} \in \mathbb{R}^n \) denote the vectors of motor shaft positions, velocities, and accelerations, respectively; \( \tau_e \in \mathbb{R}^n \) is the vector of external disturbance.

\[
\tau_e - \vec{\tau} = (11)
\]

where \( \tau_v \in \mathbb{R}^n \) represents the vector of load torque; \( J_v \in \mathbb{R}^{n \times n} \) is the diagonal matrix of armature inductance; \( K_v \in \mathbb{R}^{n \times n} \) is the diagonal matrix of armature resistance; \( L_\alpha \in \mathbb{R}^{n \times n} \) is the diagonal matrix of armature inductance; \( K_z \in \mathbb{R}^{n \times n} \) is the diagonal matrix of back electromotive force (EMF) coefficients. In order to apply the DC servomotors for actuating an \( n \)-link robot manipulator, a relationship between the joint position \( q \) and the motor-shaft position \( \theta_e \) can be represented as follows:

\[
g_e = \frac{\theta_e - \tau}{q} = (4)
\]

where \( g_e \in \mathbb{R}^{n \times n} \) is a diagonal positive-definite matrix of the gear ratios for the \( n \) joints; \( \tau \in \mathbb{R}^n \) is the vector of control torque developed at the joint side; \( q \in \mathbb{R}^n \) is the vector of joint positions. According to (1), (2) and (4), the vector of armature input voltages in (3) could be rewritten as

\[
v_e = L_v \dot{\theta} = R_v i_e + L_v J_v \ddot{\theta} + (R_v J_v + L_v) \dddot{\theta} + (R_v B_v + K_v) \dddot{\theta} \tag{5}
\]

where \( L_v = L_v (g_v K_v)^{-1} \), \( R_v = R_v (g_v K_v)^{-1} \), \( J_v = g_v J_v \), \( B_v = g_v B_v \) and \( K_v = g_v K_v \).

After the description of actuator dynamics, the mechanical behavior of an \( n \)-link robot manipulator is considered and expressed in the following Lagrange form [13]:

\[
M(q) \dddot{q} + C(q, \dot{q}) \dddot{q} + G(q) + N = \tau \tag{6}
\]

where \( \dot{q}, \dddot{q} \in \mathbb{R}^n \) are the joint velocity and acceleration vectors, respectively; \( M(q) \in \mathbb{R}^{n \times n} \) denotes the inertia matrix; \( C(q, \dot{q}) \in \mathbb{R}^{n \times n} \) expresses the matrix of centripetal and Coriolis forces; \( G(q) \in \mathbb{R}^{n \times 1} \) is the gravity vector; \( N \in \mathbb{R}^{n \times 1} \) represents the vector of external disturbance \( t \), friction term \( f(\dot{q}) \) and unmodelled dynamics. Substituting (6) into (5), the governed equation of an \( n \)-link robot manipulator including actuator dynamics can be obtained as [5]

\[
M^{\ddot{q}} + D(q, \dot{q}, \ddot{q}) + d = U \tag{7}
\]

where \( U \in \mathbb{R}^{n \times 1} \) represents the control effort vector, i.e. armature input voltages; \( M(q) = M_\alpha + M_e(q) \), and

\[
M = L_v [M_\alpha + J_v] \tag{8}
\]

\[
D(q, \dot{q}, \ddot{q}) = \{L_v [M_e(q, \dot{q}) + C(q, \dot{q}) + B_v] + R_v [M(q) + J_v] \dddot{\theta} + [L_v C(q, \dot{q}, \dddot{\theta}q
\]

+ R_v C(q, \dot{q}) \dddot{\theta} + R_v B_v \dddot{\theta} + K_v \dddot{\theta} \}

\[
+ L_v G(q) + R_v G(q) \}
\]

\[
d = L_v M_e(q) \dddot{\theta} + L_v N + R_v N \tag{10}
\]

In (7), the term \( d \) is taken as the lumped dynamic uncertainty. Here the bound of the lumped dynamic uncertainty is assumed to be given, i.e. \( \|[d] < d \_ \) in which \( \| \) denotes the Euclidean norm and \( d \_ \) is a given positive constant. The control problem is to find a control law so that the joint position vector \( q(t) \) can track a specific command vector \( q_\_ \), where \( q(t) \in \mathbb{R}^n \). Define a tracking error vector \( \vec{q}(t) \in \mathbb{R}^n \) and an error function vector \( e(t) \in \mathbb{R}^n \) as

\[
\vec{q}(t) = q(t) - q_\_ \tag{11}
\]

\[
e(t) = \frac{\vec{q}(t)}{\| \vec{q}(t) \|} \tag{12}
\]

where \( K_v, K_\alpha \in \mathbb{R}^{n \times n} \) are positive-definite diagonal matrices of adjustable variables. In order to achieve high-precision control performance, a robust feedback linearization control (RFLC) system as shown in Fig. 1 is introduced initially and its control algorithm is summarized in Theorem 1.

**THEOREM 1:** Consider an \( n \)-link robot manipulator including actuator dynamics represented by (7), if the RFLC law is devised as (13) and its control gains are designed as (14) and (15), then the convergence of tracking error and the stability of the RFLC system can be guaranteed.

\[
U_e = M_e (\dot{e} + \vec{K}_v \vec{q} + \ddot{q}) + D(q, \dot{q}, \dddot{q}) + \dot{k}_v \text{sgn}(e) \tag{13}
\]

\[
K_v = \begin{bmatrix} K_{\alpha} \end{bmatrix} \tag{14}
\]

\[
k_v = d_\_ \tag{15}
\]

where \( \text{sgn}(x) \) denotes the sign function of each element in the vector, \( x \).

According to Lyapunov stability theorem [14], the stable control performance of the RFLC system can be assured. However, if the system dynamics are perturbed or unknown, it is difficult to implement this control scheme in practical applications, and the stability of the controlled system may be destroyed. Besides, the chattering control effort could be induced due to the conservative selection of large control gain, \( k_v \). To ensure the stability of the controlled system despite the existence of the uncertainties, an RNFNC system is investigated in the following section.
iii. Robust Neural-Fuzzy-Network Control

In order to control the joint position of the robot manipulator more effectively, a RNFNC system as shown in Fig. 2(a) is constructed in this section. Moreover, a four-layer NFN as shown in Fig. 2(b), which comprises the input, membership, rule and output layers, is adopted to implement the RNFNC in this study [8], [9]. The inputs of the NFN are the elements in the error function vector, and the output of the RNFNC in this study \[8\], \[9\]. The inputs of the NFN are introduced as follows.

1) Input layer transmits the input linguistic variables \( e_i \) \((i = 1, \cdots, n)\) to the next layer, where \( e_i \) is the elements in the error function vector, \( e \).

2) Membership layer represents the input values with the following Gaussian membership functions:

\[
\mu_i(e) = \exp\left[-(e - m_i^j)^2/(s_i^j)^2\right]
\]

(16)

where \( m_i^j \) and \( s_i^j \) \((i = 1, \cdots, n; j = 1, \cdots, N_r)\), respectively, are the mean and standard deviation of the Gaussian function in the \( j \)th term of the \( i \)th input linguistic variable \( e_i \) to the node of this layer. For ease of notation, define adjustable parameter vectors \( m \) and \( s \) collecting all mean and standard deviation of Gaussian membership functions as \( m = [m_1^1 \cdots m_1^{N_r} \cdots m_n^1 \cdots m_n^{N_r}] \in R^{n \times N_r} \) and \( s = [s_1^1 \cdots s_1^{N_r} \cdots s_n^1 \cdots s_n^{N_r}] \in R^{n \times N_r} \), where \( N_r = \sum_{i=1}^{n} N_{r_i} \) denotes the total number of membership functions.

3) Rule layer implements the fuzzy inference mechanism, and each node in this layer multiplies the input signals and outputs the result of the product. The output of this layer is given as

\[
l_i = \prod_{i=1}^{n} w_{ik} \mu_i(e_i)
\]

(17)

where \( l_i \) \((k = 1, \cdots, N_r)\) represents the \( k \)th output of the rule layer; \( w_{ik} \), the weights between the membership layer and the rule layer, are assumed to be unity; \( N_r \) is the total number of rules.

4) Layer four is the output layer, and nodes in this layer represent output linguistic variables. Each node \( y_o \) \((a = 1, \cdots, N_o)\), which computes the output as the summation of all input signals, can be represented as

\[
y_o = \sum_{i=1}^{n} w_{oi} l_i
\]

(18)

Moreover, the output of the NFN can be represented in the following vector form:

\[
y = [y_1 y_2 \cdots y_N]^T = WL \equiv U_N(e, W, m, s)
\]

(19)

with

\[
W = [w_1 w_2 \cdots w_N]^T
\]

(20)

\[
l = [l_1 l_2 \cdots l_N]^T
\]

(21)

where \( w_i = [w_i^1 w_i^2 \cdots w_i^{N_r}] \).

The proposed RNFNC scheme comprises a NFN control and its associated network parameters tuning algorithm. The NFN control is designed to mimic the RFLC law in (13) to maintain the robust control performance without the requirement of system information and auxiliary compensated control. Moreover, the network parameters tuning laws are derived in the sense of projection algorithm [10] and Lyapunov stability theorem [14] to ensure the network convergence as well as stable control performance. According to the powerful approximation ability [6], there exists an optimal NFN control \( U_{N_{o}} \) to learn the RFLC law \( U_{\rho} \) such that

\[
U_{\rho} = U_{N_{o}}(e, W^*, m^*, s^*) + e = W^Tl^* + e
\]

(22)
where $\varepsilon$ is a minimum reconstructed error; $W^*$, $m^*$ and $s^*$ are optimal parameters of $W$, $m$ and $s$ in the NFN.

Thus, the NFN control law is assumed to take the following form:

$$U = \hat{U}_m (e, \hat{W}, \hat{m}, \hat{s}) = \hat{W} \hat{I}$$  \hspace{1cm} (23)

where $\hat{W}$, $\hat{m}$ and $\hat{s}$ are some estimates of the optimal parameters, as provided by tuning algorithms to be introduced later. Subtracting (23) from (22), an approximation error $\tilde{U}$ is defined as

$$\tilde{U} = U_g - U = W^* \hat{I} + \varepsilon - \hat{W} \hat{I} = \hat{W} \hat{I} + \hat{W} \tilde{I} + \varepsilon$$  \hspace{1cm} (24)

where $\tilde{W} = W^* - \hat{W}$ and $\tilde{I} = I^* - \hat{I}$. In this study, the linearization technique is employed to transform the membership functions into partially linear form so that the expansion of $\tilde{I}$ in Taylor series to obtain [2]

$$\tilde{I} = l_m \hat{m} + l_s \hat{s} + o_m$$  \hspace{1cm} (25)

where

$$l_m = \left[ \frac{\partial l_1}{\partial m} \frac{\partial l_2}{\partial m} \cdots \frac{\partial l_{n}}{\partial m} \right]_{m=a} R^{n \times n};$$

$$l_s = \left[ \frac{\partial l_1}{\partial s} \frac{\partial l_2}{\partial s} \cdots \frac{\partial l_{n}}{\partial s} \right]_{s=0} R^{n \times n};$$

$s = s^* - \hat{s}$; $o_m \in R^{n \times n}$ is a vector of higher-order terms.

Rewrite (25), one can obtain

$$l^* = \hat{I} + l_m \hat{m} + l_s \hat{s} + o_m$$  \hspace{1cm} (26)

Substituting (26) into (24), it is revealed that

$$\tilde{U} = W^* \hat{I} + \varepsilon - \hat{W} \hat{I}$$

$$= W^* [\hat{I} + l_m \hat{m} + l_s \hat{s} + o_m] + \varepsilon - \hat{W} \hat{I}$$

$$= (W^* - \hat{W}) \hat{I} + (\hat{W} + \hat{W}) l_m \hat{m} + (\hat{W} + \hat{W}) l_s \hat{s} + \varepsilon + W^* o_m$$

$$= \hat{W} \hat{I} + \hat{W} l_m \hat{m} + \hat{W} l_s \hat{s} + \hat{W} l_m \hat{m} + \hat{W} l_s \hat{s} + W^* o_m + \varepsilon$$

$$= \hat{W} \hat{I} + \hat{W} l_m \hat{m} + \hat{W} l_s \hat{s} + y'$$  \hspace{1cm} (27)

where $y' = \hat{W} l_m \hat{m} + \hat{W} l_s \hat{s} + W^* o_m + \varepsilon$.

**Theorem 2:** Consider an $n$-link robot manipulator including actuator dynamics represented by (7), if the NFN control law is designed as (23) and the adaptation laws of the NFN parameters are designed as (28)–(30), then the convergence of network parameters and tracking error of the proposed RNFNC system can be assured.

$$\dot{\tilde{m}} = \begin{cases} a_1 \varepsilon & \text{if } (\|\tilde{m}\| < b_1) \text{ or } (\|\tilde{m}\| = b_1 \text{ and } \varepsilon \|\tilde{m}\| > 0) \\ a_2 (\varepsilon \tilde{W}_1)^T \tilde{m} & \text{if } (\|\tilde{m}\| = b_1 \text{ and } \varepsilon \|\tilde{m}\| > 0) \end{cases}$$  \hspace{1cm} (29a)

$$\dot{\tilde{m}} = \begin{cases} a_3 (\varepsilon \tilde{W}_1)^T \tilde{m} & \text{if } (\|\tilde{m}\| < b_1) \text{ or } (\|\tilde{m}\| = b_1 \text{ and } \varepsilon \|\tilde{m}\| > 0) \\ a_4 (\varepsilon \tilde{W}_1)^T \tilde{m} & \text{if } (\|\tilde{m}\| = b_1 \text{ and } \varepsilon \|\tilde{m}\| > 0) \end{cases}$$  \hspace{1cm} (29b)

where $a_1$, $a_2$, and $a_3$ are positive learning rates; $b_1$, $b_2$, and $b_1$ are predetermined network parameter bounds.

Note that, $M^*$ and $y_s$ do not need to be known or specified beforehand since the terms $K_s$ and $k_s$ are not required in the RNFNC system. As a result, the stable control behavior can be ensured without the requirement of system information and the compensation of auxiliary control design. The effectiveness and robustness of the proposed RNFNC system can be verified by the following numerical simulations.

**IV. NUMERICAL SIMULATIONS**

A two-link robot manipulator as shown in Fig. 3 is utilized in this study to verify the effectiveness of the proposed control scheme. In Fig. 3, $q_1$ and $q_2$ are the angle of joints 1 and 2; $m_1$ and $m_2$ are the mass of links 1 and 2; $l_{11}$ and $l_{12}$ are the total length of links 1 and 2; $l_c$ and $l_s$ are the center-of-gravity length of links 1 and 2, and $g$ is the gravity acceleration. The detailed system parameters of this robot manipulator actuated by DC servomoters are given as follows:

$$J_{a1} = 3.7 \times 10^{-5} \text{ kgm}^2 \quad J_{a2} = 1.47 \times 10^{-4} \text{ kgm}^2$$

$$B_{a1} = 1.3 \times 10^{-4} \text{ Nm/s} \quad B_{a2} = 2 \times 10^{-4} \text{ Nm/s}$$

$$K_{s1} = 0.21 \text{ Nm/A} \quad K_{s2} = 0.23 \text{ Nm/A}$$

$$R_{a1} = 2.8 \Omega \quad L_{a1} = 3 \text{ mH} \quad K_{e1} = 2.42 \times 10^{-4} \text{ s/radV}$$

$$R_{a2} = 4.8 \Omega \quad L_{a2} = 2.4 \text{ mH} \quad K_{e2} = 2.18 \times 10^{-4} \text{ s/radV}$$
where the suffixes “1” and “2” denote the $i$th actuated motor or link of the robot manipulator. The control objective is to control the joint angles of a two-link robot manipulator to follow periodic sinusoidal trajectories. To show the effectiveness of the RNFNC system, the NFN has two, ten, twenty-five and two neurons at the input, membership, rule and output layer, respectively. That is, $n = 2$, $N_{r_1} = 5$, $N_{r_2} = 5$, $N_r = 10$, $N_r = 25$ and $N_r = 2$. Usually, some heuristics can be used to roughly initialize the parameters of the NFN for practical applications; e.g., the mean and standard deviation of Gaussian functions can be determined according to the maximum variation of $e(t)$. The effect due to the inaccurate selection of the initialized parameters can be retrieved by the on-line training methodology. In this study, the simulation is carried out using Window Matlab software.

The most important parameters that affect the control performance of the robotic system are the external disturbance $t_1(t)$, the friction term $f(q)$ and the parameter variation of the mass of link 2, $m_2$. In simulation, three circumstances including nominal situation ($m_2 = 0.75$kg and $N = 0$) at the beginning, parameter variation situation occurring at 4s, and disturbance situation occurring at 7s, are considered. The parameter variation situation is that 1kg weight is added to the mass of link 2, i.e., $m_2 = 1.75$kg. The disturbance situation is that external forces are injected into the robotic system and their shapes are expressed as follows:

$$t_1(t) = [5 \sin(5t) \ 0.5 \sin(5t)]$$

(32)

In addition, friction forces are also considered in this simulation and are given as

$$f(q) = [20 \dot{q}_1 + 0.8 \text{sgn}(\dot{q}_1) \ 4 \dot{q}_2 + 0.16 \text{sgn}(\dot{q}_2)]$$

(33)

To the end, two simulation cases (case S1 and case S2) including the aforementioned three situations are adopted to demonstrate the robust property of the proposed control scheme. Case S1 denotes the three situations without joint friction, and the friction forces are considered in case S2.

In order to exhibit the superior control performance of the proposed RNFNC scheme, the RFLC system as shown in Fig. 1 is examined in the meanwhile. In the RFLC, the system parameters $\overline{M}$ and $\overline{D}$ are used in (13)–(15), where the symbol “~” denotes the system parameters at nominal condition. The gains in these control systems are given as

$$K_s = [43 \ 0 \ 0], \ K_s = [120 \ 0 \ 0], \ K_s = [160 \ 0 \ 0],$$

(34)

$$k_1 = 1, \ a_1 = 1500, \ a_2 = 0.9, \ a_3 = 0.9, \ b_u = 10, \ b_u = 1, \ b_1 = 1$$

All the gains are chosen to achieve the superior transient control performance in the simulation considering the limitation of control effort, the requirement of stability and the possible operating conditions. The simulated results of the RFLC system at cases S1 and S2 are depicted in Fig. 4. The joint position responses at cases S1 and S2 are depicted in Fig. 4(a), (d) and 4(g), (j), respectively. The associated control efforts are depicted in Fig. 4(b), (e) and 4(h), (k); the associated tracking errors are depicted in Fig. 4(c), (f) and 4(i), (l). The robust control performance of the RFLC system is obvious under the occurrence of system uncertainties. However, the undesirable chattering phenomena in the control efforts, which are depicted in Figs. 4(b), (e) and 4(h), (k), are serious due to the excess selection of a bound value in (15). Now, the proposed RNFNC system shown in Fig. 2(a) is applied to control the robot manipulator for comparison. The joint position responses at cases S1 and S2 are depicted in Fig. 5(a), (d) and 5(j), (m), respectively. The associated control efforts are depicted in Fig. 5(b), (e) and 5(k), (n); the associated tracking errors are depicted in Fig. 5(c), (f) and 5(l), (o); the two norm of adjustable network parameters are depicted in Fig. 5(g)–(i) and 5(p)–(r). Since all parameters in the NFN are roughly initialized, the tracking errors are gradually reduced through on-line training process whether the uncertainties exist or not. Moreover, the robust control performance of the RNFNC system, both in the conditions of joint friction, parameter
variation and external disturbance, are obvious. Compared
these results with the RFLC system, the proposed RNFNC
system indeed yields superior tracking performance without
control chattering.

V. CONCLUSIONS

This study has been successfully implemented a RNFNC
system to control the joint position of an n-link robot
manipulator including actuator dynamics for achieving high-
precision position control. In the RNFNC system, all the
system dynamics could be unknown and no strict constraints
and auxiliary compensated control were required in the
control process. All adaptive learning laws in the RNFNC
system were derived in the sense of projection algorithm and
Lyapunov stability theorem, so that the network
convergence and system-tracking stability of the closed-loop
control system can be ensured whether the uncertainties
occur. According to the results as shown in Figs. 4 and 5, the
joint-position tracking responses of the RNFNC system can
be controlled to more closely follow specific reference
trajectories under a wide range of operating conditions and
the occurrence of uncertainties than the RFLC scheme.

VI. ACKNOWLEDGMENT

The authors would like to acknowledge the financial
support of the National Science Council of Taiwan, R.O.C.
through grant number NSC 93-2213-E-155-014.

VII. REFERENCES

for robotic motion control,” IEEE Trans. Ind. Electron., vol. 47, no. 2,
network controller for robot manipulators,” IEEE Trans. Syst., Man,
enhancement of robot systems including motor dynamics: a fuzzy-
Based dynamic game approach,” IEEE Trans. Fuzzy Syst., vol. 6, no. 4,
Hall, 1996.
fast learning algorithms,” IEEE Trans. Neural Networks, vol. 9, no. 1,
network control system for tracking periodic inputs,” IEEE Trans.
no. 4, pp. 519-532, July 2002.
Hall, 1997.
a robot manipulator,” IEE Proc. Contr. Theory Appl., vol. 144, no. 1,
Addison-Wesley, 1995.

Fig. 5. Numerical simulation of RNFNC system: (a)–(i) at joints 1 and 2 for
case S1; (j)–(r) at joints 1 and 2 for case S2.