Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 6, No. 1, February 2015), Pages: 47-62
www.jacr.iausari.ac.ir

Combined Neural Network Feedforward and RISE Feedback Control Structure for a 5 DOF Upper-limb Exoskeleton Robot with Asymptotic Tracking

Marzieh Yazdanzad[1]✉, Alireza Khosravi[1], Reza Ghaderi[2], Pouria Sarhadi[1]

1) Department of Electrical and Computer Engineering, Noshirvani Univ. of Technology, Babol, Iran
2) Department of Control Engineering, Shahid Beheshti Univ., Tehran, Iran

s.yazdanzad@stu.nit.ac.ir; akhosravi@nit.ac.ir; r_ghaderi@sbu.ac.ir; pouria.sarhadi@gmail.com

Received: 2014/04/14; Accepted: 2014/05/14

Abstract

Control of robotic systems is an interesting subject due to their wide spectrum applications in medicine, aerospace and other industries. This paper proposes a novel continuous control mechanism for tracking problem of a 5-DOF upper-limb exoskeleton robot. The proposed method is a combination of a recently developed robust integral of the sign of the error (RISE) feedback and neural network (NN) feed-forward terms. The feed-forward NN learns nonlinear dynamics of the system and compensates for uncertainties while the NN approximation error and nonlinear bounded disturbances are overcome by the RISE term. Typical NN-based controllers generally result in uniformly ultimately bounded (UUB) stability due to the NN reconstruction error. In this paper to eliminate this error and achieve asymptotic tracking, the RISE feedback term is integrated into the NN compensator. Finally, a comparative study on the system performance is conducted between the proposed control strategy and two other conventional control methods. Simulation results illustrate the effectiveness of the proposed method.

Keywords: *Robust integral of the sign of the error (RISE) feedback, Neural network (NN), Feed-forward compensation, 5-DOF upper-limb exoskeleton robot, Asymptotic tracking*

1. Introduction

Robotic systems have been used in various applications and can be found in many areas such as aerospace, medicine, automotive and other industries. Because of this wide range of applications, design of tracking controller for them has attracted significant attention in the past few decades. Robots are complex mechanical systems with highly nonlinear dynamics that are inevitably subject to various uncertainties such as parameters variations, nonlinear frictions, unknown external disturbances and etc. Several control strategies have been developed to achieve an accurate tracking control of robot manipulators such as robust control strategy [1], adaptive control [2,3,4], sliding mode control (SMC) approach [5,6,7] and neural network(NN) techniques [8,9].

Robust control is a common control scheme to attain good tracking performance in the presence of uncertainties. To design a controller using this technique, the bounds of uncertainties should be known in advance, which is usually hard to achieve. Hence, they are often selected conservatively. The implementation of the control law, based on these conservative bounds, leads to degradation of control performance due to unnecessarily high feedback gain selection. Adaptive controllers are suitable to apply on nonlinear systems with parametric uncertainties. But, this technique is often restricted to systems that are linear in the unknown parameters. Moreover, it requires an accurate dynamic model of system and tedious computations have to be done to determine the regression matrix. Sliding mode control is one of the effective strategies to control uncertain nonlinear systems. Main advantage of this method is strong robustness against system uncertainties [10]. However, this technique has some disadvantages including creating undesirable chattering phenomenon due to the discontinuous nature of control law, the requirement of knowing an exact knowledge of nonlinear dynamics, and the necessity of knowing bounds of uncertainties to achieve robust characteristics [11,12].

All the above mentioned methods require an accurate knowledge of dynamics of the system. However, it is difficult to obtain an exact mathematical model of robotic systems because of complexity of nonlinear dynamics, modeling uncertainties, unknown disturbances and etc. To cope with this problem, utilization of intelligent control approaches such as NN-based control and fuzzy control have received considerable attention [13, 14, 15]. Neural networks have an inherent learning ability and can approximate nonlinear continuous functions to arbitrary accuracy. They are therefore powerful tools to compensate for uncertainties without the requirement for a prior knowledge of the plant [16]. Lewis et al. in [17] developed a multilayer neural network controller for a general serial-link rigid robot which guarantees the tracking performance. However, the NN-based controllers only achieve uniformly ultimately bounded (UUB) stability results due to residual reconstruction error coming from NN approximation. Motivated by the desire to eliminate this error and achieving asymptotic tracking, Cheng et al. presented a sliding-mode neural-network controller for a mobile manipulator [18]. Despite obtaining asymptotic results, existence of a discontinuous sign function in the robust term causes chattering phenomenon. In this study, the robust integral of the sign of the error (RISE) feedback term is augmented to the NN feed-forward element to overcome the NN approximation error. The proposed control strategy generates a continuous-time control signal. Therefore, in addition to achieving asymptotic tracking, chattering problem of conventional SMC technique is solved.

Recently, a new feedback control scheme called robust integral of the sign of the error is proposed in [19]. This control strategy has been extensively studied because it can compensate for additive disturbances and uncertainties under the assumption that the disturbances are C^2 with bounded time derivatives by generating a continuous control signal [20, 21]. In [22], Patre et al. utilized this method to develop a tracking controller for a class of uncertain nonlinear systems. The RISE method is a high gain feedback tool. Motivated by this issue, in [23, 24], a feedforward element is combined with the RISE feedback structure in order to reduce the gain values of the RISE feedback.

In this paper, a control mechanism is presented for the tracking problem of a 5-DOF upper-limb exoskeleton robot. The proposed method is composed of a NN feed-forward and the RISE feedback control term. The NN feed-forward element is used to compensate for unknown system dynamics. However, due to NN approximation error the robot cannot asymptotically track the desired trajectory. Therefore, the RISE feedback control term is integrated into the NN feed-forward term to eliminate NN approximation error and achieve asymptotic tracking. This paper is organized as follows: Section 2 presents the characteristics of a dynamical model of a 5-DOF upper-limb exoskeleton robot. Then, in section 3, multilayer feed-forward neural network is briefly explained. In section 4, the proposed combined NN-RISE controller is introduced. Simulation results for the 5-DOF upper-limb exoskeleton robot are provided in section 5. Finally, conclusions are given in Section 6.

2. Dynamics of 5 DOF upper-limb exoskeleton robot

The dynamic of a general rigid link manipulator having n degrees of freedom in free space is as follows [16]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_d(t) = \tau(t) \qquad (1)$$

This dynamic equation is obtained by means of the Lagrangian approach. In equation (1) for the 5-DOF upper-limb exoskeleton robot, $q(t), \dot{q}(t), \ddot{q}(t) \in R^5$ represent the position, velocity and acceleration of the joints, respectively. $M(q) \in R^{5\times5}$ is the positive definite inertia matrix, $C(q,\dot{q}) \in R^{5\times5}$ expresses the Coriolis/centripetal matrix and $G(q) \in R^5$ is the gravity vector. $\tau_d(t) \in R^5$ denotes the vector of disturbances and unmodeled dynamics, and $\tau(t) \in R^5$ is the input torque vector applied to the joints. Figure 1 shows a schematic diagram of the 5-DOF upper-limb exoskeleton robot.

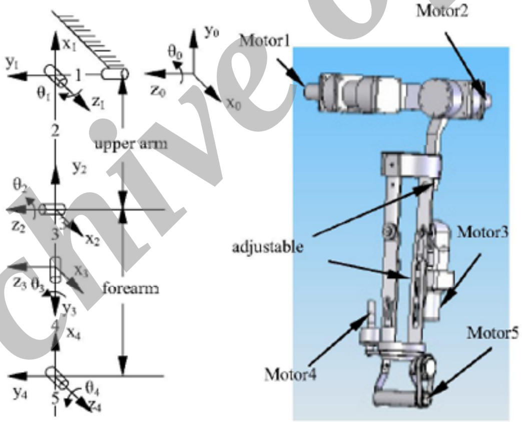


Figure 1. The model of 5-DOF upper-limb exoskeleton robot

The robot has 5 DOF, which are shoulder abduction/adduction, shoulder flexion/extension, elbow flexion/extension, wrist flexion/extension and internal/external rotation. Hence, it can simulate the human upper-limb movements.

In this system the inertia matrix $M(q)$, the Coriolis/centripetal matrix $C(q,\dot{q})$ and the gravity vector $G(q)$ are as follows [4]:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 \\ M_{21} & M_{22} & M_{23} & 0 & 0 \\ M_{31} & M_{32} & M_{33} & 0 & M_{35} \\ 0 & 0 & 0 & M_{44} & 0 \\ 0 & 0 & M_{53} & 0 & M_{55} \end{bmatrix}$$

with

$$\begin{aligned} M_{11} = I_2 + I_{19} + I_3 cos^2(q_2) + I_4 sin^2(q_2+q_3) + I_5 \sin(q_2+q_3)\cos(q_2+q_3) \\ + I_6 \sin(q_2)\cos(q_2) + I_7 sin^2(q_2+q_3) + 2 + I_8 \cos(q_2)\sin(q_2+q_3) \\ + I_9 \cos(q_2)\cos(q_2+q_3) + I_{10} sin^2(q_2+q_3) \\ + I_{11}\cos(q_2)\sin(q_2+q_3) + I_{12}\sin(q_2+q_3)\cos(q_2+q_3) \end{aligned}$$

$$\begin{aligned} M_{12} = M_{21} = I_{13}\sin(q_2) + I_{14}\cos(q_2+q_3) + I_{15}\cos(q_2) + I_{16}\sin(q_2+q_3) \\ - I_{17}\cos(q_2+q_3) \end{aligned}$$

$$M_{13} = M_{31} = I_{14}\cos(q_2+q_3) + I_{16}\sin(q_2+q_3) - I_{17}\cos(q_2+q_3)$$

$$M_{22} = I_{18} + I_{19} + I_{20} + 2I_8\sin(q_3) + I_9\cos(q_2) + I_{10} + I_{11}\sin(q_3)$$

$$M_{23} = M_{32} = I_8\sin(q_3) + I_{20} + I_9\cos(q_3) + I_{11}\sin(q_3) + 2I_{10}$$

$$M_{33} = I_{23} + I_{20} + 2I_{10}$$

$$M_{35} = M_{53} = I_{10} + I_{21}$$

$$M_{44} = I_{22} + I_{23}$$

$$M_{55} = I_{24} + I_{21}$$

and

$$C(q,\dot{q}) = \begin{bmatrix} c_1\dot{q}_2 & c_2\dot{q}_3 + c_3\dot{q}_2 & c_4\dot{q}_2 + c_5\dot{q}_3 & 0 & 0 \\ c_6\dot{q}_1 & c_7\dot{q}_3 & c_8\dot{q}_3 & 0 & 0 \\ c_5\dot{q}_1 & c_9\dot{q}_2 & 0 & 0 & 0 \\ c_{10}\dot{q}_2 & 0 & c_{11}\dot{q}_1 & 0 & 0 \\ 0 & c_{12}\dot{q}_2 & 0 & 0 & 0 \end{bmatrix}$$

where

$$\begin{aligned} c_1 = 2[-I_3\sin(q_2)\cos(q_2) + I_8\cos(2q_2+q_3) + I_4\sin(q_2+q_3)\cos(q_2) \\ - I_9 sin(2q_2+q_3) - 2I_{10}\sin(q_2+q_3) + I_{11}\cos(2q_2+q_3) \\ + I_7\sin(q_2+q_3)\cos(q_2+q_3) + I_{12}(1 - 2\sin(q_2+q_3)\sin(q_2+q_3))] \\ + I_5(1 - 2\sin(q_2+q_3) + I_6(1 - 2sin^2(q_2))) \end{aligned}$$

$$c_2 = 2[-I_{14}\sin(q_2+q_3) + I_{16}\cos(q_2+q_3) + I_{17}\sin(q_2+q_3)]$$

$$\begin{aligned} c_3 = I_{13}\cos(q_2) - I_{14}\sin(q_2+q_3) - I_{15}\sin(q_2) + I_{16}\cos(q_2+q_3) + I_{17}\sin(q_2 \\ + q_3) \end{aligned}$$

$$\begin{aligned} c_4 = 2[I_8\cos(q_2)\cos(q_2+q_3) + I_4\sin(q_2+q_3)\cos(q_2+q_3) - I_9\cos(q_2)\sin(2q_2) \\ + 2I_{10}\sin(q_2+q_3)\cos(q_2+q_3) + I_{11}\cos(q_2)\cos(q_2+q_3) \\ + I_7\sin(q_2+q_3)\cos(q_2+q_3) + I_{12}(1 - 2\sin(q_2+q_3)\sin(q_2+q_3))] \\ + I_5(1 - 2\sin(q_2+q_3)\sin(q_2+q_3)) \end{aligned}$$

$$c_5 = 0.5c_2$$

$$c_6 = -0.5c_1$$

$$c_7 = 2[-2I_9\sin(q_3) + I_8\cos(q_3) + I_{11}\cos(q_3)]$$

$$c_8 = -0.5c_4$$

$$c_9 = \sin(q_2 + q_3)\cos(q_2 + q_3) - 2I_{10}\sin(q_2 + q_3)\cos(q_2 + q_3) - I_{11}\cos(q_2)\cos(q_2 + q_3) - I_{12}cos^2(q_2 + q_3)$$
$$c_{10} = -[I_{23}\sin(q_2 + q_3) + I_{19}\sin(q_2 + q_3) + I_{20}\sin(q_2 + q_3)]$$
$$c_{11} = -I_{20}\sin(q_2 + q_3) + I_{23}\sin(q_2 + q_3) + I_{19}\sin(q_2 + q_3)$$
$$c_{12} = -I_{11}\cos(q_3) - I_{12}$$

and

$$G(q) = \begin{bmatrix} 0 \\ G_2 \\ G_3 \\ 0 \\ G_5 \end{bmatrix}$$

where

$$G_2 = g_1\cos(q_2) + g_2\sin(q_2 + q_3) + g_3\sin(q_2) + g_4\cos(q_2 + q_3) + g_5\sin(q_2 + q_3)$$
$$G_3 = g_2\sin(q_2 + q_3) + g_4\cos(q_2 + q_3) + g_5\sin(q_2 + q_3)$$
$$G_5 = g_5\sin(q_2 + q_3) \qquad (2)$$

where $q = [q_1\, q_2\, q_3\, q_4\, q_5]^T$ and $\dot{q} = [\dot{q}_1\, \dot{q}_2\, \dot{q}_3\, \dot{q}_4\, \dot{q}_5]^T$ contain the joints positions and velocities, respectively. The inertial constants and gravitational constants are listed in Tables 1 and 2.

Table 1. Inertial constants

$I_1 = 1.14$	$I_2 = 1.43$
$I_3 = 1.38$	$I_4 = 0.298$
$I_5 = -0.0213$	$I_6 = -0.0142$
$I_7 = -0.0001$	$I_8 = 0.372$
$I_9 = -0.011$	$I_{10} = 0.00125$
$I_{11} = -0.0124$	$I_{12} = 0.000058$
$I_{13} = -0.69$	$I_{14} = 0.134$
$I_{15} = 0.238$	$I_{16} = 0.00379$
$I_{17} = 0.000642$	$I_{18} = 4.71$
$I_{19} = 1.75$	$I_{20} = 0.333$
$I_{21} = 0.000642$	$I_{22} = 0.2$
$I_{23} = 0.00164$	$I_{24} = 0.179$

Table 2. Gravitational constants

$g_1 = -37.2$	$g_2 = -8.43$
$g_3 = 1.02$	$g_4 = 0.249$
$g_5 = -0.0292$	

It is assumed that $q(t)$ and $\dot{q}(t)$ are measurable and $M(q), C(q,\dot{q}), G(q)$ and $\tau_d(t)$ are unknown. Furthermore, the following properties for the dynamic model of the robot are hold:

Property 1: The inertia matrix $M(q)$ is symmetric, positive definite and holds the following inequality true:

$$m_1\|y\|^2 \leq y^T M(q) y \leq \bar{m}(q)\|y\|^2 \quad \square y \square R^5 \tag{3}$$

where $m_1 \square R$ is a known positive constant, $\bar{m}(q) \in R$ is a known positive function and $\|.\|$ represents the standard Euclidean norm.

Property 2: The nonlinear disturbance term and its first two time derivatives are bounded i.e. $\tau_d(t), \dot{\tau}_d(t), \ddot{\tau}_d(t) \square \mathcal{L}_\infty$.

Property 3: If $q(t), \dot{q}(t) \in \mathcal{L}_\infty$, then $C(q, \dot{q}), G(q)$ are bounded. Further, if $q(t), \dot{q}(t) \square \mathcal{L}_\infty$, then the first and second partial derivatives of the elements of $M(q), C(q, \dot{q}), G(q)$ with respect to $q(t)$ exist and are bounded. In addition, the first and second partial derivatives of the elements of $C(q, \dot{q})$ with respect to $\dot{q}(t)$ exist and are bounded as well.

3. Multilayer feed-forward neural network

It has been shown that a feed-forward NN with at least two layers can approximate any given smooth function on a compact set [16]. This property is called the universal function approximation. It is an important property that empowers the NNs in closed-loop control applications, especially when the dynamic model of system includes nonlinear disturbances and uncertain elements. Let $f(x): R^{N_1+1} \rightarrow R^n$ be a smooth function. Then for a given compact set $S \square R^{N_1+1}$ and a positive number ε_N, there is a NN having at least two layers, with sufficiently large number of neurons L in hidden layer such that:

$$f(x) = W^T \sigma(V^T x) + \varepsilon(x) \tag{4}$$

where $\|\varepsilon\| < \varepsilon_N$ for all given inputs $x \square S$ [16, 25]. The value $\varepsilon(x)$ is called the NN functional reconstruction error. In (4), $V \square R^{(N_1+1)\times N_2}$ and $W \square R^{(N_2+1)\times n}$ are bounded constant ideal weight matrices for the input-to-hidden and hidden-to-output layers respectively, where N_1 is the number of input-layer neurons, N_2 is the number of hidden-layer neurons and n is the number of output-layer neurons. $\sigma(.): R^{N_2+1} \rightarrow R^{N_2+1}$ is the activation function in the hidden layer. The input vector $x(t)$ and the activation function $\sigma(.)$ are augmented by "1", so that the first column of the weight matrices V and W are considered as thresholds for the first and second layers [16, 25]. Therefore, any tuning of V and W contains tuning of thresholds.

Now based on (4), the NN functional estimate of $f(x)$ is given as [16]:

$$\hat{f}(x) \triangleq \widehat{W}^T \sigma\left(\hat{V}^T x\right) \tag{5}$$

where $\hat{V}(t) \square R^{(N_1+1)\times N_2}$ and $\widehat{W}(t) \square R^{(N_2+1)\times n}$ are the estimates of the ideal weight matrices.

The weight estimation errors, denoted by $\tilde{V}(t) \square R^{(N_1+1)\times N_2}$ and $\widetilde{W}(t) \square R^{(N_2+1)\times n}$, are defined as follows:

$$\tilde{V} = V - \hat{V}, \widetilde{W} = W - \widehat{W} \tag{6}$$

and the hidden-layer output error for a given input $x(t)$, denoted by $\tilde{\sigma}(x)\ \square\ R^{N_2+1}$, is defined as follows:

$$\tilde{\sigma} \triangleq \sigma - \hat{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x) \tag{7}$$

Property 4: (Boundedness of the ideal weights) Based on the assumption that the ideal weights exist and are bounded by known positive values, the following inequalities hold:

$$\|V\|_F{}^2 = tr(V^T V) \leq \bar{V}_B \tag{8}$$

$$\|W\|_F{}^2 = tr(W^T W) \leq \bar{W}_B \tag{9}$$

where $\|.\|_F$ and $tr(.)$ are the Frobenius norm and the trace of a matrix, respectively.

4. Combined NN feed-forward and RISE feedback controller design

4.1 Control objective

The objective is to design a controller for the robot that guarantees tracking of a desired trajectory, in spite of uncertainties and bounded disturbances in the dynamic model. To formulate this objective, a position tracking error, denoted by $e_1(t)\ \square\ R^n$, is defined as:

$$e_1 = q_d - q \tag{10}$$

where $q_d(t)\ \square\ R^n$ is a desired trajectory. In the following, it is assumed that the desired trajectory is designed such that $q_d{}^{(i)}(t) \in \mathcal{L}_\infty, i = 0,1,\ldots,7$.
The filtered tracking errors, denoted by $e_2(t), r(t)\ \ \square\ R^n$, are also defined as:

$$e_2 = \dot{e}_1 + \alpha_1 e_1 \tag{11}$$

$$r = \dot{e}_2 + \alpha_2 e_2 \tag{12}$$

where $\alpha_1\ \square\ R^{n\times n}$ is a positive constant matrix and $\alpha_2\ \square\ R$ is a positive constant.

4.2 Open-loop error system

Multiplying both sides of equation (12) by $M(q)$ and then using (1), (10) and (11) into it, the open-loop error system is obtained as follows:

$$M(q)r = N(q_d, \dot{q}_d, \ddot{q}_d) + S(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d) + \tau_d - \tau \tag{13}$$

where the auxiliary functions $N(q_d, \dot{q}_d, \ddot{q}_d)$ and $S(q, \dot{q}, q_d, \dot{q}_d, \ddot{q}_d)$ are defined as:

$$N = M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) \tag{14}$$

$$S = M(q)(\alpha_1 \dot{e}_1 + \alpha_2 e_2) + C(q, \dot{q})\dot{q} + G(q) + M(q)\ddot{q}_d - N \tag{15}$$

The expression in (14) can be represented by a three-layer NN as follows:

$$N = W^T \sigma(V^T x_d) + \varepsilon(x_d) \qquad (16)$$

where the input $x_d(t)$ R^{3n+1} is defined as $x_d \triangleq [1\ q_d^T(t)\ \dot{q}_d^T(t)\ \ddot{q}_d^T(t)]^T$, so that $N_1 = 3n$ where N_1 was introduced in (4).

With the assumption that the desired trajectory is bounded, the following inequalities are hold:

$$\|\varepsilon(x_d)\| \le \varepsilon_{b_1}, \|\dot{\varepsilon}(x_d, \dot{x}_d)\| \le \varepsilon_{b_2}, \|\ddot{\varepsilon}(x_d, \dot{x}_d, \ddot{x}_d)\| \le \varepsilon_{b_3} \qquad (17)$$

where $\varepsilon_{b_1}, \varepsilon_{b_2}, \varepsilon_{b_3}$ R are known positive constants.

4.3 Closed-loop error system

In previous section, a neural network is employed to estimate uncertain nonlinear dynamics of the robot. However, it should be noted that the existence of NN functional approximation error leads to UUB stability results. In this paper, in order to eliminate this error and achieve asymptotic tracking, the NN-based feed-forward method is augmented by the RISE feedback control term, $\mu(t)$ R^n, which is defined as follows [19]:

$$\mu(t) = (k_s + 1)e_2(t) - (k_s + 1)e_2(0)$$
$$+ \int_0^t [(k_s + 1)\alpha_2 e_2(\sigma) + \beta_1 sgn(e_2(\sigma))]d\sigma \qquad (18)$$

where its time derivative is as follows:

$$\dot{\mu}(t) = (k_s + 1)r + \beta_1 sgn(e_2) \qquad (19)$$

In (18), k_s, β_1 R are positive constant control gains. Figure 2 shows the block diagram of the proposed control scheme.

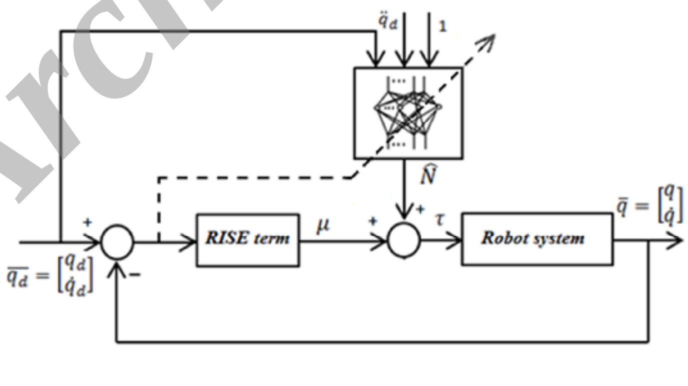

$\ddot{q}_d$
1
$\hat{N}$
+
+
+
-
RISE term
μ
τ
Robot system
$\bar{q} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$
$\overline{q_d} = \begin{bmatrix} q_d \\ \dot{q}_d \end{bmatrix}$

Figure 2. Block diagram of the control scheme

Therefore, based on the open-loop error system in (13), the control torque input is composed of two components, a NN feed-forward term and the RISE feedback control term as below:

$$\tau = \hat{N} + \mu \qquad (20)$$

where the NN component, $\hat{N}(t) \in R^n$, is as follows:

$$\hat{N} \triangleq \hat{W}^T \sigma\left(\hat{V}^T x_d\right) \tag{21}$$

The adaptation law for the estimates of the ideal weight matrices, $\hat{W}$ and $\hat{V}$, are as follows:

$$\dot{\hat{W}} \triangleq proj(\Gamma_1 \hat{\sigma}' \hat{V}^T \dot{x}_d {e_2}^T) \tag{22}$$

$$\dot{\hat{V}} \triangleq proj(\Gamma_2 \dot{x}_d ({\hat{\sigma}'}^T \hat{W} e_2)^T) \tag{23}$$

where $\Gamma_1 \in R^{(N_2+1)\times(N_2+1)}$ and $\Gamma_2 \in R^{(N_1+1)\times(N_1+1)}$ are constant, positive definite, symmetric control gain matrices. The operator $proj(.)$ denotes the projection algorithm [26]. It is used to guarantee that the estimated weights, $\hat{W}$ and $\hat{V}$, remain bounded. By substituting the control input (20) into (13), the close-loop tracking error system is achieved as follows:

$$M(q)r = N - \hat{N} + S + \tau_d - \mu \tag{24}$$

By differentiating the equation (24), and substituting (16) and (21) into it, then adding and subtracting the terms $W^T \hat{\sigma}' \hat{V}^T \dot{x}_d + \hat{W}^T \hat{\sigma}' \tilde{V}^T \dot{x}_d$ to the resulted equation, and substituting (19), (22) and (23) into it, the following equation can be obtained:

$$M(q)\dot{r} = -\frac{1}{2}\dot{M}(q)r + \tilde{Q} + Q - e_2 - (k_s + 1)r - \beta_1 sgn(e_2) \tag{25}$$

where

$$\tilde{Q} \triangleq -\frac{1}{2}\dot{M}(q)r - proj\left(\Gamma_1 \hat{\sigma}' \hat{V}^T \dot{x}_d {e_2}^T\right)^T \hat{\sigma} - \hat{W}^T \hat{\sigma}' proj\left(\Gamma_2 \dot{x}_d \left({\hat{\sigma}'}^T \hat{W} e_2\right)^T\right)^T x_d$$

$$+\dot{S} + e_2 \tag{26}$$

and in a similar manner as in [22], Q is separated as:

$$Q \triangleq Q_d + Q_B \tag{27}$$

$$Q_d \triangleq W^T \sigma' V^T \dot{x}_d + \dot{\varepsilon} + \dot{\tau}_d \tag{28}$$

$$Q_B \triangleq Q_{B_1} + Q_{B_2} \tag{29}$$

$$Q_{B_1} \triangleq -W^T \hat{\sigma}' \hat{V}^T \dot{x}_d - \hat{W}^T \hat{\sigma}' \tilde{V}^T \dot{x}_d \tag{30}$$

$$Q_{B_2} \triangleq \hat{W}^T \hat{\sigma}' \tilde{V}^T \dot{x}_d + \tilde{W}^T \hat{\sigma}' \hat{V}^T \dot{x}_d \tag{31}$$

Using the Mean Value Theorem, Property 2, (8), (9), (17), (22), (23), (29)-(31), the following inequalities are hold:

$$\begin{aligned}&\left\|\tilde{Q}\right\| \le \rho(\|z\|)\|z\|\\ &\|Q_d\| \le \xi_1, \|Q_B\| \le \xi_2, \left\|\dot{Q}_d\right\| \le \xi_3 \qquad (32)\\ &\left\|\tilde{Q}_B\right\| \le \xi_4 + \xi_5\|e_2\|\end{aligned}$$

where $z(t)$ □ R^{3n} is defined as:

$$z(t) \triangleq [e_1^T \;\; e_2^T \;\; r^T]^T \qquad (33)$$

and the function $\rho(\|z\|)$ □ R is a positive globally invertible nondecreasing function. Furthermore ξ_i □ $R, i = 1,2,3,4,5$ are known as positive constants.

4.4 Stability analysis

Theorem: The proposed controller in (20) guarantees that all system signals are bounded under closed-loop operation, and the position tracking error goes to zero as follows:

$$\|e_1(t)\|, \|e_2(t)\|, \|r(t)\| \to 0 \;\; as \;\; t \to \infty \qquad (34)$$

The mentioned results can be achieved while the RISE feedback control gain k_s introduced in (18), is selected large enough, and β_1 and β_2 are chosen according to the following sufficient conditions:

$$\beta_1 > \xi_1 + \xi_2 + \frac{1}{\alpha_2}\xi_3 + \frac{1}{\alpha_2}\xi_4 \, , \beta_2 > \xi_5 \qquad (35)$$

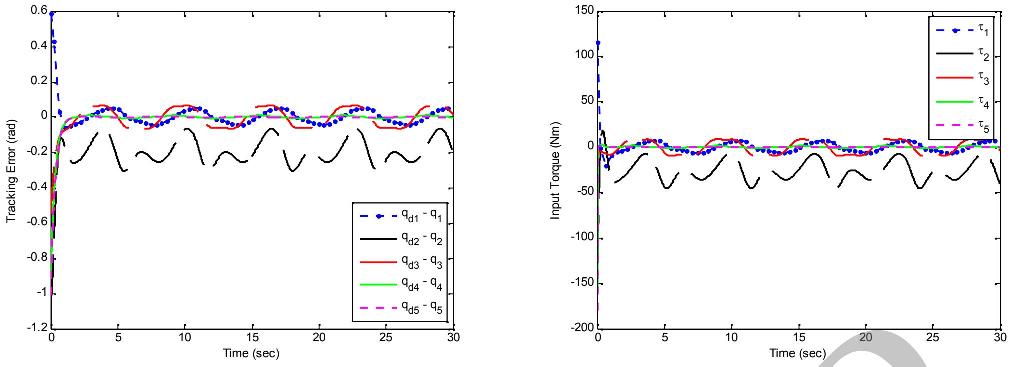
where ξ_i □ $R, i = 1,2,3,4,5$ are introduced in (32) and β_2 will be introduced in the proof.
Proof: See the Lyapunov based proof in reference [22].

5. Simulation results

In order to evaluate the validity of the proposed controller, a numerical simulation for tracking control of a 5-DOF upper-limb exoskeleton robot is performed. Furthermore, simulation results are compared with the other control methods including PD control and NN-based control to illustrate the effectiveness and superiority of the proposed control scheme.

Consider the robot system introduced in section 2. The control objective is to design a controller to track the desired trajectories, $q_{di} = \sin\left(t + {}^{i\pi}/_{5}\right), i = 1, \dots, 5$. Further, the external disturbances $\tau_{d1} = 0.2\sin(2t), \tau_{d2} = 0.1\cos(2t), \tau_{d3} = 0.1\sin(t), \tau_{d4} = 0.2\sin(t), \tau_{d5} = 0.1\cos(t)$ are applied to the system. Meanwhile, The initial state of the system is set at $q_d(0) = [0\ 2\ 1.5\ 1.5\ 1\ 0\ 0\ 0\ 0\ 0]^T$.

First case (PD controller):
The PD controller is implemented as $\tau = k_v e_2 = k_v \dot{e}_1 + k_v \alpha_1 e_1$ where the feedback control gain matrix, k_v, and the design parameter matrix, α_1, are selected as $45 \times I^{5\times5}$ and $3 \times I^{5\times5}$. The results of PD controller are shown in figure 3. As it can be seen, the tracking errors always exist.



3(a). Tracking error

3(b). Control inputs

Figure 3. The results of PD controller

Second case (NN-based controller): The NN-based controller is constructed by [17]:

$$
\begin{aligned}
&\tau = \hat{W}^T \sigma\left(\hat{V}^T x\right) + k_v e_2 - v \\
&\dot{\hat{W}}^T = F \hat{\sigma} e_2^T - F \hat{\sigma}' \hat{V}^T x e_2^T - k_W F \left\|e_2\right\| \hat{W} \\
&\dot{\hat{V}} = G x \left(\hat{\sigma}' \hat{W}^T e_2\right)^T - k_v G \left\|e_2\right\| \hat{V} \\
&v(t) = -K_z \left(\left\|\hat{Z}\right\|_F + Z_B\right) e_2
\end{aligned}
\tag{36}
$$

where the NN input, x, is defined as $x = [1, e_1^T, \dot{e}_1^T, q_d^T, \dot{q}_d^T, \ddot{q}_d^T]$, $Z = diag\{W, V\}$ and $\|Z\|_F \le Z_B$. The structure of the NN-based control method is shown in figure 4.

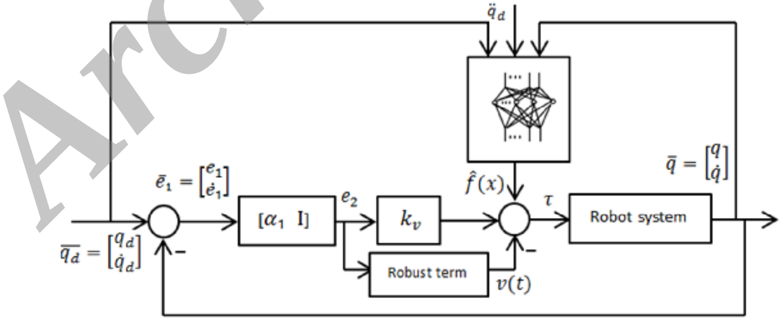
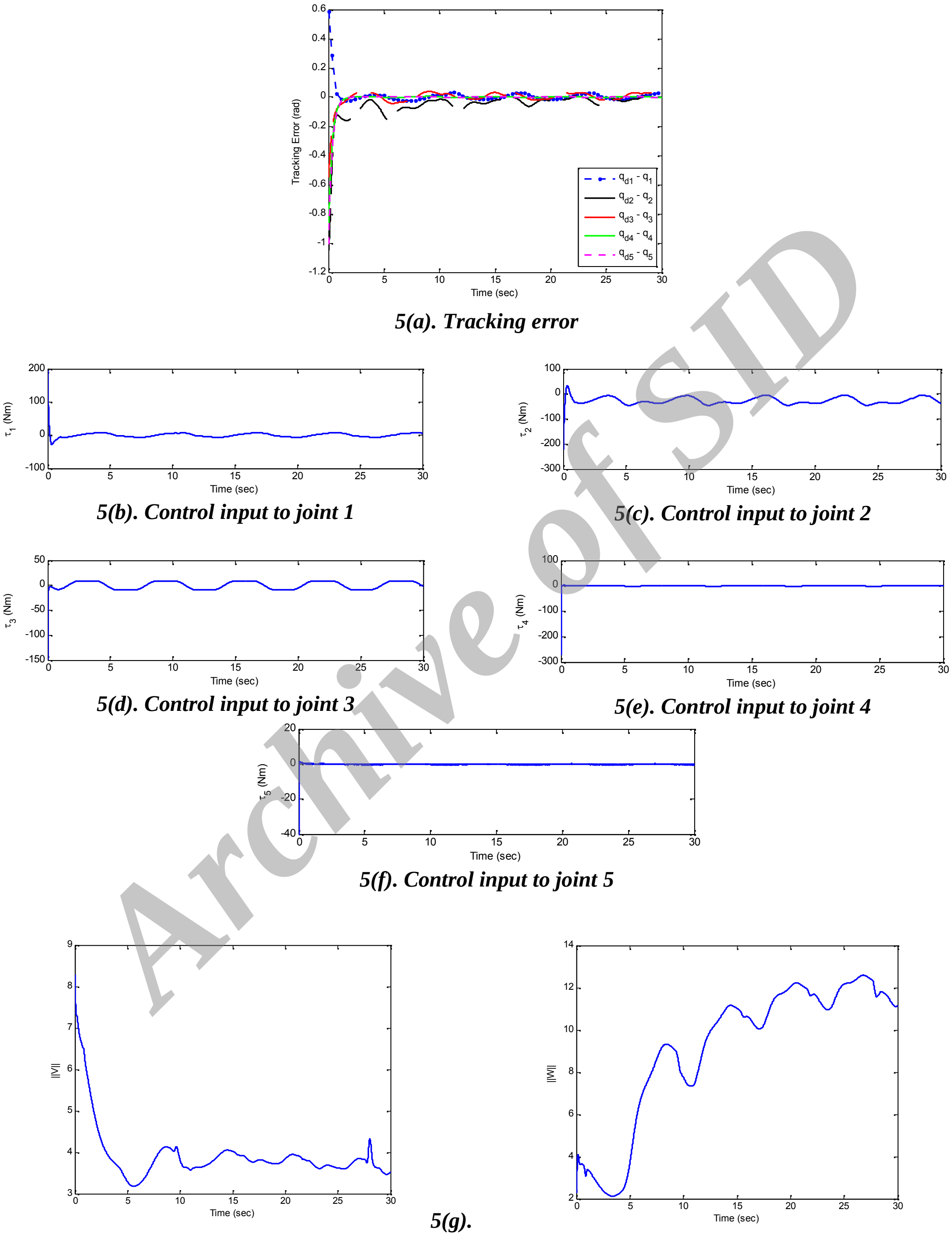

$\ddot{q}_d$
$\bar{q} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$
$\bar{e}_1 = \begin{bmatrix} e_1 \\ \dot{e}_1 \end{bmatrix}$
$\hat{f}(x)$
e_2
τ
$[\alpha_1 \ \ \mathrm{I}]$
k_v
Robot system
$\overline{q_d} = \begin{bmatrix} q_d \\ \dot{q}_d \end{bmatrix}$
−
Robust term
−
$v(t)$

Figure 4. Multilayer NN controller structure

The values of α_1 and k_v, are similar to those of PD controller. The design parameters are selected as $F = 5 \times I^{10\times10}$, $G = 5 \times I^{26\times26}$, $\kappa_W = 0.1$, $\kappa_v = 0.1$, $K_z = 2$ and $Z_B = 6$. The number of neurons in hidden layer of the NN is selected as $N_2 = 10$ and its activation function is considered as $logsig(.)$. The weights of the first and second layer of the NN are initialized at $rand(26,10)$ and $0.5 \times rand(10,5)$. The results of NN-based controller are shown in figure 5.



5(a). Tracking error

5(b). Control input to joint 1

5(c). Control input to joint 2

5(d). Control input to joint 3

5(e). Control input to joint 4

5(f). Control input to joint 5

5(g).

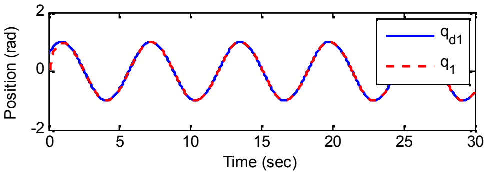
Norm of the first layer weights

5(h). Norm of the second layer weights

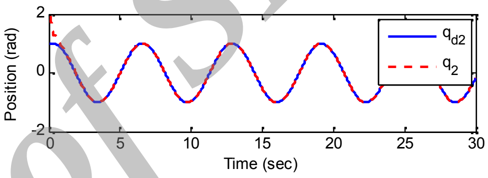
Figure 5. The results of NN controller

It can clearly be seen that the tracking errors are smaller than those of PD controller. In spite of the improvement in the tracking performance, the system cannot asymptotically track the desired trajectory because of the NN reconstruction errors. In the following, it is demonstrated that the tracking errors can be asymptotically converge to zero in a shorter time by using the proposed method.

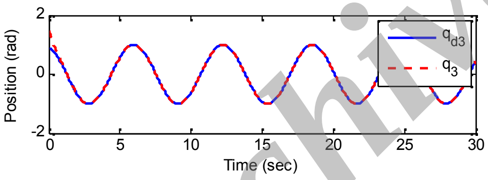
Third case (The RISE feedback controller in combination with the NN feed-forward term): Now, in order to design the proposed control scheme in this paper, the control torque input presented in (20) is used. The value of α_1 is the same as before. The control gains are chosen as $\alpha_2 = 28$, $k_s = 50$, $\beta_1 = 6$. In this simulation a neural network with $N_2 = 10$ hidden-layer neurons and $n = 5$ neurons in the output layer is considered. The initial weights of the first and second layer of NN are set as $\hat{V} = randn(16{,}10)$ and $\widehat{W} = 10 \times randn(10{,}5)$. The activation function of the NN is selected as $logsig(.)$ and the adaptation gains are selected as $\Gamma_1 = 5 \times I^{10\times10}$, $\Gamma_2 = 0.05 \times I^{16\times16}$. The results of this controller are shown in figure 6 which confirms its superior tracking performance.



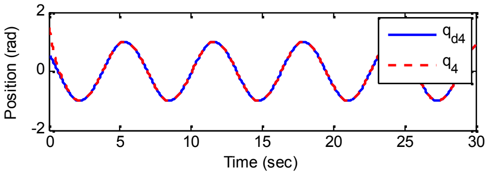
6(a). The position tracking of q_1 to the desired trajectory q_{d1}



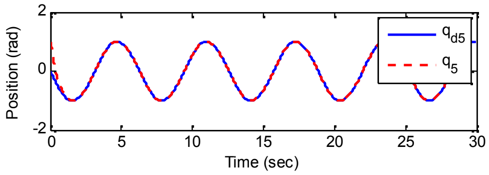
6(b). The position tracking of q_2 to the desired trajectory q_{d2}



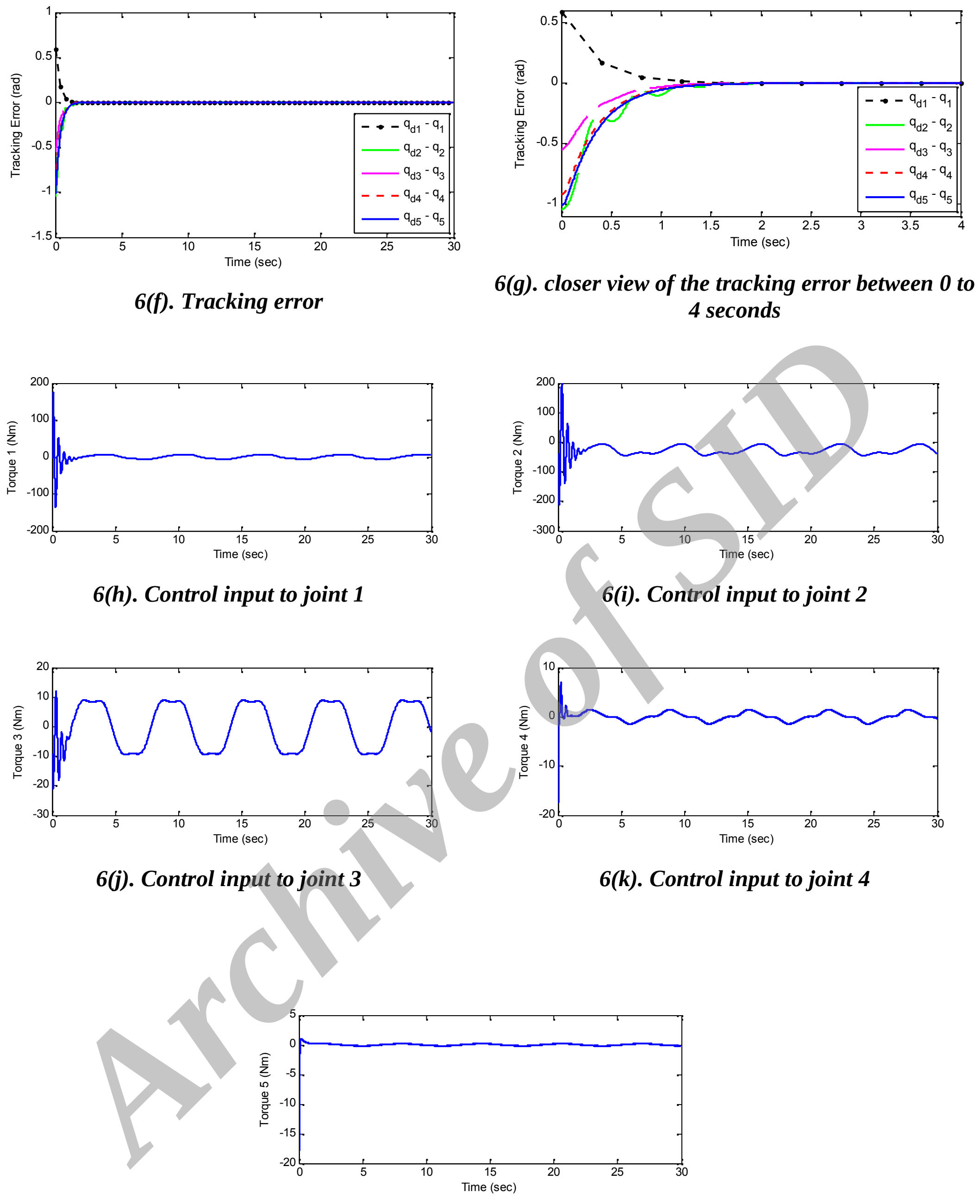
6(c). The position tracking of q_3 to the desired trajectory q_{d3}



6(d). The position tracking of q_4 to the desired trajectory q_{d4}



6(e). The position tracking of q_5 to the desired trajectory q_{d5}



6(f). Tracking error

6(g). closer view of the tracking error between 0 to 4 seconds

6(h). Control input to joint 1

6(i). Control input to joint 2

6(j). Control input to joint 3

6(k). Control input to joint 4

6(l). Control input to joint 5

Figure 6. The results of the proposed controller in this paper

Figures 6(a) to 6(e) show the tracking performance of the proposed method. The curves demonstrate that the control objective is successfully achieved. The position tracking error is shown in figure 6(f). It can be seen that an asymptotic tracking is achieved for all joints of the robot. Figures 6(h) to 6(l) show the generated control torque input. As it can be seen, the proposed controller generates a continuous control signal. In fact, the existence of a unique integral sign term in the RISE controller causes a continuous control structure which avoids the occurrence of chattering phenomenon that usually happens in sliding mode controllers. By comparing figure 6 with figure 5, it can be seen that unlike the NN-based controller which provides the convergence only with UUB error, the proposed control scheme in this paper yields asymptotic tracking of course in better performance.

6. Conclusion

This paper has presented a continuous control strategy for the tracking problem of a 5-DOF upper-limb exoskeleton robot in the presence of uncertainties and bounded external disturbances. The proposed method is a combination of the recently developed RISE feedback control technique and a NN feed-forward term. The feed-forward NN approximates nonlinear dynamics of the robot and compensates for uncertainties without the requirement for a precise model of system. The RISE feedback term simultaneously eliminates the NN approximation error. The results of the proposed control scheme are compared with a PD and also a NN-based controller. The Comparative study on the system performance demonstrated that the proposed method yielded superior tracking performance. The NN-based controller provided the convergence only with UUB error due to the NN approximation error. However, the proposed method achieves asymptotic tracking by eliminating this error. Furthermore, Continuity of the generated control signal in this method prevents the occurrence of undesirable chattering phenomenon.

7. References

[1] Guangjun Liu, Andrew A and Goldenberg, “Robust Control of Robot Manipulators Based on Dynamics Decomposition,” *IEEE Trans. Robotics and automation,* 13, (1997) 783-789.

[2] R. Colbaugh and K. Glass, “Adaptive tracking control of rigid manipulators using only position measurements,” *Journal of Robot and System,* 14 no. 1 (1997) 9–26.

[3] C. C. Cheah, C. Liu and J. J. E. Slotine, “Adaptive Tracking Control for Robots with Uncertainties in Kinematic, Dynamic and Actuator Models,” *IEEE Transactions on Automatic Control,* 51(6) (2006) 1024-1029.

[4] H. B. Kang and J. H. Wang, “Adaptive control of 5 DOF upper-limb exoskeleton robot with improved safety,” *ISA Transactions, 2013*.

[5] Guangjun Liu and A.A. Goldenberg, “Uncertainty Decomposition–Based Robust Control of Robot Manipulators,” *IEEE Trans Control Sys. Technology*. 4 (1996) 384–393.

[6] G.J. Luis Enrique, L. Alexander G., E.J. Bayro-Corrochano, “Integral Nested sliding Mode Control for Robotic Manipulators,” *Proceedings of 17th IFAC World Congress*. (2008) 9899-9904.

[7] Y.-W. Liang, S.-D. Xu, D.-C. Liaw, and C.-C. Chen, “A study of T–S model-based SMC scheme with application to robot control,” *IEEE Trans. Ind. Electron,* 55, no. 11 (2008) 3964–3971.

[8] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma,Y. Uchikawa, “Trajectory control of robotic manipulator using neural networks,” *IEEE Trans. Ind. Electron,* 38 (1991) 195–202.

[9] F. Sun, Z. Sun, P.Y.Woo, “Neural network-based adaptive controller design of robotic manipulators with an observer,” *IEEE Trans. Neural Networks,* 12 (2001) 54–67.

[10] X. Yu, O. Kaynak, “Sliding-Mode Control with Soft Computing: A Survey,” *IEEE Trans. On Industrial Electronics,* vol. 56, no. 9, 2009.

[11] J.J.E. Slotine, W.P. Liu, Applied Nonlinear Control, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[12] Boiko, *et al.*, “Analysis of chattering in systems with second-order sliding modes,” *IEEE Transactions on Automatic Control,* 52 (2007) 2085-2102.

[13] S. N. Huang, K. K. Tan, T.H. Lee, “Adaptive neural network algorithm for control design of rigid-link electrically driven robots,” *Neurocomputing,* 71 (2008) 885–894.

[14] Y.H. Kim, F.L. Lewis, “Optimal design of CMAC neural-network controller for robot manipulators,” IEEE Trans. Systems Man Cybernet. 30 (2000) 22.

[15] Rong-jong Wa and Po-Chen Chen, “Robust Neural-Fuzzy-Network Control for Robot Manipulator Including Actuator Dynamics,” *IEEE Trans. Indst. Elect,* 53, no. 4, (2006).

[16] F. L. Lewis, S. Jagannathan, and A. Yesildirek, Neural Network Control of Robot Manipulators and Nonlinear Systems, Tayor & Francis, 1999.

[17] F.L. Lewis, A. Yesildirek, K. Liu, “Multilayer neural-net robot controller with guaranteed tracking performance,” *IEEE Trans. Neural Networks,* 7 (1996) 388– 399.

[18] M. B. Cheng and C. C. Tsai, “Hybrid robust tracking control for a mobile manipulator via sliding-mode neural network,” in Proc. IEEE Int. Conf. Mechatronics, Taipei, Taiwan, (2005) 537–542.

[19] B. Xian, D.M. Dawson, M.S. de Queiroz and J. Chen, “A continuous asymptotic tracking control strategy for uncertain multi-input nonlinear systems,” *IEEE Transactions on Automatic Control,* 49 (2004) 1206–1211.

[20] C. Makkar, G. Hu, W. G. Sawyer, and W. E. Dixon, “Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction,” *IEEE Trans. Automat. Control*, 52 (2007).

[21] Z. Cai, M. S. de Queiroz, and D. M. Dawson, “Robust adaptive asymptotic tracking of nonlinear systems with additive disturbance,” *IEEE Trans. Automat. Control*, 51(3) (2006).

[22] P.M. Patre, W. MacKunis, W.E. Dixon, “Asymptotic tracking for uncertain dynamic systems via a multilayer NN feedforward and RISE feedback control structure,” *IEEE Transactions on Automatic Control*, 53(9) (2008) 2180–2185.

[23] W. MacKunis, P. M. Patre, M. K. Kaiser, and W. E. Dixon, “Asymptotic tracking for aircraft via robust and adaptive dynamic inversion methods,” *IEEE Trans. Cont. Syst. Technol*, 18(6) (2010).

[24] P. M. Patre, W. MacKunis, C. Makkar, and W. E. Dixon, “Asymptotic tracking for systems with structured and unstructured uncertainties,” *IEEE Trans. Cont. Syst. Technol*. 16(2) (2008).

[25] F. Lewis, J. Campos, and R. Selmic, Neuro-Fuzzy Control of Industrial Systems With Actuator Nonlinearities. Philadelphia, PA: SIAM, 2002.

[26] W. E. Dixon, A. Behal, D. M. Dawson, and S. P. Nagarkatti, Nonlinear Control of Engineering Systems: a Lyapunov-Based Approach. Bostot, Birkhuser, (2003).