An improved Adaptive Online Neural Control for Robot Manipulator Systems Using Integral Barrier Lyapunov Functions

Jun Xia[a], Yujia Zhang[b], Chenguang Yang[c,*], Min Wang[d] and Andy Annamalai[e]

[a] Sun Yat-sen University, Guangzhou, People's Republic of China;
[b] City University of Hong Kong, Kowloon Tong, Hong Kong;
[c] Bristol Robotics Laboratory, University of the West of England, Bristol, UK;
[d] South China University of Technology, Guangzhou, People's Republic of China;
[e] University of Winchester, Hampshire, UK

ARTICLE HISTORY
Compiled February 7, 2019

ABSTRACT
Conventional Neural Network (NN) control for robots use radial basis function (RBF) and for n-link robot with online control, the number of nodes and weighting matrix increases exponentially, which requires a number of calculations to be performed within a very short duration of time. This consumes a large amount of computational memory and may subsequently result in system failure. To avoid this problem, this paper proposes an innovative NN robot control using a dimension compressed RBF (DCRBF) for a class of n-degree of freedom (DOF) robot with full-state constraints. The proposed DCRBF NN control scheme can compress the nodes and weighting matrix greatly and provide an output that meets the prescribed tracking performance. Additionally, adaption laws are designed to compensate for the internal and external uncertainties. Finally, effectiveness of the proposed method has been verified by simulations. The results indicate that the proposed method, integral Barrier Lyapunov Functions (iBLF) avoids the existing defects of Barrier Lyapunov Functions (BLF) and prevents the constraint violations.

KEYWORDS
Neural network (NN); Radial basis function (RBF); Integral Barrier Lyapunov Functions (iBLF); Prescribed trajectory tracking

1. Introduction

With the widely use of complex robot manipulators which are nonlinear systems in our modern society and industry, research into robot technologies has attracted enormous attention Alford and Belyeu (1984); Cheng, Hou, Tan, and Zhang (2012); Gueaieb, Karray, and Al-Sharhan (2007); G.-W. Lee and Cheng (1996); T. Li, Duan, Liu, Wang, and Huang (2016); Na, Mahyuddin, Herrmann, Ren, and Barber (2015); Namvar and Aghili (2005)Alford and Belyeu (1984); Cheng et al. (2012); Gueaieb et al. (2007); G.-W. Lee and Cheng (1996); T. Li et al. (2016); Na et al. (2015); Namvar and Aghili (2005). Meanwhile, the control of robot system in the presence of uncertain parameters and motion constraints were extensively studied (Z. Li, Ge, & Ming, 2007; Tang, Ge,

*Corresponding author: Chenguang Yang. Email: cyang@ieee.org

Tee, & He, 2016a). In recent decades, adaptive control of complex nonlinear systems such as robot manipulators with full-state constraints and uncertainties has been developed to deal with theoretical challenges and practical needs (Cheng, Cheng, Yu, Deng, & Hou, 2016; Y. Huang, Na, Wu, Liu, & Guo, 2015; T. Lee, Koh, & Loh, 1996; G.-H. Yang & Ye, 2006). In the field of adaptive control, neural networks (NNs) are always considered as an efficient way to handle the uncertain or poorly known dynamics due to their universal approximation capabilities (Cheng, Liu, Hou, Yu, & Tan, 2015; Hou, 2001; Kennedy & Chua, 1988; Z. Li, Li, & Feng, 2016). It is very difficult to establish exact mathematical dynamics models with various uncertainties e.g., unknown payloads (Arefi & Jahed-Motlagh, 2013; Arefi, Jahed-Motlagh, & Karimi, 2015; G.-H. Yang & Wang, 2001; Zhang & Ge, 2009). However, by exploiting NN approximation, many complex and challenging models can be established easier but not sacrificing many characteristics of accurate models (Chen & Ge, 2013; Dai, Wang, & Wang, 2014; Gao & Selmic, 2006; T. H. Lee & Harris, 1998; Z. Li, Ge, Adams, & Wijesoma, 2008). In (C. Yang, Wang, Cheng, & Ma, 2016), a direct adaptive NN scheme is presented for a class of uncertain nonlinear strict-feedback systems. By utilizing a special property of the affine term, the developed scheme can avoid the controller singularity problem completely. The adaptive control of a strict feedback nonlinear systems using multilayer neural network was studied in (Zhang, Ge, & Hang, 2000) so as to guarantee the uniform ultimate boundedness of the closed-loop adaptive systems. In (C. Yang, Jiang, Li, He, & Su, 2017), RBF NN based control for coordinated dual arms robots have been proposed to settle the uncertainties. (He, Chen, & Yin, 2016) utilized NN of the conventional RBF NN structure for n-link robot but with lots of nodes. In this paper, an innovative DCRBF NN is proposed. The $n - DOF$ input is split into n $1-DOF$ inputs and built a conventional RBF NN for each $1-DOF$ input. Subsequent mathematical manipulations provide the output of every node in each NN (by adding two layers) and it ensures that the tracking performance of the controller output does not deteriorate. In this way, we can compress the number of nodes and the weights in an extreme degree with the performance similar to the traditional one, which might be important for particular practical applications by saving time and energy. The approximation error between the output of DCRBF and conventional RBF is proved to be bounded.

The stabilization of robot system is another important requirement of controller design. In practical systems, violation of constraints may cause degeneration of the control performance or even system failures (L. Huang, Ge, & Lee, 2006; Su, Leung, & Zhou, 1992; Tee, Ren, & Ge, 2011). To handle with the constraint problem, many methods have been proposed such as model predictive control (MPC), optimal control and reference governors so on. It is always necessary to know the exact model which is quiet complicated for complex robot when utilizing the method of MPC and optimal control (Berkovitz, 2013; Luo, Wu, & Li, 2015; Mayne, Rawlings, Rao, & Scokaert, 2000; Rubio, 2012). However, in the situation of uncertainties or some sophisticated model which cannot be calculated, these control methods can not be employed and it is necessary to find some alternatively solutions.

To solve the problem of system control in the presence of constraints and uncertainties, Barrier Lyapunov Functions (BLF) are popularly used, since they have the ability to shape the control performance (Liu & Tong, 2017). For example, the tracking control problem is studied in (He et al., 2016) for an uncertain n-link robot with full-state constraints and a BLF is designed to guarantee the uniform ultimate boundedness of the closed-loop system. In (Tee et al., 2011), BLF is employed at the outset to prevent the output from violating the time-varying constraint in strict feedback nonlinear systems. However, BLF-based controls have its limitations. One is that the feasibility conditions have a tendency to be conservative when ensuring constraint satisfaction, due to the original state are enforced indirectly by imposing transformed constraints on the errors. In (He, Zhang, Ge, & Liu, 2014) iBLF based boundary controls was proposed for a class of inhomogeneous Timoshenko beam satisfying the needs of suppressing the undesirable vibrations and preventing the constraint transgression. In (Tang, Ge, Tee, & He, 2016b), iBLFs are constructed to handle the unknown affine control gains with state constraints . In order to accomplish the prescribed tracking performance considering transient and steady states, the iBLF technique is exploited in this paper. The main contribution of this paper are as follows:

- An innovative DCRBF NN is proposed and it can be employed to avoid the exponential growth of nodes and weights with an increase in the DOF. Additionally, it inherently takes care of the inevitable uncertainties in the dynamics of the robot.
- The mathematical proof of the DCRBF NN is presented. A rigorous proof of the new algorithm is presented and its effectiveness is verified in theory and simulation.
- In order to avoid the violation of constraints while using BLF on n-link robots, a novel iBLF is utilized to design the control strategy which incorporates the output constraints and provides an enhanced system stability.

The rest of the paper is organized as follows. Section II gives the problem formulation of the n-link robot manipulator and some useful preliminaries for deriving proof. In Section III, The control design and the stability confirmation for the system are proposed. Besides, a comparison between conventional RBF NN control and DCRBF NN control is demonstrated and the mathematical proof of DCRBF NN is presented. Simulation studies are carried out to testify the effectiveness of the designed control and DCRBF NN in Section IV.

2. Problem Formulation and Preliminaries

2.1. System Description

The dynamics of an n-link rigid robotic system in the following Lagrange form (Craig, 2005):

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau(t) - J^T f(t). \tag{1}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ represent the position, velocity and acceleration respectively; $M(q) \in \mathbb{R}^{n\times n}$ denotes a symmetric positive definite inertia matrix; $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the centripetal and Coriolis torques, which is hard to obtain; $G(q) \in \mathbb{R}^{n\times n}$ is the unknown gravitational force; J^T is the Jacobian matrix for $f(t)$; $f(t)$ represents the unknown internal and external disturbances such as friction and so on; $\tau \in \mathbb{R}^{n\times n}$ represents the input torques.

Property 1. The matrix $M(q)$ is symmetric and positive definite and there exist positive constraints $0 < m_1 < m_2$ so that $M(q)$satisfies $m_1 I < M(q) < m_2 I$.

Due to the known position and velocity of robot system , we can obtain the accelerations. Choosing $x_1 = q \in \mathbb{R}^n, x_2 = \dot{q} \in \mathbb{R}^n$, we have the description of the robot as

$$\begin{cases} \dot{x_1} = x_2, \\ \dot{x_2} = M^{-1}(x_1)[\tau - C(x_1, x_2) - G(x_1) - J^T(x_1)f(t)], \end{cases} \tag{2}$$

According to property 1, there exist positive constants n_1, n_2, such that $n_1 I < M^{-1} < n_2 I$. The desired trajectory of the position is $x_d(t) = [q_{d1}(t), q_{d2}(t), ..., q_{dn}(t)]^T$ and the desired trajectory is $\alpha(t) = [\alpha_1(t), \alpha_2(t), \cdots, \alpha_n(t)]$. Assuming all signals and state constraints are bounded, we have constants k_{c1}, such that $-k_{c1} \leq x_1(t) \leq k_{c1}, \forall t \geq 0$, where $k_{c1} = [k_{c11}, k_{c12}, ..., k_{c1n}]^T$ are positive constant vectors.

2.2. Required Technology Lemmas and Definitions

Lemma 2.1 ((He et al., 2016))**.** *If there existe a Lyapunov function $V(x)$,which is positive definite and continuous satisfying $\xi_1(||x||) \leq V(x) \leq \xi_2(||x||)$ so that $\dot{V}(x) \leq -c_1 V(x) + c_2$, where c_1, c_2 are the positive constants and ξ_1, ξ_2 are the functions making $\mathbb{R}^n \to \mathbb{R}$, the parameters and states of the system will remain in a compact set and eventually converge to a specific compact sets.*

Lemma 2.2. *For the adaptive law (50), there exists a compact set*

$$\Omega_w = \{\hat{W}_{k,\gamma} | ||\hat{W}_{k,\gamma}|| \leq nm^2 \frac{s}{\theta_\gamma}\} \tag{3}$$

where for$||S(Z)|| \leq s$, $||S'(Z)|| \leq nm^2 s$ with $s > 0$, such that $\hat{W}_{k,\gamma(t)} \in \Omega_w, \forall t \geq 0$ provided that $\hat{W}_{k,\gamma}(0) \in \Omega_w$.

Proof. For the conventional RBF NN, according to the nature of Gauss's function, it can be seen that there exists a positive constant s so that $||S(Z)|| \leq s$. According to (39), the structure of our $S'(Z)$, we know that for $i = 1, 2, \cdots, nm^4$, $|S'(Z)_i| \leq \sqrt{n}s$. Thus, for $S'(Z) = [S'(z)_1, S'(z)_2, \cdots, S'(z)_{nm^4}]$, we have $||S'(Z)|| \leq nm^2 s$, The according to (Craig, 2005), we can obtain (3). □

Lemma 2.3 (Barbalat's Lemma)**.** *Suppose $f(t) \in C^1(a, \infty)$ and $lim_{t\to\infty} f(t) = \alpha$ where $\alpha \leq \infty$. If $\dot{f}(x)$ is uniformly continuous, then $lim_{t\to\infty} \dot{f}(t) = \alpha$.*

Lemma 2.4 (RBF approximation(Wang & Yang, 2017))**.** *If there allow sufficient nodes, under suitable width Δ and node centers $\bar{\boldsymbol{\delta}}$, RBF NN can approximate any smooth function $\boldsymbol{F_a(x)}$ over a compact set $\boldsymbol{x} \in \boldsymbol{\Omega_x}$ with convergent errors: $\boldsymbol{F_a(x)} = \boldsymbol{W^* S(x)} + \boldsymbol{\eta(x)}$ where $\boldsymbol{W^*}$ is the ideal weight matrix, $\boldsymbol{\eta(x)}$ is the convergent errors, satisfying $||\boldsymbol{\eta(x)}|| \leq \bar{\boldsymbol{\eta}}$, $\bar{\boldsymbol{\eta}}$ is the constant vector.*

Lemma 2.5 (RBF and optimal weights)**.** *According to (Haykin, Haykin, Haykin, & Haykin, 2009), without loss of generality, we can use least-square method and recursive least-square method to solve the optimal wights and there is equivalence between two methods in mathematical terms. For leat-square method, we have*

$$W = \phi(Z)^\dagger g(n_i) \tag{4}$$

where n_i is the number of training sample and $g(n_i)$ is $n_i \times 1$ desired response.† is the symbol for generalized inverse. $\phi(Z) = [S(Z_1)^T, S(Z_2)^T, \cdots, S(Z_{n_i})^T]^T$ is a interpo-

lating matrix. As for using recursive least-square method, we have

$$\begin{aligned} W(n_i) =& [\phi(n_i)^T\phi(n_i)]^{-1}S(n_i)d(n_i) \\ &+ W(n_i-1) \end{aligned} \tag{5}$$

where $\phi(n_i) = [S(Z_1)^T, S(Z_2)^T, \cdots, S(Z_{n_i})^T]^T$, $S(n_i) = S(Z_{n_i})$ *and* $d(n_i)$ *is the desired response of* n_i *sample.*

3. Control Design

3.1. Integral Barrier Lyapunov Functions Design

3.1.1. Integral Barrier Lyapunov Functions Design

Before proceeding to control design, we denote $e = [e_1, e_2, \cdots, e_n]^T = x_1 - x_d$, $z = [z_1, z_2, \cdots, z_n]^T = x_2 - \alpha$.

Assumption 1. For $i = 1, 2, \cdots, n$, there exists $k_{ai} = k_{c1i} + x_o$ and x_o is a small positive constant,such that $x_1 = [x_{11}, x_{12}, \cdots, x_{1n}]$ and the desired trajectory $x_d = [x_{d1}, x_{d2}, \cdots, x_{dn}]$ satisfy

$$|x_{1i}| \leq k_{c1i} < k_{ai}, |x_{di}| \leq k_{c1i} < k_{ai} \tag{6}$$

for all $i = 1, 2, \cdots, n$.

For the $n - link$ robotic arm, consider an iBLF candidate

$$V_1 = \sum_{i=1}^{n} \int_0^{e_i} \frac{\sigma k_{ai}^2}{k_{ai}^2 - (\sigma + x_{di})^2} \, d\sigma \tag{7}$$

where $e_i = x_{1i} - x_{di}$ and x_{di} are continuously differentiable functions satisfying $|x_{di}| < k_{ai}$ for $i = 1, 2, \cdots, n$. According to Property 3, we can see that V_1 is positive definite.

Differentiating V_1 with respect to time, we have

$$\dot{V}_1 = \sum_{i=1}^{n} \frac{k_{ai}^2 e_i (z_i + \alpha_i)}{k_{ai}^2 - x_{1i}^2} - \sum_{i=1}^{n} \rho_i e_i \dot{x}_{di} \tag{8}$$

where

$$\rho_i = \int_0^1 \frac{k_{ai}^2}{k_{ai}^2 - (\beta e_i + x_{di})^2} \, d\beta \tag{9}$$

$$= \frac{k_{ai}}{2e_i} \ln \frac{(k_{ai} + e_i + x_{di})(k_{ai} - x_{di})}{(k_{ai} - e_i - x_{di})(k_{ai} + x_{di})} \tag{10}$$

Then, a virtual controller α_i can be designed as

$$\alpha_i = (-k_{1i} e_i + \frac{(k_{ai}^2 - x_{1i}^2)\dot{x}_{di}}{k_{ai}^2} \rho_i), i = 1, 2, \cdots, n \tag{11}$$

where k_{1i} is a positive control gain for $i = 1, 2, \cdots, n$, we obtain

$$\dot{V}_1 = -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2} + \sum_{i=1}^{n} \frac{k_{ai}^2 e_i z_i}{k_{ai}^2 - x_{1i}^2} \tag{12}$$

Then, we design a positive Lyapunov candidate function as

$$V_2 = V_1 + \frac{1}{2} z^T M z \tag{13}$$

Then differentiating V_2 with respect to time leads to

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + z^T M \dot{z} \\ &= -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2} + \sum_{i=1}^{n} \frac{k_{ai}^2 e_i z_i}{k_{ai}^2 - x_{1i}^2} + z^T[\tau - J^T(x_1) f \\ &\quad - C(x_1, x_2) x_2 - G(x_1) - M(x_1)\dot{\alpha}]\end{aligned} \tag{14}$$

According to the expression of $\dot{V}_2$, we design the control law as $\tau = \tau_1 + \tau_2$, where according to lemma 2.4, τ_1 uses adaptive dimension compressed RBF neural network control described in subsection B and τ_2 is designed as

$$\tau_2 = -(z^T)^{\dagger} \sum_{i=1}^{n} \frac{k_{ai}^2 e_i z_i}{k_{ai}^2 - x_{1i}^2} - k_2 z \tag{15}$$

where k_2 is the control gain, and k_{2i}, $i = 1, 2, \cdots, n$ are positive constants. Then substituting τ into (14), according to Moore-Penrose inverse, $\dot{V}_2 = -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2}$, when $z = [0, 0, \cdots, 0]^T$. According to the lemma 2.3, we can still draw the asymptotic stability of the system. Otherwise, in the case of $z \neq [0, 0, \cdots, 0]^T$, we obtain

$$\begin{aligned}\dot{V}_2 = &-\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2} - z^T k_2 z + z^T[\tau_1 - J^T(x_1) f - C(x_1, x_2) x_2 \\ &- G(x_1) - M(x_1)\dot{\alpha}]\end{aligned} \tag{16}$$

3.1.2. Useful Property

Property 2. For any positive constant k_{ai}, the following inequality holds for any e_i and x_{di}, in the interval $|x_{di}| < k_{ai}$,$|e_i + x_{di}| = |x_{1i}| < k_{ai}$, for $i = 1, 2, \cdots, n$:

$$\begin{aligned}V_1 &= \sum_{i=1}^{n} \int_0^{e_i} \frac{\sigma k_{ai}^2}{k_{ai}^2 - (\sigma + x_{di})^2} d\sigma \\ &\leq \sum_{i=1}^{n} \frac{k_{ai}^2 e_i^2}{{k_{ai}}^2 - (e_i + x_{di})^2}\end{aligned} \tag{17}$$

Proof. Denote $p(\sigma, x_{di}) = (\sigma k_{ai}^2)/(k_{ai}^2 - (\sigma + x_{di})^2)$. We can show that

$$\sum_{i=1}^{n} \frac{\partial p}{\partial \sigma} = \sum_{i=1}^{n} \frac{k_{ai}^2 + \sigma^2 - x_{di}^2}{k_{ai}^2 - (\sigma + x_{di})^2} \tag{18}$$

which is positive in the set $|\sigma + x_{di}| < k_{ai}$. Since $p(0, x_{di}) = 0$ for $|x_{di}| < k_{ai}$, and $p(\sigma, x_{di})$ is monotonically increasing with the σ in the set $|\sigma + x_{di}| < k_{ai}$, we can obviously see that

$$\sum_{i=1}^{n} \int_0^{e_i} \frac{\sigma k_{ai}^2}{k_{ai}^2 - (\sigma + x_{di})^2} d\sigma \leq \sum_{i=1}^{n} e_i p(e_i, x_{di}) \tag{19}$$

for $|e_i + x_{di}| < k_{ai}$, which leads to the (17) after substituting for p. □

Property 3. By Assumption1, the V_1 is positive definite, continuously differentiable, and satisfies the decrescent condition in the set $|x_{1i}| \leq k_{c1i} < k_{ai}$, for $i = 1, 2, \cdots, n$:

$$\sum_{i=1}^{n} \frac{e_i^2}{2} \leq V_1 \leq \sum_{i=1}^{n} e_i^2 \int_0^1 \frac{\beta k_{ai}^2}{k_{ai}^2 - (e_i \beta + sgn(e_i) k_{c1i})^2} d\beta \tag{20}$$

which is useful for establishing uniformly stability.

Property 4. Using L'Hôpital's rule, it can be shown that

$$\lim_{e_i \to 0} \rho_i = \lim_{e_i \to 0} \frac{k_{ai}^2}{k_{ai}^2 - (e_i + x_{di})^2} \tag{21}$$

$$= \frac{k_{ai}^2}{k_{ai}^2 - x_{di}^2} \tag{22}$$

Since $|x_{di}| \leq k_{c1i} < k_{ai}$, for $i = 1, 2, \cdots, n$, by Assumption 1, ρ_i is bounded and well-defined in a neighborhood of $e_i = 0$.

3.2. NN Design

3.2.1. Dimension Split for Radial Basis Function

Figure 1 shows the architecture of the inputs space of an adaptive neural network control with $n-DOF$ arm. $Z = [x_1, x_2, \alpha, \dot{\alpha}]^T$ are the inputs, which has $4n$ dimensions. If each dimension has m types of centres, there will be m^{4n} dots and nm^{4n} weights in neural network. Thus it can be seen that with the degree of freedom n increasing, the dots and weights increase exponential.

Without loss of generality, for $n - link$ arms ($n - DOF$), let us express the conventional RBF NN as the form below.

$$F(Z) = W^T S(Z) \tag{23}$$

where $W = [W_1^T, W_2^T, \cdots, W_n^T]$ are the weights of the neural networks, $S(Z) = [S(Z)_1, S(Z)_2, \cdots, S(Z)_{m^{4n}}]^T$ are the basis functions. Without loss of generality, we

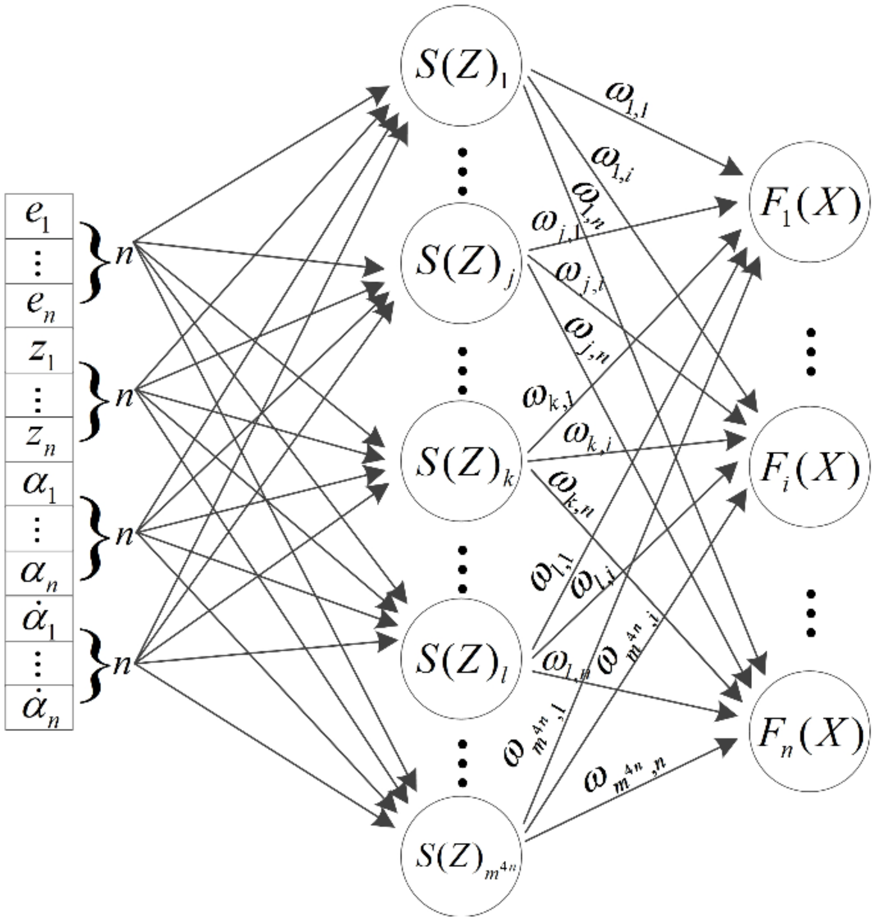


Figure 1. Conventional RBF NN

choose $S(Z)_i$ where $i = 1, 2, \cdots, m^{4n}$, and expanse it. So we can obtain

$$\begin{aligned} S(Z)_i &= e^{-\frac{\|Z-\bar{\delta}_i\|^2}{\Delta^2}} \\ &= e^{-\frac{\sum\limits_{j=1}^{n}(e_j-\delta_{i,j})^2+\sum\limits_{j=1}^{n}(z_j-\delta_{i,n+j})^2+\sum\limits_{j=1}^{n}(\alpha_j-\delta_{i,2n+j})^2+\sum\limits_{j=1}^{n}(\dot{\alpha}_j-\delta_{i,3n+j})^2}{\Delta^2}} \end{aligned} \tag{24}$$

where Δ is the bandwidth and $\bar{\delta}_i = [\delta_{i,1}, \delta_{i,2}, \cdots, \delta_{i,4n}], i = 1, 2, \cdots, m^{4n}$ is the ith centre. It should be noted that the aforementioned expression of $S(Z)_i$ can be split by each dimension and recombined as

$$\begin{aligned} S(Z)_i &= e^{-\frac{\sum\limits_{j=1}^{n}((e_j-\delta_{i,j})^2+(z_j-\delta_{i,n+j})^2+(\alpha_j-\delta_{i,2n+j})^2+(\dot{\alpha}_j-\delta_{i,3n+j})^2)}{\Delta^2}} \\ &= \prod_{j=1}^{n} f_{k_j} \end{aligned} \tag{25}$$

where $k_j = 1, 2, \cdots, m^4$,for $j = 1, 2, \cdots, n$,and

$$f_{k_j} = e^{-\frac{(e_j-\delta_{k_j,j})^2+(z_j-\delta_{k_j,n+j})^2+(\alpha_j-\delta_{k_j,2n+j})^2+(\dot{\alpha}_j-\delta_{k_j,3n+j})^2}{\Delta^2}} \tag{26}$$

Then substituting (25) and (26) into (23), we can obtain the expression of RBF NN after the dimension split. Choose a certain dimension γ of $F(Z)$ to show it below, $\gamma = 1, 2, \cdots, n$.

$$\begin{aligned} F(Z)_\gamma &= W_\gamma^T S(Z) \\ &= \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} \omega_{k_1,k_2,\cdots,k_n,\gamma} f_{k_n} \end{aligned} \tag{27}$$

where

$$W_\gamma = \begin{bmatrix} \omega_{k_1=k_2=k_3=\cdots=k_n=1,\gamma} \\ \omega_{k_1=2,k_2=k_3=\cdots=k_n=1,\gamma} \\ \vdots \\ \omega_{k_1=m^4,k_2=k_3=\cdots=k_n=1,\gamma} \\ \omega_{k_1=1,k_2=2,k_3=k_4=\cdots=k_n=1,\gamma} \\ \omega_{k_1=2,k_2=2,k_3=k_4=\cdots=k_n=1,\gamma} \\ \vdots \\ \omega_{k_1=m^4,k_2=2,k_3=k_4=\cdots=k_n=1,\gamma} \\ \vdots \\ \omega_{k_1=1,k_2=m^4,k_3=k_4=\cdots=k_n=1,\gamma} \\ \omega_{k_1=2,k_2=m^4,k_3=k_4=\cdots=k_n=1,\gamma} \\ \vdots \\ \omega_{k_1=m^4,k_2=m^4,k_3=k_4=\cdots=k_n=1,\gamma} \\ \vdots \\ \omega_{k_1=k_2=k_3=k_4=\cdots=k_n=m^4,\gamma} \end{bmatrix} \tag{28}$$

and

$$S(Z) = \begin{bmatrix} f_{k_1=1} f_{k_2=1} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ f_{k_1=2} f_{k_2=1} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ \vdots \\ f_{k_1=m^4} f_{k_2=1} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ f_{k_1=1} f_{k_2=2} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ f_{k_1=2} f_{k_2=2} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ \vdots \\ f_{k_1=m^4} f_{k_2=2} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ \vdots \\ f_{k_1=1} f_{k_2=m^4} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ f_{k_1=2} f_{k_2=m^4} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ \vdots \\ f_{k_1=m^4} f_{k_2=m^4} f_{k_3=1} f_{k_4=1} \cdots f_{k_n=1} \\ \vdots \\ f_{k_1=m^4} f_{k_2=m^4} = \cdots = f_{k_n=m^4} \end{bmatrix} \tag{29}$$

3.2.2. Compression Matrix A

For the better illustration of DCRBF, let us introduce an operator matrix $A(m^{4n} \times nm^4)$first, which could be used to compress the numbers of the weights. To construct compression matrix A, a series of $m^4 \times m^4$ submatrices ψ_i, for $i = 1, 2, 3 \cdots, m^4$ is built as

$$\psi_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \tag{30}$$

$$\psi_2 = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \cdots & 0 \end{bmatrix} \tag{31}$$

$$\vdots$$

$$\psi_{m^4} = \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{32}$$

Then use ψ_i and unit matrix E to design the compression matrix A:

$$A = \begin{bmatrix} E & \psi_1 & \psi_1 & \psi_1 & \cdots & \psi_1 \\ E & \psi_2 & \psi_1 & \psi_1 & \cdots & \psi_1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ E & \psi_{m^4} & \psi_1 & \psi_1 & \cdots & \psi_1 \\ E & \psi_1 & \psi_2 & \psi_1 & \cdots & \psi_1 \\ E & \psi_2 & \psi_2 & \psi_1 & \cdots & \psi_1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ E & \psi_{m^4} & \psi_2 & \psi_1 & \cdots & \psi_1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ E & \psi_1 & \psi_{m^4} & \psi_1 & \cdots & \psi_1 \\ E & \psi_2 & G_{m^4} & \psi_1 & \cdots & \psi_1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ E & \psi_{m^4} & \psi_{m^4} & \psi_1 & \cdots & \psi_1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ E & \psi_{m^4} & \psi_{m^4} & \psi_{m^4} & \cdots & \psi_{m^4} \end{bmatrix} \tag{33}$$

3.2.3. DCRBF

Inspired by expression format of conventional RBF NN in (27), a dimension-split RBF NN of $n - DOF$ is built in Figure2, which only has nm^4 dots and nm^4 weights for each degree of freedom of the outputs, which avoids the exponential growth with the

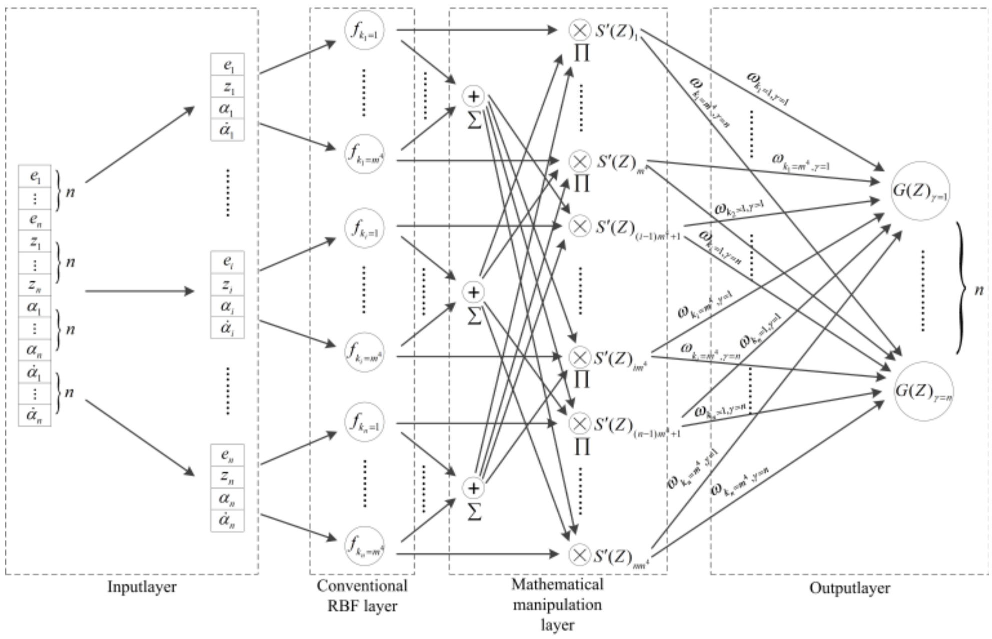


Figure 2. Dimension compressed RBF NN

DOF. As Figure2 shown, in input layer, each DOF of x_1, x_2, α and $\dot{\alpha}$ are taken out from the input Z, and reassembled as a new input vector $\xi_i = [x_{1i}, x_{2i}, \alpha_i, \dot{\alpha}_i]$, for $i = 1, 2, \cdots, n$. For each new input vector ξ_i, in conventional RBF layer, we build a m^4-dot conventional RBF NN. The mathematical manipulation layers consist of addition layer and multiplication layer. In the output layer, there are m^4 weights for each conventional RBF NN. Thus, there are much less nodes and weights to be built and updated (nm^4 nodes and n^2m^4 weights). The number n denoting DOF is successfully dropped from power side to multiplication side, which avoids the exponential increasing of the number of nodes and weights with DOF.

Similar to (27), the output of DCRBF NN in γ dimension can be formulated as

$$\begin{aligned} W_{k,\gamma}^T S'(Z) = &\sum_{k_1=1}^{m^4} \omega_{k_1,\gamma} f_{k_1} (\sum_{k_2=1}^{m^4} f_{k_2} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ &+ \sum_{k_2=1}^{m^4} \omega_{k_2,\gamma} f_{k_2} (\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ &+ \cdots + \sum_{k_n=1}^{m^4} \omega_{k_n,\gamma} f_{k_n} (\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}}) \end{aligned} \tag{34}$$

where

$$\begin{aligned} W_{k,\gamma} =& [\omega_{k_1=1,\gamma}, \omega_{k_1=2,\gamma}, \cdots, \omega_{k_1=m^4,\gamma}, \omega_{k_2=1,\gamma}, \omega_{k_2=2,\gamma}, \cdots \\ &, \omega_{k_2=m^4,\gamma}, \cdots, \omega_{k_n=1,\gamma}, \omega_{k_n=2,\gamma}, \cdots, \omega_{k_n=m^4,\gamma}]^T \end{aligned} \tag{35}$$

and

$$S'(Z) = \begin{bmatrix} f_{k_1=1}(\sum_{k_2=1}^{m^4} f_{k_2} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ f_{k_1=2}(\sum_{k_2=1}^{m^4} f_{k_2} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ \vdots \\ f_{k_1=m^4}(\sum_{k_2=1}^{m^4} f_{k_2} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ f_{k_2=1}(\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ f_{k_2=2}(\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ \vdots \\ f_{k_2=m^4}(\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_3=1}^{m^4} f_{k_3} \cdots \sum_{k_n=1}^{m^4} f_{k_n}) \\ \vdots \\ f_{k_n=1}(\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}}) \\ f_{k_n=2}(\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}}) \\ \vdots \\ f_{k_n=m^4}(\sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}}) \end{bmatrix} \tag{36}$$

Comparing (36), (35) with (28), (29), using operator matrix A, for $\gamma = 1, 2, \cdots, n$, it can be seen that

$$AW_{k,\gamma} = W_\gamma \tag{37}$$

and for $W_k = [W_{k,1}^T, W_{k,2}^T, \cdots, W_{k,n}^T]$,

$$AW_k = W \tag{38}$$

$$A^T S(Z) = S'(Z) \tag{39}$$

3.2.4. Solution of W_k and Approximation Error

Considering a training procedure for the weights, according to lemma 2.5, use a least-square method to solve the weights W and we can obtain the optimal weights W_γ^*, for $\gamma = 1, 2, \cdots, n$.

$$W_\gamma^* = \phi(Z)^\dagger g_\gamma(n_i) \tag{40}$$

where n_i is the number of training sample and $g_\gamma(n_i)$ is $n_i \times 1$ desired response in dimension γ of $F(Z)$. $\phi(Z) = [S(Z_1)^T, S[(Z_2)^T, \cdots, S(Z_{n_i})^T]^T$ is a $n_i \times m^{4n}$ interpolating matrix Considering an error constant $m^{4n} \times m^{4n}$ matrix κ:

$$\kappa = E_1 - AA^\dagger \tag{41}$$

where E_1 is a $m^{4n} \times m^{4n}$ unit matrix. Considering (37) and (39), the least-square method solvation of W_k can be derived of

$$\phi A W_{k,\gamma} = g_\gamma(n_i) \tag{42}$$

where A is the compress matrix derived above and $\phi_k(Z) = \phi A = [S'(Z_1), S'(Z_2), \cdots, S'(Z_{n_i})]^T$ is a $n_i \times nm^4$ interpolating matrix for $S'(Z)$.The optimal solution of $W^*_{k,\gamma}$ is

$$\begin{aligned} W^*_{k,\gamma} &= A^\dagger \phi^\dagger g_\gamma(n_i) \\ &= A^\dagger W^*_\gamma \end{aligned} \tag{43}$$

The weights approaching error ϵ can be expressed as

$$\begin{aligned} \epsilon_\gamma &= W^*_\gamma - AW^*_{k,\gamma} \\ &= \kappa W^*_\gamma \end{aligned} \tag{44}$$

which is a constant for $\gamma = 1, 2, \cdots, n$. Using the expression of ϵ_γ, and the output approximation error $\mu_\gamma(Z)$ can be expressed as

$$\begin{aligned} \mu_\gamma(Z) &= W^{*T}_\gamma S(Z) - {W^*_{k,\gamma}}^T S'(Z) \\ &= W^{*T}_\gamma (E_1 - AA^\dagger)^T S(Z) \\ &= (\kappa W^*_\gamma)^T S(Z) \end{aligned} \tag{45}$$

According to lemma 2.2

$$|\mu_\gamma(Z)| = |\epsilon^T_\gamma S(Z)| \leq \bar{\mu}_\gamma \tag{46}$$

where $\bar{\mu}_\gamma = \| \epsilon^T_\gamma \| s$ is a positive constant.

It can be seen that the DCRBF can obtain the similar desired response as conventional RBF with much less weights and dots,and a transform from conventional RBF to DCRBF with a bounded error $\bar{\mu}_\gamma$ for $\gamma = 1, 2, \cdots, n$ is accessible by using the operator matrix A.

3.3. Stability Analysis

Considering the dynamics of robot in $\dot{V}_2$, applying DCRBF described in subsection B, we see that over a compact set Ω_Z

$$\begin{aligned} &(\hat{W}^T_k - \widetilde{W}^T_k) S'(Z) = W^{*T}_k S'(Z) = W^{*T} S(Z) - \mu(Z) \\ &= -J^T(x_1)f - C(x_1, x_2)x_2 - G(x_1) - M(x_1)\dot{\alpha} - \mu(Z) - \eta(Z) \end{aligned} \tag{47}$$

where $\widetilde{W}_k = \hat{W}_k - W^*_k$ and $\eta(Z)$ is the NN approximation error satisfying $|\eta(Z)| \leq \bar{\eta}$, with $\bar{\eta} > 0$ as an unknown constant. According to (46), $\mu(Z) = [\mu(Z)_1, \mu(Z)_2, \cdots, \mu(Z)_n]$ satisfy $|\mu(Z)| \leq \bar{\mu}$, where $\bar{\mu} = [\bar{\mu}_1, \bar{\mu}_2, \cdots, \bar{\mu}_n]$. We propose $\tau_1 = -\hat{W}_k S'(Z)$, where $\hat{W}_k S'(Z)$ is used to approximate $W^*_k S'(Z)$. The adaptive NN robot control law is designed as

$$\begin{aligned} \tau &= \tau_1 + \tau_2 \\ &= -(z^T)^\dagger \sum_{i=1}^{n} \frac{k^2_{ai} e_i z_i}{k^2_{ai} - x^2_{1i}} - k_2 z - \hat{W}_k S'(Z) \end{aligned} \tag{48}$$

Considering the following Lyapunov candidate function:

$$V_3 = V_2 + \frac{1}{2}\sum_{\gamma=1}^{n} \widetilde{W}_{k,\gamma}^T Q_\gamma^{-1} \widetilde{W}_{k,\gamma} \tag{49}$$

for $\gamma = 1, 2, \cdots, n$, where Q_γ are positive definitive matrices. The adaptive law is given as follow:

$$\dot{\hat{W}}_{k,\gamma} = Q_\gamma [S'(Z) z_\gamma - \theta_\gamma \hat{W}_{k,\gamma}] \tag{50}$$

where $\theta_\gamma > 0 (\gamma = 1, 2, \cdots, n)$ are gain constants. Differentiating V_3 with respect to time, yields

$$\dot{V}_3 = \dot{V}_2 + \sum_{\gamma=1}^{n} \widetilde{W}_{k,\gamma}^T Q_\gamma^{-1} \dot{\widetilde{W}}_{k,\gamma} \tag{51}$$

Substituting (48), (47) and (50) into (51), we obtain

$$\begin{aligned}
\dot{V}_3 = & -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2} - z^T k_2 z + z^T (\eta(Z) + \mu(Z) - \widetilde{W}_k S'(Z)) \\
& + \sum_{i=1}^{n} \frac{k_{ai}^2 e_i z_i}{k_{ai}^2 - x_{1i}^2} - z^T (z^T)^\dagger \sum_{i=1}^{n} \frac{k_{ai}^2 e_i z_i}{k_{ai}^2 - x_{1i}^2} \\
& + \sum_{\gamma=1}^{n} \widetilde{W}_{k,\gamma}^T S'(Z) z_\gamma - \sum_{\gamma=1}^{n} \widetilde{W}_{k,\gamma} \theta_\gamma \hat{W}_{k,\gamma}
\end{aligned} \tag{52}$$

when $z = [0, 0, \cdots, 0]^T$, $\dot{V}_2 = -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2}$. We can still draw the asymptotic stability of the system according to the lemma 2.3. In the case of $z \neq [0, 0, \cdots, 0]^T$, we have

$$\begin{aligned}
\dot{V}_3 \leq & -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2} - z^T k_2 z + z^T z + \frac{1}{2}(\| \bar{\mu} \|^2 + \| \bar{\eta} \|^2) \\
& - \sum_{\gamma=1}^{n} \widetilde{W}_{k,\gamma} \theta_\gamma \hat{W}_{k,\gamma} \\
\leq & -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{1i}^2} - z^T (k_2 - I) z + \frac{1}{2}(\| \bar{\mu} \|^2 + \| \bar{\eta} \|^2) \\
& - \sum_{\gamma=1}^{n} (\theta_\gamma - \frac{1}{2}\theta_\gamma^2) \| \widetilde{W}_{k,\gamma} \|^2 + \frac{1}{2}\sum_{\gamma=1}^{n} \| W_{k,\gamma}^* \|^2
\end{aligned} \tag{53}$$

Thus, considering Property 2 we can obtain

$$\dot{V}_3 \leq -pV_3 + C \tag{54}$$

where

$$p = min[k_{1i}, \frac{\lambda_{min}(2(k_2 - I))}{\lambda_{max}(M)}, \frac{2(\theta_\gamma - \frac{1}{2}\theta_\gamma^2)}{\lambda_{max}(Q_\gamma^{-1})}] \tag{55}$$

$$C = \frac{1}{2}(\parallel \bar{\mu} \parallel^2 + \parallel \bar{\eta} \parallel^2) + \frac{1}{2}\sum_{\gamma=1}^{n} \parallel W_{k,\gamma}^* \parallel^2 \tag{56}$$

where $\lambda_{min}(\bullet)$ and $\lambda_{max}(\bullet)$ denote the minimum and maximum eigenvalues of matrix $\bullet$. To ensure $p > 0$, k_2 and θ_γ must satisfy the following conditions

$$\lambda_{min}(2(k_2 - I)) > 0 \tag{57}$$

$$0 < \theta_\gamma < \sqrt{2} \tag{58}$$

If C can be zero, the system can be said to achieve the exponential stability. However, considering the approximation error of NN, for our controller, $c = \frac{\|\bar{\mu}\|^2\|\bar{\eta}\|^2}{2}$ which is a positive constant. Thus, the system can only be obtained stable instead of exponential stability.

Theorem 3.1. *According to Property 2, we can know that $x_{1i} = e_i + x_{di} < k_{ai}$ is bounded. For the condition satisfies $-k_{ai} < -k_{c1i} \leq x_{1i} \leq k_{c1i} < k_{ai}$, according to (6) and (7), we have $-(k_{ai} - k_{c1i}) < e_i < (k_{ai} - k_{c1i})$, which is bounded. Then considering the definition of α in (11) and Property 4, α is bounded too. According to (54), lemma 2.1, Property 3 and in terms of (7), (13) and (49), we can safely conclude the e, z and NN weight estimated error $\widetilde{W}_k$ are bounded. In the term of the boundness of α and z, according to $x_2 = z + \alpha$, x_2 is bounded. Thus, we can safely say that the signals of the closed-loop system are semiglobally uniformly bounded (SGUB). And the closed-loop error signals e and z will remain within the compact sets Ω_e, Ω_z, respectively, defined by*

$$\Omega_e : \ = \{e \in R^n \mid |e_i| \leq \sqrt{(k_{ai} - k_{c1i})^2(1 - e^{-D})}\} \tag{59}$$

$$\Omega_z : \ = \{z \in R^n \mid ||z|| \leq \sqrt{\tfrac{D}{\lambda_{max}(M)}}\} \tag{60}$$

where $i = 1, 2, \cdots, n$ and $D = 2(V_3(0) + C/p)$, p and C are two positive constants.

Remark 1. The designed parameter k_{ai} in the controller can be chosen simply as positive and the matrix k_2 should satisfy the condition in (57). The gains in NN adaptive law Q_γ and θ_γ should be positive. According to (58), θ_γ should also be smaller than $\sqrt{2}$. In terms of (55),(59),(60), if the parameters k_{ai}, k_2 and θ_γ are chosen to be relatively small, while Q_γ chosen relatively large, then the amplitude of trucking error could be made smaller.

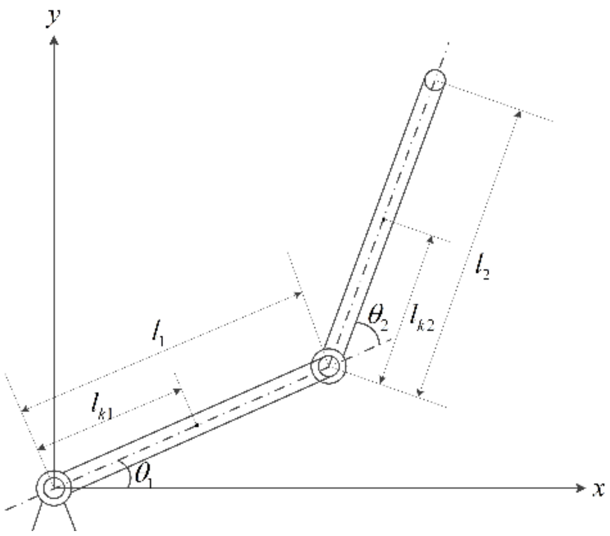


Figure 3. Structure of the system

4. Simulation Studies

In order to test the validity of the control, we have done a simulation on 2-DOF robot manipulator which have two revolute joints in the vertical plane. In the model which is shown Figure 3, the manipulator material is uniform. We define m_i, l_i, l_{ki}, I_i as the mass, the length of link i, the center distance of link i and the inertia of link i, $i = 1, 2$. $q = [q1, q2]$. Other simulation parameters are shown in Table. 1.

According to the method of (Craig, 2005), we can get the dynamics parameters of the robot as follows:

$$G(q) = \begin{bmatrix} (m_1 l_{k2} + m_2 l_1) g \cos q_1 \\ m_2 l_{k2} g \cos(q_1 + q_2) \end{bmatrix} \tag{61}$$

$$M(q) = \begin{bmatrix} m_1 l_{k1}^2 + I_1 + I_2 + m_2(l_1^2 + l_{k2}^2 + 2 l_1 l_{k2} \cos q_2) & m_2(l_{k2}^2 + l_1 l_{k2} \cos q_2) + I_2 \\ m_2(l_{k2}^2 + l_1 l_{k2} \cos q_2) + I_2 & m_2 l_{k2}^2 + I_2 \end{bmatrix} \tag{62}$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_2 l_1 l_{k2} \dot{q}_2 \sin q_2 & -m_2 l_1 l_{k2} (\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_2 l_1 l_{k2} \dot{q}_1 \sin q_2 & m_2 l_1 l_{k2} \dot{q}_1 \sin q_2 \end{bmatrix} \tag{63}$$

$$J(q) = \begin{bmatrix} -l_1 \sin q_1 - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix} \tag{64}$$

We choose the desired trajectory as $x_d = [\sin(0.5t), 2cos(0.5t)]$, where $t \in [0, T]$ and $T = 30s$. To maintain the constraints $|x_{1i}| \leq k_{c1i} < k_{ai}, |x_{di}| \leq k_{c1i} < k_{ai}$, we design $k_a = [1.1, 2.1]^T$ and $f = [2\cos(t) + 0.5 + d(t), \sin(t) + 1 + d(t)]$, where $d(t)$ is a

Table 1. Simulation Parameters.

Description	Parameter	Value	Unit
Mass of Link 1	m_1	2.5	kg
Mass of Link 2	m_2	1.2	kg
Moment of inertia of Link 1	I_1	351.56×10^{-3}	kgm^2
Moment of inertia of Link 2	I_2	60.75×10^{-3}	kgm^2
Length of Link 1	l_1	0.75	m
Length of Link 2	l_2	0.45	m

white Gaussian noise. To satisfy condition (57), k_2 is chosen as $[65, 15]$. k_1 is chosen as $[35, 6]$. We choose initial conditions are given as $x_1(0) = [0, 2]^T$ and $x_2(0) = [0, 0]^T$. In the simulation, we studied two different cases. The conventional RBF with 256 nodes and the proposed DCRBF with 32 nodes. The approaching error between W and W_k are examined with the calculated error ϵ_γ for $\gamma = 1, 2$, which consists of calculated constant matrix κ. For the DCRBF, the centres are chosen in the area of $[-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1]$ for each dimension, which constitute $f_{k_1=1}, f_{k_1=2}, \cdots, f_{k_1=2^4}$ and $f_{k_2=1}, f_{k_2=2}, \cdots, f_{k_2=2^4}$ all 32 nodes. For the conventional RBF, the centres are chosen in the area of $[-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1]$ arraying like

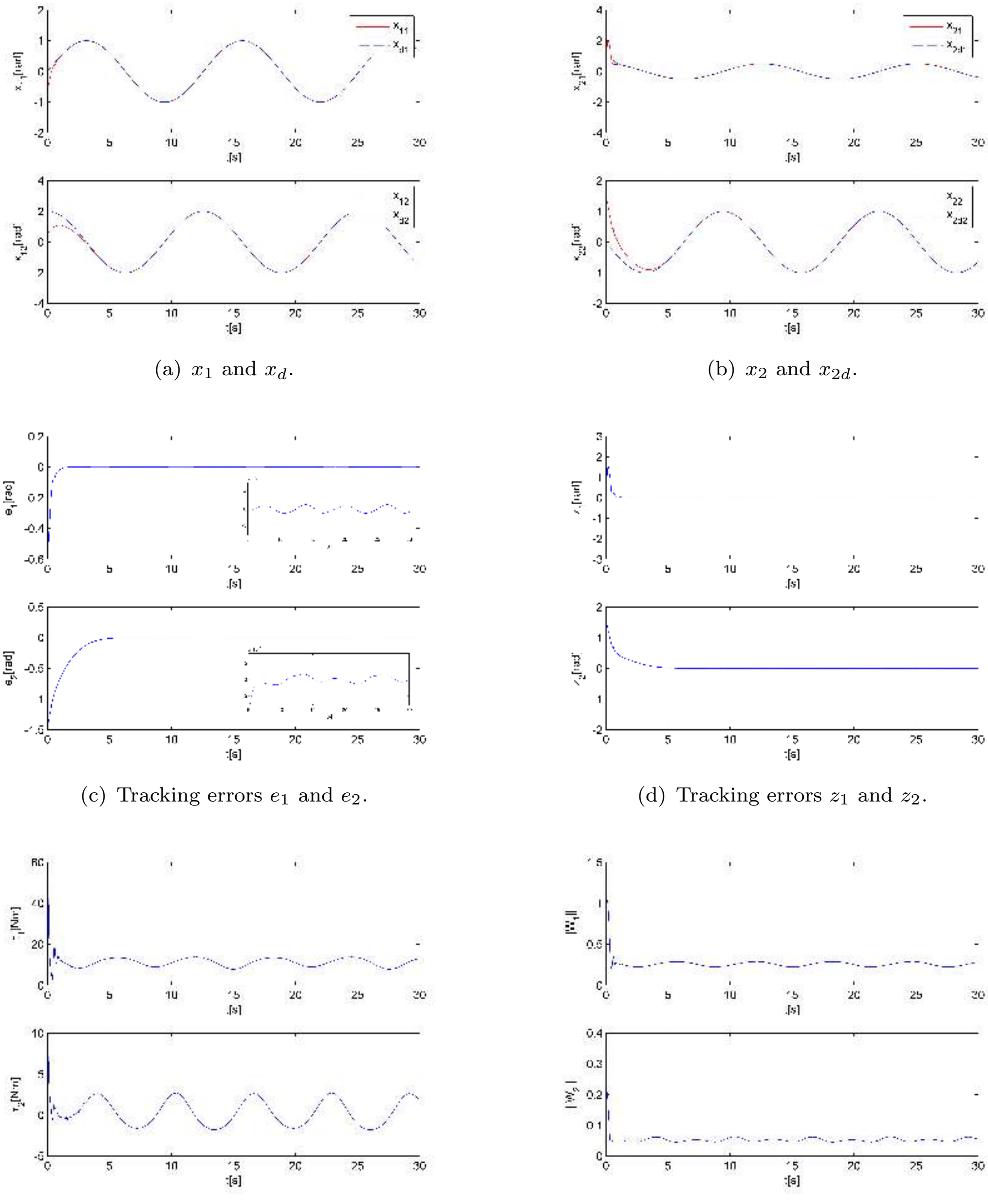
$$\begin{aligned} S(Z) = [&f_{k_1=1}f_{k_2=1}, f_{k_1=2}f_{k_2=1}, \cdots, f_{k_1=2^4}f_{k_2=1}, \\ &f_{k_1=1}f_{k_2=2}, f_{k_1=2}f_{k_2=2}, \cdots, f_{k_1=2^4}f_{k_2=2}, \cdots, \\ &f_{k_1=1}f_{k_2=2^4}, f_{k_1=2}f_{k_2=2^4}, \cdots, f_{k_1=2^4}f_{k_2=2^4}]. \end{aligned} \tag{65}$$

Thus, the compression matrix A is designed as (33), where E is a 16×16 unit matrix and ψ_i for $i = 1, 2, \cdots, n$ are 16×16 submatrices.

The results of the simulation are shown in Figure 4-Figure 6. All the pictures in Figure 4 represent the results of DCRBF with iBLF. Pictures in Figure 5 denote the results of conventional RBF with iBLF and Figure 6 is graphed to display the approximating error for weights between DCRBF and conventional RBF. Specially, the small pictures inserted in Figure 4(c) and Figure 5(c) represent the magnified position tracking errors from time 5s - 30s.

In these pictures, we know that the prescribed trajectory tracking performance of proposed DCRBF with the implementation of iBLF is satisfactory from Figure 4(a) and Figure 4(b). The system errors shown from Figure 4(c) and Figure 4(d) are converging to a small value which is close to zero. And comparing Figure 4(a)-Figure 4(e) with Figure 5(a)-Figure 5(e), we can conclude that the DCRBF can approximate the system uncertainties as well as the conventional RBF.

According to Figure 4(f), Figure 5(f), W_k and W approach to be stable with the test time going. So we assume that neural network approaches to the ideal model W_k^*, W^* when times comes to 30s. Then we graph Figure 6 to show the approaching error between the weights of DCRBF and the conventional RBF, which is formulated as $\Delta W_\gamma = W_\gamma^* - AW_{k,\gamma}^* - \epsilon_\gamma$, for $\gamma = 1, 2$, where $\epsilon_\gamma = \kappa W_\gamma^*$ and $\kappa = E_1 - AA^\dagger$. The results are very small values close to zero, which proves that the approaching error of weights is definitely as what we calculate in (44). Then according to (45) and (46), we know that the output approximation error $W^*S(Z) - W^*S'(Z)$ is bounded. All the arguments above show that in terms of fitting ability, DCRBF can obtain a similar performance to conventional RBF with a bounded approximation error.



(a) x_1 and x_d.

(b) x_2 and x_{2d}.

(c) Tracking errors e_1 and e_2.

(d) Tracking errors z_1 and z_2.

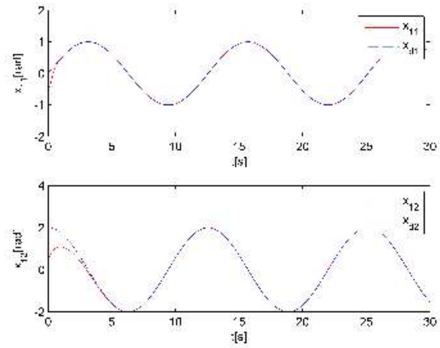
(e) τ_1 and τ_2.

(f) $||W_{k,1}||$ and $||W_{k,2}||$

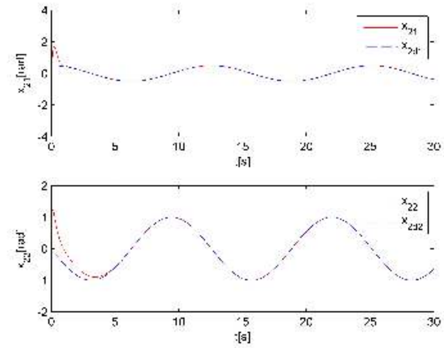
Figure 4. The results of DCRBF with IBLF.

5. Conclusion

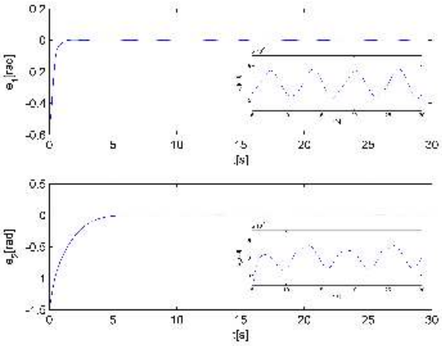
This paper presents an innovative adaptive neural network control using DCRBF for n-DOF robot system with full-state constraints and unknown dynamics. By utilizing DCRBF, the problem of superfluous number of nodes and weights in conventional RBF is overcome without compromising the tracking performance. The rigorous mathematical proofs of the effectiveness of DCRBF have been denoted. And an adaptive control for the system is formulated based on the methods of iBLF and backstepping for tracking performance and stability of the system with constraints and unknown disturbance. Finally, the effectiveness of the proposed method we proposed has been



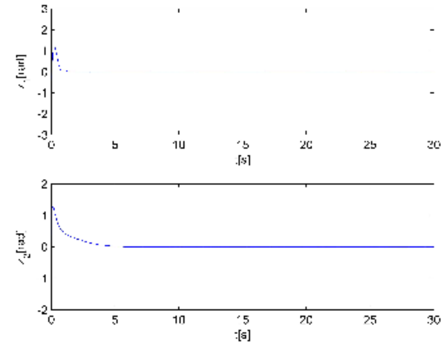
(a) x_1 and x_d.



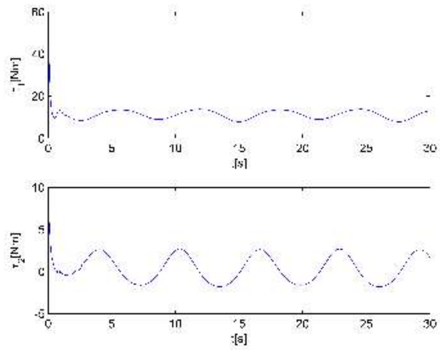
(b) x_2 and x_{2d}.



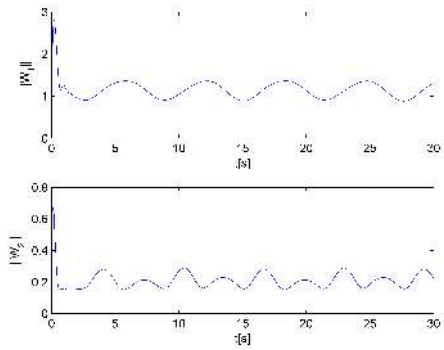
(c) Tracking errors e_1 and e_2.



(d) Tracking errors z_1 and z_2.



(e) τ_1 and τ_2.



(f) $||W_1||$ and $||W_2||$

Figure 5. The results of conventional RBF with IBLF.

demonstrated through the results presented in this paper.

Acknowledgement

The authors would like to thank Dr. Yiming Jiang of Key Lab of Autonomous Systems and Networked Control, South China University of Technology for the technical support provided.

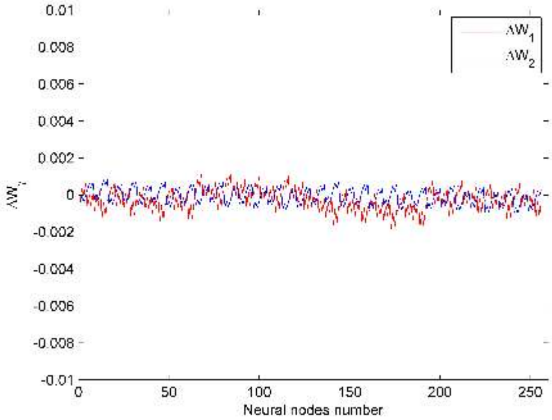


Figure 6. Weights approaching error between DCRBF and conventional RBF

References

Alford, C., & Belyeu, S. (1984). Coordinated control of two robot arms. In *Robotics and automation. proceedings. 1984 ieee international conference on* (Vol. 1, pp. 468–473).

Arefi, M. M., & Jahed-Motlagh, M. R. (2013). Observer-based adaptive neural control of uncertain mimo nonlinear systems with unknown control direction. *International Journal of Adaptive Control and Signal Processing*, *27*(9), 741–754.

Arefi, M. M., Jahed-Motlagh, M. R., & Karimi, H. R. (2015). Adaptive neural stabilizing controller for a class of mismatched uncertain nonlinear systems by state and output feedback. *IEEE transactions on cybernetics*, *45*(8), 1587–1596.

Berkovitz, L. D. (2013). *Optimal control theory* (Vol. 12). Springer Science & Business Media.

Chen, M., & Ge, S. S. (2013). Direct adaptive neural control for a class of uncertain nonaffine nonlinear systems based on disturbance observer. *IEEE Transactions on Cybernetics*, *43*(4), 1213–1225.

Cheng, L., Cheng, M., Yu, H., Deng, L., & Hou, Z.-G. (2016). Distributed tracking control of uncertain multiple manipulators under switching topologies using neural networks. In *International symposium on neural networks* (pp. 233–241).

Cheng, L., Hou, Z.-G., Tan, M., & Zhang, W.-J. (2012). Tracking control of a closed-chain five-bar robot with two degrees of freedom by integration of an approximation-based approach and mechanical design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(5), 1470–1479.

Cheng, L., Liu, W., Hou, Z.-G., Yu, J., & Tan, M. (2015). Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Transactions on Industrial Electronics*, *62*(12), 7717–7727.

Craig, J. J. (2005). *Introduction to robotics: mechanics and control* (Vol. 3). Pearson Prentice Hall Upper Saddle River.

Dai, S.-L., Wang, C., & Wang, M. (2014). Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems. *IEEE Transactions on Neural Networks and learning systems*, *25*(1), 111–123.

Gao, W., & Selmic, R. R. (2006). Neural network control of a class of nonlinear systems with actuator saturation. *IEEE Transactions on Neural Networks*, *17*(1), 147–156.

Gueaieb, W., Karray, F., & Al-Sharhan, S. (2007). A robust hybrid intelligent position/force control scheme for cooperative manipulators. *IEEE/ASME Transactions on Mechatronics*, *12*(2), 109–125.

Haykin, S. S., Haykin, S. S., Haykin, S. S., & Haykin, S. S. (2009). *Neural networks and learning machines* (Vol. 3). Pearson Upper Saddle River, NJ, USA:.

He, W., Chen, Y., & Yin, Z. (2016). Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE Transactions on Cybernetics*, *46*(3), 620–629.

He, W., Zhang, S., Ge, S. S., & Liu, C. (2014). Adaptive boundary control for a class of inhomogeneous timoshenko beam equations with constraints. *IET Control Theory & Applications*, *8*(14), 1285–1292.

Hou, Z.-G. (2001). A hierarchical optimization neural network for large-scale dynamic systems. *Automatica*, *37*(12), 1931–1940.

Huang, L., Ge, S., & Lee, T. (2006). Position/force control of uncertain constrained flexible joint robots. *Mechatronics*, *16*(2), 111–120.

Huang, Y., Na, J., Wu, X., Liu, X., & Guo, Y. (2015). Adaptive control of nonlinear uncertain active suspension systems with prescribed performance. *ISA transactions*, *54*, 145–155.

Kennedy, M. P., & Chua, L. O. (1988). Neural networks for nonlinear programming. *IEEE Transactions on Circuits and Systems*, *35*(5), 554–562.

Lee, G.-W., & Cheng, F.-T. (1996). Robust control of manipulators using the computed torque plus h compensation method. *IEE Proceedings-Control Theory and Applications*, *143*(1), 64–72.

Lee, T., Koh, E., & Loh, M. (1996). Stable adaptive control of multivariable servomechanisms, with application to a passive line-of-sight stabilization system. *IEEE Transactions on Industrial Electronics*, *43*(1), 98–105.

Lee, T. H., & Harris, C. J. (1998). *Adaptive neural network control of robotic manipulators* (Vol. 19). World Scientific.

Li, T., Duan, S., Liu, J., Wang, L., & Huang, T. (2016). A spintronic memristor-based neural network with radial basis function for robotic manipulator control implementation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*(4), 582–588.

Li, Z., Ge, S. S., Adams, M., & Wijesoma, W. S. (2008). Robust adaptive control of uncertain force/motion constrained nonholonomic mobile manipulators. *Automatica*, *44*(3), 776–784.

Li, Z., Ge, S. S., & Ming, A. (2007). Adaptive robust motion/force control of holonomic-constrained nonholonomic mobile manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *37*(3), 607–616.

Li, Z., Li, T., & Feng, G. (2016). Adaptive neural control for a class of stochastic nonlinear time-delay systems with unknown dead zone using dynamic surface technique. *International Journal of Robust and Nonlinear Control*, *26*(4), 759–781.

Liu, Y.-J., & Tong, S. (2017). Barrier lyapunov functions for nussbaum gain adaptive control of full state constrained nonlinear systems. *Automatica*, *76*, 143–152.

Luo, B., Wu, H.-N., & Li, H.-X. (2015). Adaptive optimal control of highly dissipative nonlinear spatially distributed processes with neuro-dynamic programming. *IEEE transactions on neural networks and learning systems*, *26*(4), 684–696.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814.

Na, J., Mahyuddin, M. N., Herrmann, G., Ren, X., & Barber, P. (2015). Robust adaptive finite-time parameter estimation and control for robotic systems. *International Journal of Robust and Nonlinear Control*, *25*(16), 3045–3071.

Namvar, M., & Aghili, F. (2005). Adaptive force-motion control of coordinated robots interacting with geometrically unknown environments. *IEEE Transactions on Robotics*, *21*(4), 678–694.

Rubio, J. (2012). Modified optimal control with a back propagation network for robotic arms. *IET Control Theory & Applications*, *6*(14), 2216–2225.

Su, C.-Y., Leung, T.-P., & Zhou, Q.-J. (1992). Force/motion control of constrained robots using sliding mode. *IEEE Transactions on Automatic Control*, *37*(5), 668–672.

Tang, Z.-L., Ge, S. S., Tee, K. P., & He, W. (2016a). Adaptive neural control for an uncertain robotic manipulator with joint space constraints. *International Journal of Control*, *89*(7), 1428–1446.

Tang, Z.-L., Ge, S. S., Tee, K. P., & He, W. (2016b). Robust adaptive neural tracking control for a class of perturbed uncertain nonlinear systems with state constraints. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*(12), 1618–1629.

Tee, K. P., Ren, B., & Ge, S. S. (2011). Control of nonlinear systems with time-varying output constraints. *Automatica*, *47*(11), 2511–2516.

Wang, M., & Yang, A. (2017). Dynamic learning from adaptive neural control of robot manipulators with prescribed performance. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Yang, C., Jiang, Y., Li, Z., He, W., & Su, C.-Y. (2017). Neural control of bimanual robots with guaranteed global stability and motion precision. *IEEE Transactions on Industrial Informatics*.

Yang, C., Wang, X., Cheng, L., & Ma, H. (2016). Neural-learning-based telerobot control with guaranteed performance. *IEEE transactions on cybernetics*.

Yang, G.-H., & Wang, J. L. (2001). Robust nonfragile kalman filtering for uncertain linear systems with estimator gain uncertainty. *IEEE Transactions on Automatic Control*, *46*(2), 343–348.

Yang, G.-H., & Ye, D. (2006). Adaptive fault-tolerant h/spl sub/spl infin//control via state feedback for linear systems against actuator faults. In *Decision and control, 2006 45th ieee conference on* (pp. 3530–3535).

Zhang, T., & Ge, S. S. (2009). Adaptive neural network tracking control of mimo nonlinear systems with unknown dead zones and control directions. *IEEE Transactions on Neural Networks*, *20*(3), 483–497.

Zhang, T., Ge, S. S., & Hang, C. C. (2000). Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. *Automatica*, *36*(12), 1835–1846.