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

aSun Yat-sen University, Guangzhou, People’s Republic of China; bCity University of Hong Kong, Kowloon Tong, Hong Kong; cBristol Robotics Laboratory, University of the West of England, Bristol, UK; dSouth China University of Technology, Guangzhou, People’s Republic of China; eUniversity of Winchester, Hampshire, UK

ABSTRACT

Conventional Neural Network (NN) control for robots uses 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, the 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.

1. Introduction

With the wide use of complex robot manipulators, which are nonlinear systems in our modern society, research into robot technologies has attracted enormous attention (Alford & Belyeu, 1984; Cheng, Hou, Tan, & Zhang, 2012; Gueaieb, Karray, & Al-Sharhan, 2007; Lee & Cheng, 1996; Li, Duan, Liu, Wang, & Huang, 2016; Na, Mahyuddin, Herrmann, Ren, & Barber, 2015; Namvar & Aghili, 2005). Meanwhile, the control of robotic systems in the presence of uncertain parameters and motion constraints has been extensively studied (Li, Ge, & Ming, 2007; Tang, Ge, 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; Huang, Na, Wu, Liu, & Guo, 2015; Lee, Koh, & Loh, 1996; 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; Li, Li, & Feng, 2016). It is very difficult to establish exact mathematical dynamic models with various uncertainties, e.g. unknown payloads (Arefi & Jahed-Motlagh, 2013; Arefi, Jahed-Motlagh, & Karimi, 2015; Yang & Wang, 2001; Zhang & Ge, 2009). However, by exploiting NN approximation, many complex and challenging models can be established more easily while not sacrificing the characteristics of accurate models (Chen & Ge, 2013; Dai, Wang, & Wang, 2014; Gao & Selmic, 2006; Lee & Harris, 1998; Li, Adams, & Wijesoma, 2008). In Yang, Wang, Cheng, and Ma (2017), a direct adaptive NN scheme is presented for a class of uncertain nonlinear strict-feedback systems. By utilising 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 system using a multilayer NN was studied in Zhang, Ge, and Hang (2000) so as to guarantee the uniform ultimate boundedness of the closed-loop adaptive systems. In Yu, Xie, Paszczynski, and Wilamowski (2011), RBF (radial basis function) NN proved to be easier to train, simpler than other NN structures, resilient to input noise, and has a good online learning ability. Thus RBF NN is widely used in the field of adaptive NN control. Tsai, Huang, and Lin (2010) used RBF NN on a self-balancing scooter with unknown plant...
parameters, unknown frictions and linearised errors. Fei and Ding (2012) combined RBF NN with adaptive sliding mode control to deal with uncertainties and unknown disturbance in a time varying system. In Yang, Jiang, Li, He, and Su (2017), RBF NN based control for coordinated dual arms robots has been proposed to settle the uncertainties. He, Chen, and Yin (2016) utilised NN of the conventional RBF NN structure for an n-link robot but with lots of nodes. However, from the lectures (Haykin, Haykin, & Haykin, 2009; Yang et al., 2017), we know that the node and weight matrices grow exponentially w.r.t. the degrees of freedom (DOF). As the DOF increase, the number of nodes will increase exponentially, to the point where the computing resources cannot cope. For example, when the DOF of a system is n, the number of nodes of the system is \( n^m \), m is the number of centres of each DOF. According to the literature, in order to make the network as accurate as possible, m is always greater than or equal to 2. When n is small, it will not take a long time for each control step (each control step is relatively short), but when n is much higher, it will consume a lot of time and energy or even make the program fail.

In this paper, an innovative DCRBF NN is proposed. The n-DOF input is split into n 1-DOF inputs and a conventional RBF NN is built for each 1-DOF input. Subsequent mathematical manipulations provide the output of every node in each NN (by adding two layers) ensuring that the tracking performance of the controller output does not deteriorate. In this way, we can compress the number of nodes and weights to an extreme degree with performance similar to the conventional RBF NN, 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 stabilisation of the system is another important requirement of controller design. In practical systems, violation of constraints may cause the degeneration of the control performance or even system failures (Huang, Ge, & Lee, 2006; Su, Leung, & Zhou, 1992; Tee, Ren, & Ge, 2011). To deal with the constraint problem, many methods have been proposed such as model predictive control (MPC), optimal control, reference governors and so on. It is always necessary to know the exact model which is quite complicated for complex robots when utilising the methods, model predictive control and optimal control (Berkovitz, 2013; Luo, Wu, & Li, 2015; Mayne, Rawlings, Rao, & Scokaert, 2000; Rubio, 2012). However, in a situation with uncertainties or some sophisticated model which cannot be calculated, these control methods cannot be employed and it is necessary to find some alternative solutions.

To solve the problem of system control in the presence of constraints and uncertainties, Barrier Lyapunov Functions (BLF) are a popular solution because 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; a BLF is designed to guarantee the uniform ultimate boundedness of the closed-loop system. In Tee et al. (2011), BLF are employed at the outset to prevent the output from violating the time-varying constraint in strict feedback nonlinear systems. However, BLF-based controls have their own limitations. One such limitation is that the feasibility conditions have a tendency to be conservative when ensuring constraint satisfaction due to the original state being indirectly enforced by imposing transformed constraints on the errors. In He, Zhang, Ge, and Liu (2014), iBLF (integral Barrier Lyapunov Function) based boundary controls were proposed for a class of inhomogeneous Timoshenko beam satisfying the needs of suppressing undesirable vibrations and preventing constraint violation. In Tang, Ge, Tee, and 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 contributions of this paper are as follows:

- An innovative DCRBF NN is proposed to avoid the exponential growth of nodes and weights as the DOF increases. Additionally, DCRBF NN 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 in simulation.
- In order to avoid the violation of constraints while using BLF on n-link robots, a novel iBLF is utilised to design the control strategy which incorporates the output constraints and provides enhanced system stability.

The rest of this paper is organised as follows. Section 2 gives the problem formulation of the n-link robot manipulator and some useful preliminaries for deriving the proof. In Section 3, the control design and the stability analysis for the system are proposed. 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 4.
2. Problem formulation and preliminaries

2.1. System description

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

\[ M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau(t) - J^T f(t), \]

where \( q, \dot{q}, \ddot{q} \in \mathbb{R}^n \) represents 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 the 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

\[ \dot{x}_1 = x_2, \]
\[ \dot{x}_2 = M^{-1}(x_1)[\tau - C(x_1, x_2) - G(x_1) - J^T(x_1)f(t)]. \]

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 \( x_d(t) = [q_{d1}(t), q_{d2}(t), \ldots, q_{dn}(t)]^T \) and the desired trajectory is \( \alpha(t) = [\alpha_1(t), \alpha_2(t), \ldots, \alpha_n(t)] \). Assuming all signals and state constraints are bounded, we have constants \( k_{c1} \), such that \( -k_{c1} \leq x_1(t) \leq k_{c1}, \forall \ t \geq 0 \), with \( k_{c1} = [k_{c11}, k_{c12}, \ldots, k_{c1n}]^T \) are positive constant vectors.

2.2. Required technology lemmas and definitions

**Lemma 1 (He et al., 2016):** If there exists 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 \( \xi_1, \xi_2 \) are the functions making \( \mathbb{R}^n \rightarrow \mathbb{R} \), the parameters and states of the system will remain in a compact set and eventually converge to specific compact sets.

**Lemma 2:** For the adaptive law (50), there exists a compact set

\[ \Omega_w = \left\{ \hat{W}_{k,y} \mid \|\hat{W}_{k,y}\| \leq nm^2 \frac{s}{\theta_y} \right\}, \]

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

**Proof:** For the conventional RBF NN, according to the nature of Gaussian 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 \( i = 1, 2, \ldots, nm^4, \|S'(Z_i)\| \leq \sqrt{ns}. \) Thus for \( S'(Z) = [S'(z_1), S'(z_2), \ldots, S'(z_{nm^4})] \), we have \( ||S'(Z)|| \leq nm^2 s \) and according to Craig (2005), we can obtain (3).

**Lemma 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 \( f(x) \) is uniformly continuous, then \( \lim_{t \to \infty} \dot{f}(t) = \alpha. \)

**Lemma 4 (RBF approximation Wang & Yang, 2017):** If there are sufficient nodes, under suitable width \( \Delta \) and node centres \( \hat{a} \), RBF NN can approximate any smooth function \( f_a(x) \) over a compact set \( x \in \Omega_x \) with convergent errors: \( f_a(x) = W^* S(x) + \eta(x) \) where \( W^* \) is the ideal weight matrix, \( \eta(x) \) is the convergent errors, satisfying \( \|\eta(x)\| \leq \hat{\eta}, \hat{\eta} \) is the constant vector.

**Lemma 5 (RBF and optimal weights):** According to Haykin et al. (2009), without loss of generality, we can use the least-squares method and recursive least-squares method to solve for the optimal weights. The two methods are mathematically equivalent. For the least-squares method, we have

\[ W = \phi(Z)^\dagger g(n_i), \]

where \( n_i \) is the size of the training sample and \( g(n_i) \) is \( n_i \times 1 \) desired response. \( \dagger \) is the symbol for generalised inverse. \( \phi(Z) = [S(Z_1)^T, S(Z_2)^T, \ldots, S(Z_{n_i})^T]^T \) is an interpolating matrix. As for using recursive least-square method, we have

\[ W(n_i) = [\phi(n_i)^T \phi(n_i)]^{-1} S(n_i) d(n_i) + W(n_i - 1), \]

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

3. Control design

3.1. iBLF design

3.1.1. iBLF design

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

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

\[
|x_{1i}| \leq k_{ci1} < k_{ai}, \quad |x_{di}| \leq k_{ci1} < k_{ai} \tag{6}
\]

for all \( i = 1, 2, \ldots, n \).

For the \( n \)-link robotic arm, consider an iBLF candidate

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

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

Differentiating \( V_1 \) with respect to time, we have

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

where

\[
\rho_i = \int_0^1 \frac{k_{ai}^2}{k_{ai}^2 - (\beta e_i + x_{di})^2} \, d\beta = \frac{k_{ai}}{2e_i} \ln \left( \frac{(k_{ai} + e_i + x_{di})(k_{ai} - x_{di})}{(k_{ai} - e_i - x_{di})(k_{ai} + x_{di})} \right). \tag{9}
\]

Then, a virtual controller \( \alpha_i \) can be designed as

\[
\alpha_i = \left( -k_{1i} e_i + \frac{(k_{ai}^2 - x_{di}^2) \dot{x}_{di}}{k_{ai}^2} \right), \quad i = 1, 2, \ldots, n,
\]

where \( k_{1i} \) is a positive control gain for \( i = 1, 2, \ldots, n \), we obtain

\[
\dot{V}_1 = -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{di}^2} + \sum_{i=1}^{n} \frac{k_{ai}^2 e_i z_i}{k_{ai}^2 - x_{di}^2} + z^T \tau - \dot{z}^T \dot{x}_1 f
\]

\[
- C(x_1, x_2) x_2 - G(x_1) - M(x_1) \ddot{u} \]. \tag{10}

According to the expression of \( \dot{V}_1 \), we define the control law as \( \tau = \tau_1 + \tau_2 \), where according to lemma 4, \( \tau_1 \) uses adaptive dimension compressed RBF NN control as described in subsection 3.2.3 and \( \tau_2 \) is defined as

\[
\tau_2 = -(z^T)^n \sum_{i=1}^{n} \frac{k_{ai}^2 e_i z_i}{k_{ai}^2 - x_{di}^2} - k_{2i} z,
\]

where \( k_{2i} \) is the control gain, and \( k_{2i}, i = 1, 2, \ldots, n \), are positive constants. Then substituting \( \tau \) into (14), according to Moore–Penrose inverse, \( \dot{V}_2 = -\sum_{i=1}^{n} (k_{ai}^2 k_{1i} e_i^2 / k_{ai}^2 - x_{di}^2) \), when \( z = [0, 0, \ldots, 0]^T \). According to Lemma 3, we can still draw the asymptotic stability of the system. Otherwise, in the case of \( z \neq [0, 0, \ldots, 0]^T \), we obtain

\[
\dot{V}_2 = -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{1i} e_i^2}{k_{ai}^2 - x_{di}^2} - z^T k_{2i} z + z^T \tau_1 - \dot{f}
\]

\[
- C(x_1, x_2) x_2 - G(x_1) - M(x_1) \ddot{u} \]. \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 \( i = 1, 2, \ldots, n \):

\[
V_1 = \sum_{i=1}^{n} \int_0^{e_i} \frac{k_{ai}^2}{k_{ai}^2 - (\sigma + x_{di})^2} \, d\sigma \leq \sum_{i=1}^{n} \frac{k_{ai}^2 e_i^2}{k_{ai}^2 - (e_i + x_{di})^2}. \tag{17}
\]

**Proof:** Denote \( p(\sigma, x_{di}) = (k_{ai}^2) / (k_{ai}^2 - (\sigma + x_{di})^2) \). 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 \( \sigma \) in the set \( |\sigma + x_{di}| < k_{ai} \), we can obviously see that

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

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

**Property 3:** By Assumption 1, \( V_1 \) is positive definite, continuously differentiable and satisfies the decrescent condition in the set \( |x_{1i}| \leq k_{ci1} < k_{ai} \), for \( i = 1, 2, \ldots, n \):

\[
\sum_{i=1}^{n} \frac{e_i^2}{2} \leq V_1 \leq \sum_{i=1}^{n} \frac{e_i^2}{2} \int_0^{e_i} \frac{\beta k_{ai}^2}{k_{ai}^2 - (e_i \beta + \text{sgn}(e_i) k_{ci1})^2} \, d\beta, \tag{20}
\]

which is useful for establishing uniform stability.
**Property 4:** Using L'Hôpital's rule, it can be shown that

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

(21)

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

### 3.2. NN design

#### 3.2.1. Dimension split for radial basis function

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

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

$$
F(Z) = W^T S(Z),
$$

(23)

where $W = [W_1, W_2, \ldots, W_n]$ are the weights of the NNs, $S(Z) = [S(Z)_1, S(Z)_2, \ldots, S(Z)_n]^T$ are the basis functions. Without loss of generality, we choose $S(Z)_i$, where $i = 1, 2, \ldots, m^4$, and expand it. So we can obtain

$$
S(Z)_i = e^{-\|Z - \tilde{\delta}_i\|^2 / \Delta^2}
$$

$$
= e^{-(\sum_{j=1}^{m^n}(e_j - \delta_{ij})^2 + \sum_{j=n+1}^{m^n}(z_j - \delta_{ij,n+j})^2)}
$$

$$
= e^{-(\sum_{j=1}^{m^n}(e_j - \delta_{ij})^2 + \sum_{j=n+1}^{m^n}(z_j - \delta_{ij,n+j})^2) / \Delta^2}
$$

(24)

where $\Delta$ is the bandwidth and $\tilde{\delta}_i = [\delta_{i1}, \delta_{i2}, \ldots, \delta_{i4n}]$, $i = 1, 2, \ldots, m^4$ is the $i$th centre. It should be noted that the aforementioned expression of $S(Z)_i$ can be split by each dimension and recombined as

$$
S(Z)_i e^{(\sum_{j=1}^{m^n}(e_j - \delta_{ij})^2 + \sum_{j=n+1}^{m^n}(z_j - \delta_{ij,n+j})^2) / \Delta^2}
$$

$$
= \prod_{j=1}^n f_{k_j}
$$

(25)

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

$$
f_{k_j} = e^{-(e_j - \delta_{kj})^2 + (z_j - \delta_{kj,n+j})^2 + (\alpha_j - \delta_{kj,2n+j})^2 + (\bar{\alpha}_j - \delta_{kj,3n+j})^2 / \Delta^2}.
$$

(26)

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

$$
F(Z)_{\gamma} = W_{\gamma}^T S(Z)
$$

$$
= \sum_{k_1=1}^{m^4} \sum_{k_2=1}^{m^4} \sum_{k_3=1}^{m^4} \cdots \sum_{k_n=1}^{m^4} \alpha_{k_1,k_2,\ldots,k_n,\gamma} f_{k_\gamma},
$$

(27)

where

$$
S(Z) = \begin{bmatrix}
\alpha_{k_1,k_2,\ldots,k_n,\gamma} f_{k_1} \\
\alpha_{k_1,k_2,\ldots,k_n,\gamma} f_{k_2} \\
\vdots \\
\alpha_{k_1,k_2,\ldots,k_n,\gamma} f_{k_n}
\end{bmatrix}
$$

(28)

and $W_{\gamma} = [W_{\gamma 1}, W_{\gamma 2}, \ldots, W_{\gamma n}]$.

#### 3.2.2. Compression matrix $A$

To better illustrate DCRBF, let us introduce an operator matrix $A(m^{4n} \times mn^4)$, which can be used to compress the numbers of weights and nodes. To construct compression matrix $A$, a series of $m^4 \times m^4$ submatrices $\psi_i$ for
Figure 1. Conventional RBF NN.

Figure 2. Dimension compressed RBF NN.
\( i = 1, 2, 3, \ldots, m^4 \) is built as

\[
\psi_1 = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 0 & \cdots & 0 
\end{bmatrix} \quad (30)
\]

\[
\psi_2 = \begin{bmatrix}
0 & 1 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \cdots & 0 
\end{bmatrix}
\]

\[
\psi_{m^4} = \begin{bmatrix}
0 & 0 & \cdots & 1 \\
0 & 0 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1 
\end{bmatrix}
\]

Then use \( \psi_i \) and unit matrix \( E \) to design the compression matrix \( A \):

\[
A = \begin{bmatrix}
E & \psi_1 & \psi_1 & \psi_1 & \cdots & \psi_1 \\
E & \psi_2 & \psi_1 & \psi_1 & \cdots & \psi_1 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
E & \psi_{m^4} & \psi_1 & \psi_1 & \cdots & \psi_1 \\
E & \psi_1 & \psi_{m^4} & \psi_1 & \psi_1 & \cdots & \psi_1 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
E & \psi_{m^4} & \psi_{m^4} & \psi_1 & \psi_1 & \cdots & \psi_1 \\
E & \psi_{m^4} & \psi_{m^4} & \psi_{m^4} & \psi_{m^4} & \cdots & \psi_{m^4} 
\end{bmatrix} \quad (33)
\]

\( m^4 \) weights for each conventional RBF NN. Thus there are much less nodes and weights to be built and updated \((nm^4 \text{ nodes and } n^2 m^4 \text{ weights})\). The number \( n \) denoting the DOF is successfully moved from an exponent to a factor, which avoids the exponential increasing of the number of nodes and weights w.r.t. DOF.

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

\[
W_{k,\gamma}^T S(Z) = \sum_{k_1=1}^{m^4} \omega_{k1,\gamma} f_{k1} \left( \sum_{k_2=1}^{m^4} f_{k2} \sum_{k_3=1}^{m^4} f_{k3} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) + \sum_{k_1=1}^{m^4} \omega_{k2,\gamma} f_{k2} \left( \sum_{k_1=1}^{m^4} f_{k1} \sum_{k_3=1}^{m^4} f_{k3} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) + \cdots + \sum_{k_1=1}^{m^4} \omega_{kn,\gamma} f_{kn} \left( \sum_{k_1=1}^{m^4} f_{k1} \sum_{k_2=1}^{m^4} f_{k2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}} \right) \quad (34)
\]

where

\[
W_{k,\gamma} = [\omega_{k1=1,\gamma}, \omega_{k1=2,\gamma}, \ldots, \omega_{k1=m^4,\gamma}, \omega_{k2=1,\gamma}, \ldots, \omega_{k2=m^4,\gamma}, \ldots, \omega_{kn=1,\gamma}, \ldots, \omega_{kn=m^4,\gamma}]^T
\]

and

\[
S(Z) = \begin{bmatrix}
\begin{bmatrix}
f_{k_1=1} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) \\
f_{k_1=2} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) \\
\vdots \\
f_{k_1=3} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) \\
\vdots \\
f_{k_1=m^4} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) \\
f_{k_2=1} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) \\
\vdots \\
f_{k_2=2} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) \\
\vdots \\
f_{k_2=m^4} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_n=1}^{m^4} f_{kn} \right) \\
\vdots \\
f_{kn=1} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}} \right) \\
\vdots \\
f_{kn=2} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}} \right) \\
\vdots \\
f_{kn=m^4} \left( \sum_{k_1=1}^{m^4} f_{k_1} \sum_{k_2=1}^{m^4} f_{k_2} \cdots \sum_{k_{n-1}=1}^{m^4} f_{k_{n-1}} \right) 
\end{bmatrix}
\end{bmatrix}
\]

### 3.2.3. DCRBF

Inspired by the format of the expression for conventional RBF NN in (27), a dimension-split RBF NN of \( n \)-DOF is constructed in Figure 2, which only has \( nm^4 \) dots and \( nm^4 \) weights for each DOF of the outputs, which avoids the exponential growth w.r.t. the DOF. As shown in Figure 2, in the input layer, each DOF of \( x_1, x_2, \alpha \) and \( \dot{\alpha} \) are taken out of the input \( Z \), and reassembled as a new input vector \( \xi_i = [x_{1i}, x_{2i}, \alpha_i, \dot{\alpha}_i] \), for \( i = 1, 2, \ldots, n \). For each new input vector \( \xi_i \), in the conventional RBF layer, we build a \( m^4 \)-dot conventional RBF NN. The mathematical manipulation layers consist of an addition layer and a multiplication layer. In the output layer, there are
Comparing (36), (35) with (28), (29), using operator matrix $A$, for $\gamma = 1, 2, \ldots, n$, it can be seen that
\begin{equation}
AW_{k,\gamma} = W_{\gamma}
\end{equation}
and for $W_k = [W_{k,1}, W_{k,2}, \ldots, W_{k,n}]$,
\begin{equation}
AW_k = W,
\end{equation}
\begin{equation}
A^T S(Z) = S'(Z).
\end{equation}

### 3.2.4. Solution of $W_k$ and approximation error

Considering a training procedure for the weights, according to Lemma 5, use a least-square method to solve the weights $W$ and we can obtain the optimal weights $W^*_\gamma$, for $\gamma = 1, 2, \ldots, n$.

\begin{equation}
W^*_\gamma = \phi(Z)^\dagger g_\gamma(n_i),
\end{equation}

where $n_i$ is the number of training samples and $g_\gamma(n_i)$ is $n_i \times 1$ desired response in dimension $\gamma$ of $F(Z)$. $\phi(Z) = [S(Z_1)^T, S(Z_2)^T, \ldots, S(Z_n)^T]^T$ is an $n_1 \times m^4$ interpolating matrix. Considering an error constant $m^{4n} \times m^{4n}$ matrix $\kappa$:
\begin{equation}
\kappa = E_1 - AA^T,
\end{equation}
where $E_1$ is a $m^{4n} \times m^{4n}$ unit matrix. Considering (37) and (39), the least-squares method solution of $W_k$ can be derived of
\begin{equation}
\phi AW_{k,\gamma} = g_\gamma(n_i),
\end{equation}

where $A$ is the compression matrix derived above and $\phi_k(Z) = \phi A = [S'(Z_1), S'(Z_2), \ldots, S'(Z_n)]^T$ is an $n_1 \times nm^4$ interpolating matrix for $S'(Z)$. The optimal solution of $W^*_k,\gamma$ is
\begin{equation}
W^*_k,\gamma = A^\dagger \phi^\dagger g_\gamma(n_i)
\end{equation}
\begin{equation}
= A^\dagger W^*\gamma.
\end{equation}

The weights approaching error $\epsilon$ can be expressed as
\begin{equation}
\epsilon_\gamma = W^*_\gamma - AW^*_k,\gamma
\end{equation}
\begin{equation}
= \kappa W^*_\gamma,
\end{equation}
which is a constant for $\gamma = 1, 2, \ldots, n$. Using the definition of $\epsilon_\gamma$, and the output approximation error, $\mu_\gamma(Z)$ can be expressed as
\begin{equation}
\mu_\gamma(Z) = W^*_\gamma ^T S(Z) - W^*_{k,\gamma} ^T S'(Z)
\end{equation}
\begin{equation}
= W^*_\gamma ^T (E_1 - AA^T)^T S(Z)
\end{equation}
\begin{equation}
= (\kappa W^*_\gamma)^T S(Z).
\end{equation}

According to Lemma 2
\begin{equation}
|\mu_\gamma(Z)| = |\epsilon_\gamma^T S(Z)| \leq \bar{\epsilon}_\gamma,
\end{equation}
where $\bar{\epsilon}_\gamma = \|\epsilon_\gamma^T\|$ is a positive constant.

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

### 3.3. Stability analysis

Considering the dynamics of the robot in $\dot{V}_2$, applying DCRBF as described in section 3.2.3, we see that over a compact set $\Omega_z$

\begin{equation}
(\dot{\tilde{W}}_k^T - \tilde{\dot{W}}_k^T) S'(Z) = W_k^T S'(Z) = W^* T S(Z) - \mu(Z)
\end{equation}
\begin{equation}
= -J^T(x_1)f - C(x_1, x_2)x_2 - G(x_1)
\end{equation}
\begin{equation}
- M(x_1)\ddot{x} - \mu(Z) - \eta(Z),
\end{equation}

where $\tilde{W}_k = \dot{\tilde{W}}_k - W^*_k$ 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), $\mu(Z) = |\mu(Z)_1, \mu(Z)_2, \ldots, \mu(Z)_n|$ satisfy $|\mu(Z)| \leq \bar{\mu}$, where $\bar{\mu} = [\mu_1, \mu_2, \ldots, \mu_n]$. We propose $\tau_1 = -\tilde{W}_k S'(Z)$, where $\tilde{W}_k S'(Z)$ is used to approximate $W_k^* S'(Z)$.

The adaptive NN robot control law is designed as
\begin{equation}
\tau = \tau_1 + \tau_2
\end{equation}
\begin{equation}
= -(\dot{z})^T \sum_{i=1}^n k_{ai} e_i z_i - k_2 z - \tilde{W}_k S'(Z).
\end{equation}

Considering the following Lyapunov candidate function:
\begin{equation}
\dot{V}_3 = \dot{V}_2 + \sum_{\gamma=1}^n \tilde{W}_k^T Q^T_{\gamma} \tilde{W}_k,\gamma
\end{equation}
for $\gamma = 1, 2, \ldots, n$, where $Q_\gamma$ are positive definitive matrices. The adaptive law is given as follows:
\begin{equation}
\tilde{W}_k,\gamma = Q_\gamma [S'(Z)z_\gamma - \theta_\gamma \tilde{W}_k,\gamma],
\end{equation}
where $\theta_\gamma > 0(\gamma = 1, 2, \ldots, n)$ are gain constants.

Differentiating $\dot{V}_3$ with respect to time yields
\begin{equation}
\dot{V}_3 = \dot{V}_2 + \sum_{\gamma=1}^n \tilde{W}_k^T Q^T_{\gamma} \tilde{W}_k,\gamma.
\end{equation}

Substituting (48), (47) and (50) into (51), we obtain
\begin{equation}
\dot{V}_3 = - \sum_{i=1}^n k_{ai}^2 \bar{k}_{1i}^2 e_i^2 - z^T k_2 z + z^T (\eta(Z)
\end{equation}
Thus considering Property 2 we can obtain
\[ + \mu(Z) - \hat{W}_kS'(Z) \]
\[ + \sum_{i=1}^{n} k_{ai}^2 e_i z_i - z^T(z^T) + \sum_{i=1}^{n} k_{ai}^2 e_i z_i \]
\[ + \sum_{i=1}^{n} \hat{W}_{k_i} S(Z) z_i - \sum_{i=1}^{n} \hat{W}_{k_i} \theta_i \hat{W}_{k_i}, \]

when \( z = [0, 0, \ldots, 0]^T \), \( V_2 = -\sum_{i=1}^{n} (k_{ai}^2 k_{ai} e_i^2 / k_{ai}^2 - x_{ai}^2) \). We can still draw the asymptotic stability of the system according to Lemma 3. In the case of \( z \neq [0, 0, \ldots, 0]^T \), we have
\[ V_3 \leq -\sum_{i=1}^{n} \frac{k_{ai}^2 k_{ai} e_i^2}{k_{ai}^2 - x_{ai}^2} - z^T(k_2 - I)z \]
\[ + \frac{1}{2} (\| \hat{\mu} \|^2 + \| \hat{\eta} \|^2) \]
\[ - \sum_{i=1}^{n} \hat{W}_{k_i, \theta_i} \hat{W}_{k_i} \]
\[ - \sum_{i=1}^{n} (\theta_i - \frac{1}{2} \theta_i^2) \| \hat{W}_{k_i, \theta} \|^2 \]
\[ + \frac{1}{2} \sum_{i=1}^{n} \| W_{k_i, \theta} \|^2 \]
\[ - \sum_{i=1}^{n} \theta_i \eta_i \leq -\sum_{i=1}^{n} \hat{W}_{k_i} \theta_i \hat{W}_{k_i}, \]

Thus considering Property 2 we can obtain
\[ V_3 \leq -p V_3 + C, \]
where
\[ p = \min \left[ k_{ai}, \frac{\lambda_{\min}(2(k_2 - I))}{\lambda_{\max}(M)}, \frac{2(\theta_i - \frac{1}{2} \theta_i^2)}{\lambda_{\max}(Q_y^1)} \right] \]
\[ C = \frac{1}{2} (\| \hat{\mu} \|^2 + \| \hat{\eta} \|^2) + \frac{1}{2} \sum_{i=1}^{n} \| W_{k_i, \theta} \|^2, \]

where \( \lambda_{\min}(\bullet) \) and \( \lambda_{\max}(\bullet) \) denote the minimum and maximum eigenvalues of matrix \( \bullet \). To ensure \( p > 0 \), \( k_2 \) and \( \theta_i \) must satisfy the following conditions:
\[ \lambda_{\min}(2(k_2 - I)) > 0, \]
\[ 0 < \theta_i < 2. \]

If \( C \) can be zero, the system can be said to achieve the exponential stability. However, considering the approximation error of the NN for our controller, \( c = (\| \hat{\mu} \|^2 \| \hat{\eta} \|^2)^{1/2} / 2 \) which is a positive constant. Thus the system can only be stable instead of exponentially stable.

Theorem 1: According to Property 2, we know that \( x_{ai} = e_i + x_{di} < k_{ai} \) is bounded. Since the condition satisfies \( -k_{ai} < -k_{ci} \leq x_{ai} < k_{ai} < k_{ci} \), according to (6) and (7), we have \( -k_{ai} - k_{ci} < e_i < (k_{ai} - k_{ci}) \), which is bounded. Then considering the definition of \( \alpha \) in (11) and Property 4, \( \alpha \) is bounded too. According to (54), Lemma 1, Property 3 and in terms of (7), (13) and (49), we can safely conclude \( e, z \) and NN weight estimated error \( \hat{W}_k \) are bounded. In terms of the boundedness of \( \alpha \) and \( z \), according to \( x_2 = x + \alpha, x_2 \) is bounded. Thus we can safely say that the signals of the closed-loop system are semiglobally uniformly bounded (SGUB). Also, the closed-loop error signals \( e \) and \( z \) will remain within the compact sets \( \Omega_e, \Omega_z \), respectively, defined by
\[ \Omega_e := \{ e \in \mathbb{R}^n \mid |e_i| \leq \sqrt{(k_{ai} - k_{ci})(1 - e^{-D})} \}, \]
\[ \Omega_z := \{ z \in \mathbb{R}^n \mid \|z\| \leq \sqrt{\frac{D}{\lambda_{\max}(M)}} \}, \]

where \( i = 1, 2, \ldots, n \) 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

| 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 \times 10^{-3} | kgm² |
| Moment of Inertia of Link 2| \( I_2 \)          | 60.75 \times 10^{-3} | kgm² |
| Length of Link 1           | \( l_1 \)          | 0.75              | m    |
| Length of Link 2           | \( l_2 \)          | 0.45              | m    |

Figure 3. Structure of the system.
satisfy the condition in (57). The gains $Q_\gamma$ and $\theta_\gamma$ in NN adaptive law should be positive. According to (58), $\theta_\gamma$ should also be smaller than 2. In terms of (55), (59), (60), if the parameters $k_{ii}$ and $k_2$ are chosen to be relatively large, while $Q_\gamma$ and $\theta_\gamma$ are chosen relatively small, then the amplitude of trucking the error could be made smaller.

4. Simulation studies

In order to test the validity of the control, we have done a simulation on a 2-DOF robot manipulator which has two revolute joints in the vertical plane. In the model shown in 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 centre distance of link $i$ and the inertia of link $i$, where $i = 1, 2$ and $q = [q_1, q_2]$. Other simulation parameters are shown in Table 1.

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

$$G(q) = \begin{bmatrix} (m_1 l_2 + m_2 l_1) g \cos q_1 \\ m_2 l_2 g \cos(q_1 + q_2) \end{bmatrix}$$

(61)

Figure 4. The results of DCRBF with iBLF: (a) $x_1$ and $x_{d1}$, (b) $x_2$ and $x_{d2}$, (c) tracking errors $e_1$ and $e_2$, (d) tracking errors $z_1$ and $z_2$, (e) $\tau_1$ and $\tau_2$ and (f) $\|W_{k,1}\|$ and $\|W_{k,2}\|$.
\[
M(q) = \begin{bmatrix}
-m_1l_2^2 + l_1 + l_2 + m_1l_2^2 + l_1l_2\cos q_2 + l_1 \\
+m_1l_2^2 + l_1l_2\cos q_2 + l_1 \\
m_1l_2^2 + l_1l_2\cos q_2 + l_1 \\
\end{bmatrix}
\]
(62)

\[
C(q, \dot{q}) = \begin{bmatrix}
-m_1l_2(\dot{q}_1 + \dot{q}_2)\sin q_2 \\
m_1l_2\sin q_2 \\
m_1l_2\sin q_2 \\
\end{bmatrix}
\]
(63)

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

We choose the desired trajectory as \( x_d = [\sin(0.5t), 2\cos(0.5t)] \), where \( t \in [0, T] \) and \( T = 30 \text{s} \). To maintain the constraints \( |x_{1i}| \leq k_{a1i} < k_{ai}, |x_{di}| \leq k_{a1i} < k_{ai}, \) we define \( k_d = [1.1, 2.1]^T \) and \( f = [2\cos(t) + 0.5 + d(t), \sin(t) + 1 + d(t)] \), where \( d(t) \) is a white Gaussian noise. To satisfy condition (57), \( k_2 \) is chosen as \([65, 15]\). \( k_1 \) is chosen as \([35, 6]\). Our chosen 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 \( W_k \) is examined with the calculated error \( \epsilon_{\gamma} \) for \( \gamma = 1, 2 \), which consists of calculated constant matrix \( \kappa \). For the DCRBF,
the centres are chosen in the area of $[-1,1] \times [-1,1]$ for each dimension, which constitute $f_{k_1}=1, f_{k_2}=2, \ldots, f_{k_4}=2^4$ and $f_{k_1}=1, f_{k_2}=2, \ldots, f_{k_8}=2^4$ all 32 nodes. For the conventional RBF, the centres are chosen in the area of $[-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \times [-1,1]$, which constitute 16 submatrices. For the conventional RBF, the centres are chosen in the area of $[-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \times [-1,1]$, which constitute all 32 nodes. For the conventional RBF, the centres are chosen in the area of $[-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \times [-1,1] \times [-1,1]$ arraying like

$$S(Z) = \begin{bmatrix} f_{k_1}=1 & f_{k_2}=1 & f_{k_4}=1 \cdots & f_{k_1}=2 & f_{k_2}=1 \cdots & f_{k_1}=2^4 & f_{k_2}=1 \cdots & f_{k_1}=2 & f_{k_2}=2^4 \cdots & f_{k_1}=2 & f_{k_2}=2^4 \cdots & f_{k_1}=2^4 & f_{k_2}=2^4 \end{bmatrix}.$$

(65)

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

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

From these graphs, we know that the prescribed trajectory tracking performance of DCRBF with the implementation of iBLF is satisfactory from Figure 4(a,b). The system errors shown in Figure 4(c,d) are converging to a small value which is close to zero. Comparing Figure 4(a–e) with Figure 5(a–e), we can conclude that the DCRBF can approximate the system uncertainties just as well as the conventional RBF.

According to Figures 4(f), 5(f), $W_k$ and $W$ approach stability over time. So we assume that NN approaches to the ideal model $W^*_{k}, W^*$ when time reaches 30 s. Then we graph Figure 6 to show the approximating error between the weights of DCRBF and the conventional RBF, which is formulated as $\Delta W^* = W^*_{k} - AW^*_{k} - \epsilon_{\gamma}$, for $\gamma = 1, 2$, where $\epsilon_{\gamma} = \kappa W^*_{k}$ and $\kappa = E_1 - A A^\dagger$. The results are very small values, close to zero, which proves that the approximating error of weights is as we calculated 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.

As for the time and energy saving, according to the structure of DCRBF, in each step of the online NN control, there will be $8nm^4 + n^2m^4 - 2n$ addition operations, $6nm^4 + n^2m^4$ multiplication operations and $nm^4$ power operations, which is much less than $m^{2n}(9n - 1) - n$ addition operations, $m^{4n} - n$ multiplication operations and $m^{3n}$ power operations in conventional RBF, whose structure is shown in Figure 1. So in theory, using DCRBF takes much less computation time than conventional RBF.

In terms of calculation speed and operation memory, we repeated the NN control simulations in Section 4 on MATLAB (Matlab R2014a with Inter Core(TM) i7-4710MQ CPU) 20 times and calculate their average. The average time consumed for a complete control cycle using DCRBF and the conventional RBF are 3.326191 s and 30.677051 s, and the peak memory footprints are 75 kB and 297 kB respectively. We can calculate that the speed improvement is 89.25% and the memory usage is 74.74% lower. Thus it can be concluded by the number of operations, our proposed NN control strategy using DCRBF is highly optimised in contrast to the conventional RBF NN control. This shows the advantages of DCRBF NN control compared to conventional RBF NN control in terms of time and energy savings.

5. Conclusion

This paper presents an innovative adaptive NN control using DCRBF for n-DOF robot system with full-state constraints and unknown dynamics. By utilising DCRBF, the problem of excessive numbers 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. 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 demonstrated through the results presented in this paper.
Acknowledgments

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.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Jun Xia received the B.Sc. and M.Sc. degrees from South China university of Technology. He is currently pursuing the Ph.D. degree with the Sun Yat-Sen University, Guangzhou, China. His research interests include surgical robot, computer version, SLAM, and machine learning.

Yujia Zhang received the B.E. degree in electrical engineering and automation in Huazhong University of Science and Technology in 2015, and the M.S. degree in electrical engineering in South China University of Technology, China, in 2018. He is currently pursuing the Ph. D. degree in City University of Hong Kong. His current research interests include machine learning and computer vision.

Chenguang Yang is a Professor of Robotics. He received the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010 and performed postdoctoral research in human robotics at Imperial College London, London, UK from 2009 to 2010. He has been awarded EU Marie Curie International Incoming Fellowship, UK EPSRC UKRI Innovation Fellowship, and the Best Paper Award of the IEEE Transactions on Robotics as well as over ten conference Best Paper Awards. His research interest lies in human robot interaction and intelligent system design.

Min Wang received the Ph.D. degree from the Institute of Complexity Science, Qingdao University, Qingdao, China, in 2009. From November 2017 to 2018, she is an academic visitor with Brunel University London, Uxbridge, United Kingdom. She is currently an Associate Professor with the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China. She has authored or co-authored nearly 30 papers in top international journals. Dr. Wang was a recipient of the Excellent Doctoral Dissertations Award of Shandong Province in 2010, the Science and Technology New Star of Zhujiang, Guangzhou, in 2014, and the youth talent of Guangdong Tezhi Plan in 2016. Her current research interests include nonlinear systems, intelligent control, robot control, and dynamic learning.

Andy Annamalai is currently a Senior Lecturer with Winchester University. Previously, he was a Lecturer with University of Highlands and Islands (UHI), Scotland, UK, prior to which he was a Researcher at the Marine and Industrial Dynamic Analysis Research Group, Plymouth University, UK.

ORCID

Jun Xia  http://orcid.org/0000-0002-1224-6142
Yujia Zhang  http://orcid.org/0000-0003-3991-7388
Chenguang Yang  http://orcid.org/0000-0001-5255-5559
Min Wang  http://orcid.org/0000-0001-7025-7651

References

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

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, 43(8), 1587–1596.

Berkovitz, L. D. (2013). Optimal control theory (Vol. 12). Berlin, Germany: 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). Springer.

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). Upper Saddle River, Pearson Prentice Hall.

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.

Fei, J., & Ding, H. (2012). Adaptive sliding mode control of dynamic system using rbf neural network. Nonlinear Dynamics, 70(2), 1563–1573.

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). Upper Saddle River, NJ: Pearson.

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

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.

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 backpropagation 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.-I., 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.-I., 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.

Tsai, C.-C., Huang, H.-C., & Lin, S.-C. (2010). Adaptive neural network control of a self-balancing two-wheeled scooter. *IEEE Transactions on Industrial Electronics*, 57(4), 1420–1428.

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*, 47, 2244–2255.

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*, 13(3), 1162–1171.

Yang, C., Wang, X., Cheng, L., & Ma, H. (2017). Neural-learning-based telerobot control with guaranteed performance. *IEEE Transactions on Cybernetics*, 47(10), 3148–3159.

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 *2006 45th IEEE conference on decision and control* (pp. 3530–3535). IEEE.

Yu, H., Xie, T., Paszczynski, S., & Wilamowski, B. M. (2011). Advantages of radial basis function networks for dynamic system design. *IEEE Transactions on Industrial Electronics*, 58(12), 5438–5450.

Zhang, T., & Ge, S. S. (2009). Adaptive neural network tracking control of mimic 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.