This is an author produced version of a paper published in:
Neurocomputing

Cronfa URL for this paper:
http://cronfa.swan.ac.uk/Record/cronfa31618

Paper:
http://dx.doi.org/10.1016/j.neucom.2016.12.048

This article is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence. Authors are personally responsible for adhering to publisher restrictions or conditions. When uploading content they are required to comply with their publisher agreement and the SHERPA RoMEO database to judge whether or not it is copyright safe to add this version of the paper to this repository.
http://www.swansea.ac.uk/iss/researchsupport/cronfa-support/
Discrete-time Optimal Adaptive RBFNN Control for Robot Manipulators with Uncertain Dynamics

Runxian Yang\textsuperscript{a,b,c}, Chenguang Yang\textsuperscript{c}\textsuperscript{*}, Mou Chen\textsuperscript{a}, Andy SK Annamalai\textsuperscript{d}

\textsuperscript{a}College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China
\textsuperscript{b}College of Electric and IT, Yangzhou Polytechnic Institute, Yangzhou, China
\textsuperscript{c}Zienkiewicz Centre for Computational Engineering, Swansea University, Swansea, UK
\textsuperscript{d}University of the Highlands and Islands, Scotland, UK

Abstract

In this paper, a novel optimal adaptive radial basis function neural network (RBFNN) control has been investigated for a class of multiple-input-multiple-output (MIMO) nonlinear robot manipulators with uncertain dynamics in discrete time. To facilitate digital implementations of the robot controller, a robot model in discrete time has been employed. A high order uncertain robot model is able to be transformed to a predictor form, and a feedback control system has been then developed without noncausal problem in discrete time. The controller has been designed by an adaptive neural network (NN) based on the feedback system. The adaptive RBFNN robot control system has been investigated by a critic RBFNN and an actor RBFNN to approximate a desired control and a strategic utility function, respectively. The rigorous Lyapunov analysis is used to perform to establish uniformly ultimate boundedness (UUB) of closed-loop signals, and the high-quality dynamic performance against uncertainties and disturbances is obtained by appropriately selecting the controller parameters. Simulation studies validate that the control scheme has performed better than other available methods currently, for robot manipulators.

Keywords: Discrete-time system; Neural networks; Robot manipulator; Adaptive control; Dynamics uncertainties

1. Introduction

Robot manipulators are typically modelled as MIMO systems with high nonlinearity, and they are usually subject to unmodelled dynamics and uncertainty [1, 2, 3]. Control signals of nonaffine nonlinear robot manipulators have nonlinear inputs with coupling effect, uncertain parameters and unknown nonlinear functions, and thus, it is still a challenging problem to design reliable control for general uncertain robot manipulators. With advances of robot technologies, application of manipulators in industry and other fields become increasingly popular, and the researches on control design for robot manipulators have attracted much attention, e.g., feedback linearization method [4, 5], sliding mode control methods [6, 7, 8, 9], have been investigated for robot trajectory tracking control and optimal control. Furthermore, intelligent control methods and complex control schemes have been proposed or extended for robot system control, e.g., adaptive control, adaptive and fuzzy complex control, and adaptive with sliding complex control [10, 11, 12, 13, 14] for robot manipulators. To compensate for the effects caused by robot uncertain dynamics, adaptive neural network (ANN) researches have been extensively exploited, due to its capacity of online learning and universal approximation of smooth nonlinear functions in [15, 16, 17].

In recent years, adaptive RBFNN methods have been developed to be more powerful to deal with dynamics uncertainties that are more complex in practical application. In [18], an adaptive RBFNN algorithm based control guaranteeing stability of closed-loop robot system online, has been investigated. In [19, 20], the robust controller with a adaptive RBFNN has been presented for the effects caused by dynamics uncertainties. The closed-loop control systems achieve UUB stability for robot manipulators, and their stability analysis has been well established in continuous time. At the same juncture, the controllers of robot manipulators using digital control technology and high-speed data transmission...
Based on digital computers are playing important roles and have more convenient in practice. Hence, recent research works for robot manipulators gradually focus on discrete-time control. In [21, 18], a robot dynamics model and a robot control method are applied in discrete time, these approaches to on-line control using acceptable discrete-time robot models seem to be very convenient. In [22], by combination of one-step-ahead control and ANN, a stable ANN approach has been developed for a class of nonlinear MIMO robot system in discrete time. In [23], an ANN control is presented for a class of MIMO nonlinear robot systems with block triangular structure in discrete time, and the systems can be separated into n subsystems in pure-feedback form, and which has unknown control directions and complex couplings. In [24], a stable adaptive controller employing neuro-fuzzy method as an estimator for a class of robot manipulators has been proposed in discrete time. In [25], by employing an adaptive fuzzy estimator, a discrete-time model-free control law has been developed to compensate for dynamics uncertainties of robot manipulators. These approaches have performed well to guarantee robot stability, and most of them mainly concern in stability of robot manipulators in discrete time. However, the researches can well guarantee stability of closed-loop robot systems, but realizing trajectory tracking optimal control are seldom. Thus, an optimal control scheme proposed in discrete time for a class of robot manipulators with uncertain dynamics is the main research objective in this paper.

To address the optimal trajectory tracking performance based on stability closed-loop robot control systems, we develop a novel discrete-time optimal adaptive RBFNN control for a class of robot manipulators with uncertain dynamics. To predict control output, the output feedback control is first studied by extending our previous research works [26] for the robot manipulators in discrete time, and an output-feedback system is investigated by transforming the discrete-time robot dynamics into a two-step ahead predictor form, the model relates to the inputs and the outputs of robot systems. Furthermore, based on the output-feedback system, a novel optimal adaptive neural control is investigated by extending our recent research results [27], which uses deterministic learning technique for a class of SISO nonlinear systems. The proposed control method includes an actor RBFNN as an approximation to the desired control input, and a critic RBFNN as an approximate to the desired strategic utility function to optimize the control process. And the weight rule is designed by applying the output of the critic RBFNN and trajectory tracking error. And stability of the closed-loop robot systems is rigorously proved by Lyapunov theory. Finally, the novel optimal adaptive RBFNN control is applied to robot systems with uncertainty dynamics, whether existing larger or smaller external disturbances or not, to achieve supreme control performance.

The main contributions of this paper are highlighted as follows:

i. Transformation of a high order discrete-time robot model to a two-step ahead predictor form, to enable output-feedback system design without non-causal problem

ii. Investigation of optimal performance based on the predictor form of robot dynamics, and RBFNN approximation.

iii. To achieve optimal trajectory tracking performance, an utility function is defined, and a critic RBFNN is designed to approximate the function.

iv. The actor RBFNN update law is designed using both the strategic utility function and tracking error.

Throughout this paper, the following notations used are detailed in Table 1

2. Problem Formulation and Preliminaries

In this paper, we consider a class of n-degrees of freedom (DOF) rigid robot manipulators with uncertain dynamics. The dynamics model is described as follows,

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_d \quad (1) \]

where \( q \in \mathbb{R}^n \) denotes the joint position, and \( \dot{q} \in \mathbb{R}^n \) is the joint velocity, \( \ddot{q} \in \mathbb{R}^n \) denotes the joint acceleration, \( M(q) \in \mathbb{R}^{n \times n} \) is the symmetric positive definite inertia matrix, \( C(q, \dot{q}) \in \mathbb{R}^{n \times n} \) is the Coriolis-Centrifugal torque matrix, \( G(q) \in \mathbb{R}^n \) denotes the gravity torque vector, \( \tau \in \mathbb{R}^n \) is the vector of control input torque, \( \tau_d \in \mathbb{R}^n \) is the external force torque caused by robotic uncertainty.

According to [1], the following properties hold for the rigid robot manipulators described in (1):

**Property 1.** The matrix \( 2C(q, \dot{q}) - M(q) \in \mathbb{R}^{n \times n} \) is a skew-symmetric matrix, such that

\[ x^T [M(q) - 2C(q, \dot{q})] x = 0, \forall x \in \mathbb{R}^n \quad (2) \]

**Property 2.** The \( M(q) \), a symmetric and positive definite inertia matrix, is uniformly bounded, there \( m > 0 \)
Table 1: NOMENCLATURE

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>∥ · ∥</td>
<td>the Euclidean norm of vectors and induced norm of matrices</td>
</tr>
<tr>
<td>a := b</td>
<td>a is defined as b</td>
</tr>
<tr>
<td>[ ]T</td>
<td>the transpose of a vector or a matrix</td>
</tr>
<tr>
<td>[ ]⁻¹</td>
<td>the inverse of a n-order reversible matrix</td>
</tr>
<tr>
<td>0[p]</td>
<td>p-dimensional zero vector</td>
</tr>
<tr>
<td>I[m]</td>
<td>m-dimensional identity matrix</td>
</tr>
<tr>
<td>W*</td>
<td>the ideal neural net weight matrix at the kth step</td>
</tr>
<tr>
<td>ˆWk</td>
<td>the estimate of W*</td>
</tr>
<tr>
<td>Ẇk</td>
<td>ˆWk − W*, the weight estimate error</td>
</tr>
<tr>
<td>q</td>
<td>the n-dimensional joint position</td>
</tr>
<tr>
<td>˘q</td>
<td>the n-dimensional joint acceleration</td>
</tr>
<tr>
<td>q̄d</td>
<td>the n-dimensional ideal joint position</td>
</tr>
<tr>
<td>M(q)</td>
<td>the n × n dimensional symmetric positive definite inertia matrix</td>
</tr>
<tr>
<td>C(q, ˘q)</td>
<td>the n × n dimensional Coriolis-Centripetal torque matrix</td>
</tr>
<tr>
<td>G(q)</td>
<td>the n-dimensional gravity torque vector</td>
</tr>
<tr>
<td>τ</td>
<td>the n-dimensional vector of control input torque</td>
</tr>
<tr>
<td>τd</td>
<td>the n-dimensional external force torque</td>
</tr>
<tr>
<td>T</td>
<td>the sampling time</td>
</tr>
<tr>
<td>tk</td>
<td>the sampling time, and tk = kT</td>
</tr>
<tr>
<td>p^i</td>
<td>the sampled joint angle at time t_k, and p^i = q(t_k)</td>
</tr>
<tr>
<td>v^i</td>
<td>the sampled joint velocity at time t_k, and v^i = ˘q(t_k)</td>
</tr>
<tr>
<td>τ^i</td>
<td>the sampled joint force at time t_k</td>
</tr>
<tr>
<td>τ^i_d</td>
<td>the sampled external disturbance force at time t_k</td>
</tr>
<tr>
<td>p^i_d</td>
<td>the sampled ideal joint position at time t_k</td>
</tr>
<tr>
<td>ξ^i</td>
<td>the trajectory error, e^i is defined as p^i − p^i_d</td>
</tr>
<tr>
<td>e^i</td>
<td>the new error function</td>
</tr>
<tr>
<td>˘τ^∗_i</td>
<td>the ideal control input</td>
</tr>
<tr>
<td>Q^i</td>
<td>the strategic utility function</td>
</tr>
<tr>
<td>Γd</td>
<td>the diagonal critic learning rate matrix</td>
</tr>
<tr>
<td>Γr</td>
<td>the diagonal action learning rate matrix</td>
</tr>
<tr>
<td>k_p</td>
<td>the scaling factor, the proportion parameter</td>
</tr>
<tr>
<td>k_d</td>
<td>the scaling factor, the integral parameter</td>
</tr>
<tr>
<td>k_pd</td>
<td>the scaling factor</td>
</tr>
<tr>
<td>m</td>
<td>and m &gt; 0 are constants, and thus, M(q) satisfies the following inequality</td>
</tr>
<tr>
<td>3</td>
<td>m ≤ ∥M(q)∥ ≤ m</td>
</tr>
</tbody>
</table>

**Property 3.** The matrix C(q, ˘q) and the vector G(q) are bounded by ∥C(q, ˘q)∥ ≤ k_c∥q∥, and ∥G(q)∥ ≤ k_g, respectively, where k_c and k_g are positive constants.

2.1. RBFNN Construct

The RBFNN is able to approximate any nonlinear function, and it has good generalization ability and fast learning convergence speed. The RBFNN structure is described as follows [28]:

$$F(W, z) = W^T S(z), \quad W \in \mathbb{R}^{N_i \times N_o}, \quad S(z) \in \mathbb{R}^{N_i}$$  \hspace{1cm} (4)

where \(z = [z_1, z_2, \cdots, z_{N_o}] \in \mathbb{R}^{N_o}\) in \(\Omega_e\) is input vector, \(N_c\) is control input dimension, \(N_i\) is neuron node number and \(N_o\) is output dimension, \(W = [w_1, w_2, \cdots, w_{N_o}]\) is weight matrix with \(w_i \in \mathbb{R}^{N_i}\), \(i = 1, 2, \cdots, N_o\), \(S(z) = [s_1(z), s_2(z), \cdots, s_{N_o}(z)]^T\) with hidden layer output function \(s_i(z)\) is RBFNN function, and the Gaussian function is chosen as follows,

$$s_i(z) = e^{-b_i (z - c_i)^2/2b_i^2}$$  \hspace{1cm} (5)

where \(i = 1, 2, \cdots, N_o\), \(j = 1, 2, \cdots, N_i\), \(c_{ij}\) is the center of the jth neuron node for the ith input signal, and \(b_i\) is the width of the jth neuron.

Numerous results indicate that for any continuous smooth function \(\varphi(z) : \Omega_e \rightarrow \mathbb{R}\) over a compact set \(\Omega_e \subset \mathbb{R}^{N_o}\), applying RBFNN (4) to approximate \(\varphi(z)\), if \(N_o\) is sufficiently large, a set of ideal bounded weights \(W^*\) exist, and we have

$$\varphi(z) = W^T S(z) + \mu(z)$$  \hspace{1cm} (6)

Considering the basis functions of RBFNN in (4), we use the following property to select relevant design parameter:

$$S(z)^T S(z) < N_i$$  \hspace{1cm} (7)

Noting that the ideal network weight \(W^*\) is unknown in (6). We often use the estimated weight \(\hat{W}\) to replace \(W^*\) to approximate a unknown, continuous, nonlinear function, and \(\hat{W}\) can be trained by a weight learning law, and thus,

$$\varphi(z) \approx \hat{W}^T S(z) \quad \text{or} \quad \hat{\varphi}(z) = \hat{W}^T S(z)$$  \hspace{1cm} (8)
2.2. Discretization for Robot Manipulator Model

Designing a robot controller is very important and meaningful in discrete time. We set the sampling time interval to be \( T \), and the sample angle at time \( t_k = kT \) is \( p^k \) for the \( n \)-DOF rigid robot manipulators in (1). Define \( p^k = q(t_k) \in \mathbb{R}^n \) and \( v^k = \dot{q}(t_k) \in \mathbb{R}^n \), the dynamic equation (1) in the continuous-time can be discretized [29, 30, 24] as

\[
(M(\xi^k)/T)(v^{k+1} - v^k) = (M(\xi^k) - M(p^k))v^k - f(p^k, v^k) + \tau^k + \tau^k_d
\]

where \( M(\xi^k) \in \mathbb{R}^{n \times n} \) is an inertia matrix with \( \xi^k = p^k + T v^k \in \mathbb{R}^n \), \( f(p^k, v^k) = C(p^k, v^k)v^k + G(p^k) \in \mathbb{R}^n \), \( C(p^k, v^k) \in \mathbb{R}^{n \times n} \) is Coriolis-Centripetal torque matrix and \( G(p^k) \in \mathbb{R}^n \) is gravitational synthetic torque vector. According to Property 1 and Property 2, \( M(\xi^k) \) is also a symmetric, positive definite inertia matrix, and it is bounded as \( m^\| \leq \| M(\xi^k) \| \leq \bar{m} \) with \( m > 0 \) and \( \bar{m} > 0 \) is able to be satisfied.

3. Robot Manipulator Feedback system

To avoid the possible noncausal problem in robot control, we extend our previous research works [26] to the MIMO robot systems in discrete time. The discrete-time robot dynamics in (9) is transferred into an output-feedback control system, and thus,

\[
\begin{align*}
    p^{k+1} &= p^k + T v^k \\
    v^{k+1} &= [(1 + T)I_n - TM^{-1}(\xi^k)M(p^k)]v^k \\
    &\quad - TM^{-1}(C(p^k, v^k))v^k - TM^{-1}(\xi^k)G(p^k) \\
    &\quad + TM^{-1}(\xi^d)^k + TM^{-1}(\xi^s)^k\tau^k_d
\end{align*}
\]

where \( \xi^d \in \mathbb{R}^n \) and \( p^k \in \mathbb{R}^n \) are system input and output in discrete time, respectively. \( \tau^k_d \) is bounded by an unknown constant \( \tau_d \), which makes \( \| \tau^k_d \| \leq \tau_d \).

It is easy to know \( M^{-1}(\xi^k) \) is also bounded, there \( m^\| > 0 \) and \( \bar{m}^\| > 0 \) are constants, and thus, the inequality \( m^\| \leq \| M^{-1}(\xi^k) \| \leq \bar{m}^\| \) is satisfied.

The control objective of this paper is to synthesize an adaptive RBFNN control \( \tau^k \) for system (10), then, all signals of the closed-loop system are bounded, and the joint position output \( p^k \) well tracks a bounded, ideal, reference trajectory \( p_d^k \in \Omega_{p_d} \), finally, the optimal control performance is able to be obtained, where \( \Omega_{p_d} \) is a compact set.

Noting (10), for the future states at the \( (k + 1) \)th step, the last state \( v^{k+1} \) depends on the control output \( \tau^k \), while \( p^{k+1} \) is associated with \( p^k \) and \( v^k \).

We rewrite the first equation of the robot model (10) as \( p^{k+1} - p^k - T v^k = 0_n \), and \( v^k \) is designed as \( v^k = 1/2(p^{k+1} - p^k) \). For the prediction \((k + 2)\) step of the robot manipulator system, we can obtain

\[
\begin{align*}
    p^{k+2} &= p^{k+1} + T v^{k+1} \\
    &= [(2 + T)I_n - TM^{-1}(\xi^k)M(p^k)]v^k \\
    &\quad - TM^{-1}(C(p^k, v^k))v^k \\
    &\quad - [(1 + T)I_n - TM^{-1}(\xi^k)M(p^k)]p^k \\
    &\quad - TM^{-1}(C(p^k, v^k))p^k \\
    &\quad - TM^{-1}(\xi^s)^kG(p^k) \\
    &\quad + 2 TM^{-1}(\xi^s)^k\tau^k_d + 2 TM^{-1}(\xi^s)^k\tau^k_d
\end{align*}
\]

To predict the output at the \((k + 2)\)th step, we move the \((k + 2)\) step back the \((k + 1)\)th step in (11), such that we get the \( p^{k+2} \) using the output-feedback method as follows

\[
\begin{align*}
    p^{k+1} &= [(2 + T)I_n - TM^{-1}(\xi^k)M(p^k)]v^k \\
    &\quad - TM^{-1}(C(p^k, v^k))v^k \\
    &\quad - [(1 + T)I_n - TM^{-1}(\xi^k)M(p^k)]p^k \\
    &\quad - TM^{-1}(C(p^k, v^k))p^k \\
    &\quad - TM^{-1}(\xi^s)^kG(p^k) \\
    &\quad + 2 TM^{-1}(\xi^s)^k\tau^k_d + 2 TM^{-1}(\xi^s)^k\tau^k_d
\end{align*}
\]

Substituting (12) to (11), we see that no future output is necessary to compute the control input. For convenience, let us define that

\[
\begin{align*}
    L^k &= (2 + T)I_n - TM^{-1}(\xi^k)M(p^k) - TM^{-1}(\xi^k)C(p^k, v^k) \\
    R^k &= (1 + T)I_n - TM^{-1}(\xi^k)M(p^k) - TM^{-1}(\xi^k)C(p^k, v^k) \\
    M^k &= T^2 M^{-1}(\xi^k), \quad G^k = G(p^k)
\end{align*}
\]

Then, by getting the values of current the \( k \) step and past the \( k - 1 \) step, we can obtain the output \( p^{k+2} \) as

\[
\begin{align*}
    p^{k+2} &= (L^k L^{k-1} - R^k)^k p^k - L^k R^{k-1} p^{k-1} \\
    &\quad - L^k M^{k-1}_d G^{k-1} - M^k G^k + L^k M^{k-1}_d \tau^{k-1} \\
    &\quad + M^k \tau^k_d + L^k M^{k-1}_d \tau^k_d + M^{k+1}_d \tau^k_d
\end{align*}
\]

and we can define

\[
\begin{align*}
    L^k &= (L^k L^{k-1} - R^k)^k p^k - L^k R^{k-1} p^{k-1} + L^k M^{k-1}_d \tau^{k-1} \\
    L^k &= L^k M^{k-1}_d G^{k-1} + M^k G^k \\
    L^k &= L^k M^{k-1}_d \tau^{k-1} + M^{k+1}_d \tau^k_d
\end{align*}
\]

Furthermore, we rewrite (13) as

\[
\begin{align*}
    p^{k+2} &= L^k p^k - L^k G^k + L^k \tau^{k-1} + \tau^k_d
\end{align*}
\]
Noting that $\psi(\cdot, \cdot, \cdot, \cdot, 0, 0)$ is continuous, such that all the arguments and continuously differentiable with respect to $\tau^k$ is continuous.

**Lemma 1.** According (9) and (10), $M^2$ is symmetric, positive definite matrix, there $m_z = T^2r^2$ and $\vec{m}_z = T^2\vec{m}$ are positive constants, then, $M^2$ is bounded with $m_z \leq \|M^2\| = T^2\|M^1(\ddot{z})\| \leq \vec{m}_z$.

Therefore, it is easy to obtain that $\|L_d^k\| \leq (3 + 2T + T\vec{m}_k \vec{m}_d)\vec{m}_d := \vec{r}_d$.

### 4. Adaptive NN Control Design

#### 4.1. Desired Control

The system ideal tracking output is $p_d^{k+2}$, the dynamics of tracking error $e^{k+2} \in \mathbb{R}^r$ can be obtained by

$$
e^{k+2} = p^{k+2} - p_d^{k+2} = L_z^k - L_{\vec{d}}^k + M^k_{\tau} \tau^k - p_d^{k+2} = 0 \quad (15)$$

It is noted that a ideal force torque control input $\tau_{n}^k$ [31], such that

$$L_z^k - L_{\vec{d}}^k + M^k_{\tau} \tau_{n}^k - p_d^{k+2} = 0 \quad (16)$$

or

$$\tau_{n}^k = M^{-1}_{\tau} \left(p^{k+2} - L_{\vec{d}}^k + L_z^k\right) \quad (17)$$

**Lemma 2.** There $m_{\tau}^2 = 1/\vec{m}_r$ and $\vec{m}_{\tau}^2 = 1/m$ are positive constants, $M^k_{\tau}$ is bounded with $m_{\tau}^2 \leq \|M^k_{\tau}\| \leq \vec{m}_{\tau}$. Then, the two-step predictor for trajectory error $e^{k+2}$ can be constrained as

$$\|e^{k+2}\| = \|L_d^k\| \leq \vec{r}_d \quad (18)$$

We know the desired control $\tau_{n}^k$ is not obtained with the unknown $M^k_{\tau}$, $L_z^k$ and $L_{\vec{d}}^k$. Applying RBFNN to approximate the desired input by adaptive learn $\tau_{n}^k$ will make tracking error $e^{k+2}$ 0 after 2 steps, if $\tau_d^k = 0$ and $\tau_{d-1}^k = 0$ in (15).

#### 4.2. Actor RBFNN Control

From Section 2.1, the ideal weight matrix $W_r$ exists, we use a Gaussian function $S_{\tau}(\ddot{z})$ to approximate $\tau_{n}^k$ as follows

$$\tau_{\theta}(\ddot{z}) = W_r^T S_{\tau}(\ddot{z}) + \epsilon_r(\ddot{z}) \quad (19)$$

where the vector $\ddot{z}$ is RBFNN input signal, and it is designed as

$$\ddot{z} = [p^{k+2}, p^{k-1}, \dot{p}^{k-1}, \dot{p}^{k}, \ddot{p}^{k} \dot{p}^{k+2}]^T \in \Omega_\ddot{z}$$

$\Omega_\ddot{z}$ is a sufficient large compact set and corresponds to $\Omega_\ddot{z}$. The number of neuron in hidden layer of RBFNN is $N_r$, and the ideal weight matrix $W_r = \in \mathbb{R}^N_x$ is given

$$W_r = [w_{11}, \ldots, w_{1n}, \ldots, w_{n1}, \ldots, w_{nn}]$$

where $i = 1, 2, \ldots, N_r$, $j = 1, 2, \ldots, n$, $w_i$ is the weight vector from all hidden layer neurons to the rth output $\tau_r$, $r = 1, 2, \ldots, n$, the $S_{\tau}(\ddot{z}) \in \mathbb{R}^{N_r}$ is the regressor matrix, $\|\epsilon_r(\ddot{z})\| \leq \epsilon_r, \epsilon_r > 0$ is an approximation error. We know the ideal control $\tau_{n}^k(\ddot{z})$ can easily be bounded.

Noticing (15) and (17), we use RBFNN as an approximation of $\tau_{n}^k(\ddot{z})$ with proportion integral (PD) control to optimize control performance. Then, the system control input is given as:

$$\dot{\tau}^k = -k_p \epsilon^k - k_d (\epsilon^k - \epsilon^{k-1}) + \dddot{\tau}\hat{\tau}_n^k(\dddot{z}) \quad (20)$$

where, $k_p > 0$ and $k_d > 0$ are scaling factors, $k_{pd} = k_p + k_d > 0$, $\hat{W}^k \in \mathbb{R}^{N_r \times n}$ is used to approximate unknown function $\tau_{n}^k(\ddot{z})$ in (19) with compact set $\Omega_\ddot{z}$. According to the equation (16), we have $p^{k+2} = L_{\vec{d}}^k - L_z^k + M_{\tau} \tau_{n}^k$.

The equation (15) is rewritten as follows

$$e^{k+2} = L_z^k - p_d^{k+2} = M_{\tau} (\dot{\tau}^k - \tau_{n}^k) + L_{\vec{d}}^k \quad (21)$$

For convenience, we define:

$$S_{\tau} = S_{\tau}(\ddot{z}) \quad (22)$$

From Lemma 1, it is obvious that $M^k_{\tau}$ is bounded with $m_{\tau}$ and $\vec{m}_{\tau}$. Noting $\hat{W}^k = \hat{W}^k - W_r^k$, and substituting (19) and (20) into (21), we obtain

$$e^{k+2} = M_{\tau} (k_{pd}\dot{\tau}^k + k_d \epsilon^{k-1}) + M_{\tau} \hat{W}^k S_{\tau}^k + \tau_{dp} \quad (22)$$

where $\tau_{dp} = -M_{\tau} \epsilon^k + L_{\vec{d}}^k$.

It is easy to show that $\|\epsilon_r(\ddot{z})\| \leq \|M_{\tau} \epsilon^k\| + \|L_{\vec{d}}^k\| \leq \vec{m}_{\tau} \epsilon_r + \vec{r}_d := \tau_{dp}$.

The error equation in (22) can be converted to:

$$e^{k+2} + M_{\tau} k_{pd}\dot{\tau}^k + M_{\tau} k_d \epsilon^{k-1} = M_{\tau} \hat{W}^k S_{\tau}^k + \tau_{dp} \quad (23)$$
There defines a new error function as below
\[ e^{k+2} = e^{k+2} + M^e_k k_pe^k - M^e_k d^{k-1} \]
and thus, the new error function equation is obtained
\[ e^{k+2} = M^e_k \hat{W}^k S^k_e + s_d^{k+1} \] (24)
To improve tracking performance, the neural net weight adaptive law \( \Delta \hat{W}^k = \hat{W}^{k+1} - \hat{W}^k \) is tuned using both a tracking error and a critic signal, therefore, critic control algorithm is introduced in the next subsection.

4.3. Critic RBFNN Control
To achieve optimal control performance and high-quality trajectory tracking performance, we extend our recent research results [27] from SISO nonlinear control using neural networks method to MIMO nonlinear control using a novel adaptive RBFNN method. And the adaptive RBFNN controller in (10) is designed for the robot manipulators in (10).
Based on tracking error \( e_k = p^k - p^*_k \) and error function \( e^{k+2}_k \), we define an utility function vector \( r_k \in \mathbb{R} \) represented the current system-performance index as
\[ r_k = \beta_d e^{k+2} \] (25)
where \( e^{k+2}_k = e^{k+2} + g_1 e^k - g_2 e^{k-1}, g_1 > 0 \) and \( g_2 > 0 \) are error coefficients.
The long-term system-performance measure or the strategy utility function \( Q^k \) is defined using
\[ Q^k = \beta_0^N e^{k+1}_k + \beta_0^{N-1} e^{k+2}_k + \cdots + \beta_0^1 e^{k+N} + \cdots \] (26)
where \( 0 < \beta_0 < 1 \) is a system design parameter, \( N_d \) is a horizon.
Thus, the equation (26) can also be expressed as
\[ Q^k = \min_{z^k} [\beta_0 Q^{k-1} - \beta_0^{N+1} r_k] \]
The RBFNN in (6) is applying to approximate the strategy utility function vector \( Q^k \) as
\[ Q^k = W^k_d S_d(z^k) + e_d(z^k) \] (27)
where \( W^k_d \in \mathbb{R}^{N_d \times n} \) is weight matrix, \( N_d \) is number of neuron in hidden layer of the critic RBFNN, \( S_d(z^k) \in \mathbb{R}^{N_d} \) is regressor matrix, \( ||e_d(z^k)|| \leq e^*_d \), \( e^*_d > 0 \) is a critic approximation error, and \( \Omega_z \) is a sufficient large compact set.

\[ z^k = [p^{kT}, p^{k-1T}, \tau^k, \tau^{k-1T}]^T \]

Because there is a mapping between the states \( p^k, \tau^k \) and \( z^k \), such that the vector \( z^k \) is selected as the critic RBFNN input in (27). The approximation matrix \( \hat{W}^k_d \in \mathbb{R}^{N_d \times n} \) of the critic weight \( W^k_d \) is estimated as
\[ \hat{Q}^k = \hat{W}^k_d S_d(z^k) \] (28)
where \( \hat{Q}^k \in \mathbb{R}^{N_d \times n} \) is the critic signal, and we select the desired critic signal \( \hat{Q}^k_d = \theta^k_0 \) at each step.
For convenience, \( S^k_e = S_d(z^k) \).
To further analyze the strategic utility function \( Q^k \), we define a prediction error vector as
\[ e^k = \hat{Q}^k - \beta^k_0 \hat{Q}^{k-1} + \hat{p}^{N+1}_d \] (29)
To minimize the prediction error, we design the critic RBFNN weight matrix update rule \( \Delta \hat{W}^k_d \) in (28) as follows
\[ \hat{W}^k_d = \hat{W}^k_d + \Delta \hat{W}^k_d = \hat{W}^k_d - \Gamma \hat{e}^k_{-d} S^k_d \] (30)
where \( \Gamma = \gamma_d I_{[N_d]} \in \mathbb{R}^{N_d \times N_d} \) is a diagonal critic learning rate matrix with \( \gamma_d > 0 \).
Substitute (28) and (29) into (30), the approximation matrix of the \((k + 1)\) step is updated as
\[ \hat{W}^{k+1}_d = \hat{W}^k_d + \Delta \hat{W}^k_d \]
\[ = \hat{W}^k_d - \Gamma \hat{e}^k_{-d} S^k_d + \hat{e}^k_{-d} \] (31)
\[ - \beta^k_0 \hat{W}^{k-1}_{-d} S^{k-1}_{-d} - \hat{e}^{N+1}_d \hat{e}^{k+1}_d^T \] (32)
Noting that the adaptive neural net algorithm in (24) is tracking error of the \((k + 2)\)th step, then, we can derive the \(k\)th step error by defining \( k_2 = k + 2 \) that
\[ e^k = M^{k^2} \hat{W}^{k^2}_{-d} S^{k^2}_{-d} + \tau^{k^2}_d \]
where \( m_r \leq ||M^k|| \leq \tilde{m} \), according Lemma 1, and
\[ e^k = e^k + M^{k^2} k_p e^k - M^{k^2} k_d e^{k-1} \]
Based on the error \( e^k = Q^k - \hat{Q}^k \) and the tracking error \( e^*_k \), the actor RBFNN update rule for (20) is given by
\[ \hat{W}^{k+1}_r = \hat{W}^{k}_r + \Delta \hat{W}^{k}_r \]
\[ = \hat{W}^{k}_r + \Gamma_r S^{k}_r [e^k - \hat{Q}^k]^T \] (33)
where \( \Gamma_r = \gamma_r I_{[N_r]} \in \mathbb{R}^{N_r \times N_r} \) is a diagonal action system learning rate matrix with \( \gamma_r > 0 \), and \( \hat{Q}^k_r = \beta_r e^{N+1}_r \hat{Q}^k \).

4.4. Stability Analysis
It has been shown that there exists an ideal control input \( \tau^k_d(z^k) \), which can guarantee the predictor error \( \epsilon^{k+2} = 0 \), if the unknown disturbance \( \epsilon^{k+1} \) = 0. Because all assumptions are only valid in compact set \( \Omega_\zeta \), all outputs and inputs of the robot system must be proved that they will remain in these compact sets in all the time.
indeed. Therefore, we can suppose that all past control inputs \(r_{k-1}^d\) are in \(\Omega_r\), all current output \(p^d\) and all past outputs \(p_{k-1}^d\) are in \(\Omega_p\), all past RBFNN weight errors \(\Delta \hat{W}_{d}^{k-1}\), \(\Delta \hat{W}_{d}^{k-2}\) are in \(\Omega_{\hat{W}_d}\) and \(\Omega_{\hat{W}_d}\), respectively.

In this subsection, we will focus on to prove that all these conditions still hold after time instant \(T\), and further prove the trajectory tracking error converges into a small neighbourhood of zero.

For analysing the system stability in (14), the theorem is presented to show how the controller parameters and adaptive parameters can appropriately be chosen to achieve the satisfied performance and optimality of the closed-loop robot system.

Choose a positive definite Lyapunov function \(V^k\) for the system (14) as

\[
V^k = V_1^k + V_2^k + V_3^k
\]

\[
= tr\{\hat{W}_d^{k+1} \Gamma_d^{-1} \hat{W}_d^k\} + \frac{1}{\rho_d}(\hat{W}_d^{k+1} S_d^{k-1})^T (\hat{W}_d^{k+1} S_d^{k-1}) + \frac{1}{\rho_t} \sum_{j=0}^n {tr}\{\hat{W}_d^{k-2+j} \Gamma_t^{-1} \hat{W}_d^{k-2+j}\}
\]

(34)

where \(\Gamma_d\) and \(\Gamma_t\) are diagonal learning rate matrices for critic RBFNN and actor RBFNN in (30) and (33), respectively. \(\hat{W}_d^k = \hat{W}_d^k - W_d^k\), \(\hat{W}_d^k - W_d^{k-1}\), \(\rho_d\) and \(\rho_t\) are positive design constants.

Noting the Lyapunov function \(V^k\) in (34) is consisted of \(V_1^k\), \(V_2^k\) and \(V_3^k\). According to the 4th error function in (32), we know that \(V^k\) contains the system tracking error \(e^k\), the strategic utility function error \(e_{d^k}\) and the design parameters.

The first difference of (34) is given by

\[
\Delta V^k = \Delta V_1^k + \Delta V_2^k + \Delta V_3^k
\]

Note (31), the first term of (35) is given by

\[
\Delta V_1^k = tr\{\hat{W}_d^{k+1} \Gamma_d^{-1} \hat{W}_d^k\} - tr\{\hat{W}_d^k \Gamma_d^{-1} \hat{W}_d^k\}
\]

\[
= tr\{\hat{W}_d^{k+1} \Gamma_d^{-1} \hat{W}_d^k\} - tr\{\hat{W}_d^k \Gamma_d^{-1} \hat{W}_d^k\}
\]

\[
= \frac{1}{\rho_d}(\hat{W}_d^{k+1} S_d^{k-1})^T (\hat{W}_d^{k+1} S_d^{k-1}) + \frac{1}{\rho_t} \sum_{j=0}^n {tr}\{\hat{W}_d^{k-2+j} \Gamma_t^{-1} \hat{W}_d^{k-2+j}\}
\]

(36)

where \(\hat{W}_d^k = \hat{W}_d^k + W_d^k\) and \(\hat{W}_d^k = \hat{W}_d^k - W_d^k\).

For convenience to analyse, we define

\[
\mathcal{A}^k = W_d^{k+1} S_d^{k-1}, \quad B^k = W_d^{k+1} S_d^{k-1} - \beta_0 W_d^{k-1} S_d^{k-1}
\]

\[
C^k = \beta_0 \hat{W}_d^{k-1} S_d^{k-1}, \quad D^k = M_d^k \hat{W}_d^{k-1} S_d^{k-1}
\]

(37)

\[
E^k = \beta_0 \beta_0^{N_0+1}, \quad F^k = E^k E^k, S^0_d \Gamma_d S^0_d = c
\]

then, the equation (36) is rewritten as

\[
\Delta V_1^k = -2\mathcal{A}^k (\mathcal{A}^k + B^k + F^k - C^k)^T + c(\mathcal{A}^k + B^k + F^k - C^k)^T
\]

\[
\leq -(1 - c)(\mathcal{A}^k + B^k + F^k - C^k)^T
\]

(38)

According the equation (25), we have

\[
F^k = E^k E^k = E^k (e^{k+1} + g_1 e^{k} - g_2 e^{k-1})
\]

Noting that

\[
F^k = E^k E^k = E^k (e^{k+1} + g_1 e^{k} - g_2 e^{k-1})
\]

Considering (32), we define a new vector \(\mathcal{F}_e^k = \mathcal{F}_e^k\) as

\[
\mathcal{F}_e^k = E^k (e^{k} + M_d^k k_{pd} e^{k} - M_d^k k_{d} e^{k-1})
\]

(39)

Analyzing vector \(\mathcal{F}_e^k\) and vector \(F^k\), it is easy to obtain

\[
F^k T \mathcal{F}_e^k = E^k T (e^{k} + M_d^k k_{pd} e^{k} - M_d^k k_{d} e^{k-1})
\]

(40)

where \(M_d^k = M_d^k T M_d^k\).

We see \(M_d^k\) is a symmetric positive definite matrix, and it is bounded with \(m_0^2 \leq ||M_d^k|| \leq m^2_0\).

**Theorem 1.** According the properties of symmetric positive definite matrix, the eigenvalues \(\lambda_i\) of \(M_d^k\) are positive values, let us define \(\lambda_{(i)}^{\max} = \max(\lambda_i)\) and \(\lambda_{(i)}^{\min} = \min(\lambda_i)\), \(i = 1, 2, \ldots, n\). According matrix norm property, we have \(n \lambda_{(i)}^{\max} \leq ||M_d^k|| = \sum_{i=1}^n \lambda_i(T_i^T M_d^k T_i) \leq n \lambda_{(i)}^{\max} \).

Define

\[
G_1^k = k_0^2 M_d^k - g_1^2 I_{[n]}, \quad G_2^k = k_0^2 M_d^k - g_2^2 I_{[n]}
\]
It is noted that $F_k T^k \leq F_k^T T^k$ can be satisfied, when the error coefficients $g_1$ and $g_2$ are given. Matrices $H_k$, $I_k$ and $O_k$ are symmetric positive definite, and they need to satisfy the following conditions:

\[
s_1^k \leq \frac{k^2 m}{\sqrt{n}}, \quad s_2^k \leq \frac{k^2 m}{\sqrt{n}}
\]

Substituting (32) to $F_k T^k = E_k \tilde{e}_k^T \tilde{e}_k^T$ in (40), we get

\[
F_k T^k \leq F_k^T T^k = E_k^T \tilde{e}_k^T e_k = E_k^T D_k^T D_k^T
\]

\[
+ 2E_k^T D_k^T r_{dp} + E_k^T r_{dp} r_{dp}^T
\]

\[
\leq 2E_k^T D_k^T D_k^T + 2E_k^T r_{dp}^2
\]

Substituting (41) to (38), we can obtain

\[
\Delta V_k^1 \leq -(1 - c)(A_k + B_k + F_k - C_k)^T
\]

\[
\times (A_k + B_k + F_k - C_k) - A_k^T A_k
\]

\[
+ 3B_k^T B_k + 3C_k^T C_k
\]

\[
+ 6E_k^T D_k^T D_k + 6E_k^T r_{dp}^2
\]

Taking the second difference $\Delta V_k^2$ of (35), we get

\[
\Delta V_k^2 = \frac{1}{\rho_d} [(W_d^T S_d) + (W_d^T S_d)^T]
\]

\[
\times (W_d^T S_d)^T
\]

\[
- (W_d^{k-1} S_d^{k-1})^T (W_d^{k-1} S_d^{k-1})^T
\]

\[
= \frac{1}{\rho_d} (A_k^T A_k - 1_{\beta_0} C_k^T C_k)
\]

The third difference $\Delta V_k^3$ of (35) along (28), (30) and (33) is given by

\[
\Delta V_k^3 = \frac{2}{\rho_t} (e_k^T - \tilde{Q}_k^T)^T W_d^{k-1} S_d^{k-1}
\]

\[
+ \frac{b}{\rho_t} (e_k^T - \tilde{Q}_k^T)^T (e_k^T - \tilde{Q}_k^T)
\]

where $b = S_d^{k-1} \Gamma_k r^{k-1}_d$. It is noted that

\[
\tilde{Q}_k^T = \beta_d^{k-1} \tilde{Q}_k^T = e_k^T W_d^{k-1} S_d^{k-1}
\]

Defining $U_k^k = s_d^{k-1} - E_k^T W_d^T S_d$ and substituting (32) into (44), we have

\[
\Delta V_k^3 \leq - \frac{1}{\rho_t} \left( D_k^T + U_k^k - E_k^T A_k^T \right)^T
\]

\[
\times (M_k^{k-1} - bI_n)
\]

\[
\times (D_k^T + U_k^k - E_k^T A_k^T)
\]

\[
- D_k^T \frac{1}{\rho_t} M_k^{k-1} D_k^T
\]

\[
+ 2\bar{U}_k^k \frac{1}{\rho_t} M_k^{k-1} \bar{U}_k^k
\]

\[
+ 2E_k^T A_k^T \frac{1}{\rho_t} M_k^{k-1} A_k^T
\]

Combining equations (42), (43) and (45) in (35), we further obtain that

\[
\Delta V_k^k \leq - A_k^T (I_n - \frac{1}{\rho_d} I_n) - 2E_k^T \frac{1}{\rho_t} M_k^{k-1} A_k^T
\]

\[
- \left( 1 - \frac{1}{\rho_d} b_0 \right) - 3C_k^T C_k - D_k^T \left( \frac{1}{\rho_t} M_k^{k-1} \right)
\]

\[
- 6E_k^T I_n) D_k^T
\]

\[
- (1 - c)(A_k + B_k + F_k - C_k)^T
\]

\[
\times (A_k + B_k + F_k - C_k)
\]

\[
- \frac{1}{\rho_t} (D_k^T + U_k^k - E_k^T A_k^T)^T (M_k^{k-1} - bI_n)_n)
\]

\[
\times (D_k^T + U_k^k - E_k^T A_k^T) + ||J||^2
\]

where

\[
||J||^2 = 3B_k^T B_k + 6E_k^T r_{dp}^2 + 2U_k^k \frac{1}{\rho_t} M_k^{k-1} U_k^k
\]

We see that

\[
c = S_d^{k-1} \Gamma_k r^{k-1}_d = \gamma_d S_d^{k-1} S_d^{k-1} < \gamma_d N_d
\]

\[
b = S_d^{k-1} \Gamma_k r^{k-1}_d = \gamma_s S_d^{k-1} S_d^{k-1} < \gamma_s N_t
\]

And thus,

\[
||J||^2 \leq ||J||^2 = (6 + 6\beta_0 + 4E_k^T \beta_0^2) ||W_d||^2 N_d
\]

\[
+ \frac{4r_{dp}^2}{\rho_t^2 m_t^2} + 6E_k^T r_{dp}^2
\]

Note Theorem 1 and Lemma 2, we know $M_k^{k-1}$ is also a symmetric positive definite matrix, and is bounded with $m_t^2 \leq ||M_k^{k-1}\| \leq \bar{m}_t^2$. Then, we define the eigenvalues of $M_k^{k-1}$ are $\lambda_i^k$, $i = 1, 2, \ldots, n$. It is obvious that $\lambda_i^k > 0$. We define $\lambda_{max} = max(\lambda_i^k)$ and $\lambda_{min} = min(\lambda_i^k)$, $i = 1, 2, \ldots, n$. Then, $n\lambda_{min} \leq ||M_k^{k-1}\|^2 = \sum_{i=1}^n A_i (M_k^{k-1} \lambda_i^k)^2 \leq n\lambda_{max}^2$.

Defining

\[
H_k = (1 - \frac{1}{\rho_d}) I_n - 2E_k^T \frac{1}{\rho_t} M_k^{k-1}
\]

\[
I_k = \frac{1}{\rho_t} M_k^{k-1} - 6E_k^T I_n)
\]

\[
O_k = M_k^{k-1} - bI_n
\]

The matrices $H_k$, $I_k$ and $O_k$ are symmetric positive definite, and they need to satisfy the following conditions:
\[
(1 - \frac{1}{\rho_d}) - 2\varepsilon\frac{1}{\rho_d} \frac{\dot{m}_d^*}{\sqrt{n}} > 0 \\
\frac{1}{\rho^*} - 6\varepsilon \frac{\dot{m}_d^*}{\sqrt{n}} > 0, \quad 1 - b \frac{\dot{m}_d^*}{\sqrt{n}} > 0
\]

**Theorem 2.** The optimal adaptive RBFNN control in (20) with RBFNN weight adaptation law (30) and (33) for the robot manipulators in (10). All signals in the closed-loop system are UUB, we provide the design parameters selected as follows:

\[
\begin{align*}
0 < \rho_d &\leq \frac{1}{3\beta_0} \\
0 < \beta_0 &< \frac{\sqrt{3}}{3} \\
\frac{2\varepsilon\dot{m}_d^*}{(1 - \frac{1}{\rho_2})\sqrt{n}} < \rho_2 < \frac{\dot{m}_d^*}{6\varepsilon\sqrt{n}} \\
0 < \gamma_d &< \frac{1}{N_d} \\
0 < \gamma_r &< \frac{\sqrt{n}}{N_d\dot{m}_d^*}
\end{align*}
\]

Assuming that the condition set above are satisfied, we have

\[
\Delta V^k \leq -\mathcal{K}^T ((1 - \frac{1}{\rho_d})I_{10} - 2\varepsilon\frac{1}{\rho_d} M_1^T)\mathcal{K}^k \\
- \frac{1}{\rho_d\beta_0^2} - 3|C^T C| - \mathcal{D}^T (1 - \frac{1}{\rho_d})M_2^{-1} \\
- 6\varepsilon I_{10} \mathcal{D}^k + ||Jf||^2
\]

Existing invertible matrix \(\mathcal{P}_H\) and \(\mathcal{P}_I\) make \(H^k = \mathcal{P}_H^T \mathcal{P}_H\) and \(I^k = \mathcal{P}_I^T \mathcal{P}_I\), accordingly, \(\Delta V^k \leq 0\) can be well satisfied under the following conditions:

\[
\begin{align*}
||Jf||^2 < ||\mathcal{P}_H^T \mathcal{A}^k||^2 (\mathcal{P}_H, \mathcal{A}^k) \\
||\mathcal{A}^k||^2 > ||Jf||^2 ||\mathcal{P}_H||^{-1}
\end{align*}
\]

or

\[
\begin{align*}
||Jf||^2 < ||\mathcal{P}_I^T \mathcal{D}^k||^2 (\mathcal{P}_I, \mathcal{D}^k) \\
||\mathcal{D}^k||^2 > ||Jf||^2 ||\mathcal{P}_I||^{-1}
\end{align*}
\]

Introducing a discrete-time delay factor \(z^{-1}\) into (24), we have

\[
e^k = (I_{10} + M_1^k \dot{z} \dot{z}^{-2} - M_2^k \dot{z} \dot{z}^{-2})^{-1} e^1
\]

Noting (49) and (50), we know there exists a finite running step \(K_0\), which makes \(||\mathcal{A}||^2 \leq ||Jf||^2 ||\mathcal{P}_H||^{-1}\) or \(||\mathcal{D}||^2 \leq ||Jf||^2 ||\mathcal{P}_I||^{-1}\), and makes \(||e^k|| \leq ||e^1||\) under \((I_{10} + M_1^k \dot{z} \dot{z}^{-2} - M_2^k \dot{z} \dot{z}^{-2})^{-1}\) being Hurwitz-stable for all \(k > K_0\).

From the definition of \(\mathcal{A}^k\) and \(\mathcal{D}^k\), the boundedness of \(\dot{w}_f^k S_d^k\) and \(\dot{w}_r^k S_r^k\) can be deduced. \(\dot{w}_f^k S_d^k\) and \(\dot{w}_r^k S_r^k\) are bounded, then, \(\dot{w}_f S_d^k\) and \(\dot{w}_r S_r^k\) are also bounded.

We see that the boundedness of \(\dot{w}_f S_d^k\) and \(\dot{w}_r S_r^k\) further implies that \(\dot{w}_f S_d^k\) and \(\dot{w}_r S_r^k\) are bounded. Let the boundedness of \(M_1^k\) and \(\tau_d\), we know the tracking error \(e_1^k\) is bounded as

\[
||e^1||^2 = e_1^T e_1^1 \leq 2\dot{w}_f S_d^k + 2\dot{w}_r S_r^k \\
< 2||Jf|| ||\mathcal{P}_H|| ||e^1|| + 2\tau_d
\]

(51) or, we can get

\[
||e^k|| \leq ||e^1|| \leq \sqrt{2||Jf|| ||\mathcal{P}_H|| ||e^1||} + 2\tau_d
\]

the proof is complete.

5. Simulation Studies

To verify the efficacy of the above developed control approach, a 2-DOF rigid robot manipulator as a testing example, is put foreword in this section.

5.1. Robot Manipulator Dynamics Model

The following parameters of the robot manipulator are given as follows: The mass are \(m_1 = m_2 = 1.0\) kg, the length are \(l_1 = l_2 = 0.2\) m, the inertia are \(I_1 = I_2 = 0.003\) kgm\(^2\), the distance are \(l_1 = l_2 = 0.1\) m.

The dynamics of the robot manipulator with \(G(q) = 0_{31}\) is given as follows:

\[
M(q) = \begin{bmatrix}
M_{11} & M_{12} \\
M_{21} & M_{22}
\end{bmatrix}
\]

\[
C(q, q') = \begin{bmatrix}
C_{11} & C_{12} \\
C_{21} & C_{22}
\end{bmatrix}
\]

(53)

where

\[
\begin{align*}
M_{11} & = m_1 l_1^2 + m_2 (l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2)) + I_1 + I_2 \\
M_{12} & = M_{21} = m_2 l_2^2 + l_1 l_2 \cos(q_2) + I_2 \\
M_{22} & = m_2 l_2^2 + I_2 \\
C_{11} & = -m_2 l_1 l_2 \sin(q_2) q_2 \\
C_{12} & = -m_2 l_1 l_2 \sin(q_2) (q_1 + q_2) \\
C_{21} & = m_2 l_1 l_2 \sin(q_2) q_1 \\
C_{22} & = 0
\end{align*}
\]

The external disturbance may be a smaller or a larger amplitude force torque \(\tau_{ds}\) or \(\tau_{db}\) in testing, respectively. They are assumed as

\[
\tau_{ds} = [0.05 \cos(0.01t) \cos(q_1), 0.05 \cos(0.01t) \cos(q_2)]^T \\
\tau_{db} = [40 \cos(0.01t) \cos(q_1), 40 \cos(0.01t) \cos(q_2)]^T
\]
The desired trajectory \( q_d \) is assumed as:

\[
q_d = [q_{1d}, q_{2d}]^T = \begin{bmatrix} 1.5 + 0.5(\sin(0.3t) + \sin(0.2t)) \\ 1.5 + 0.5(\cos(0.4t) + \sin(0.3t)) \end{bmatrix}
\]

### 5.2. Test Results

The initial states of the robot manipulate is assumed in (53), \( q(0) = [0, 0]^T \) and \( \dot{q}(0) = [0, 0]^T \). We construct the critic RBFNN \( \hat{W}_d^c \) \( S_d^c \) approximating the strategic utility function by using \( N_d = 1024 \) with all the centres of Gaussian function are evenly in \([-1; 1]\) and the widths=1, while the actor RBFNN \( \hat{W}_a^s \) \( S_a^s \) approximating the system tracking error using \( N_d = 4096 \) with all the centres of Gaussian function evenly in \([-1; 1]\) and the widths=1. The design parameters are chosen as \( \gamma_d = 0.0005, \gamma_1 = 0.0001, \beta_d = 0.8, N_a = 3, \beta_0 = 0.5, k_p = 0.5, k_d = 120 \). The initial weights \( \hat{W}_d(0) = 0_{2 \times N_d}, \hat{W}_a(0) = 0_{2 \times N_a} \), and we choose the controller sampling interval \( T = 0.01 \).

To show the effectiveness, we have done the relevant comparative analysis for trajectory tracking accuracy and capability for the robot manipulator with \( \tau_d \) and \( \tau_{db} \), e.g., PD control and robust control in Figs. 1-10.

In contrast with PD control, the PD controller \( \tau^k = -k_p e^k - k_d (e^k - e^{k-1}) \) is applied.

And in contrast with robust control based on bounded observer, the controller \( \tau^k = sat(K_1 e^k + K_2 f(k)) \) is applied, where \( K_1 \) and \( K_2 \) are gain matrices, and \( f(k) \) is the estimation of all uncertain terms \( f(k) \). The parameters \( K_1 = [K_11 K_12] \) and \( K_2 \) obtained by using LMIs theory as \( K_11 = [-10; -5; -2; -10], K_12 = [-10; -5; -2; -10], \) and \( K_2 = [0.0150; 0.0350; 0.0505] \), respectively.

Fig. 1-2 show tracking curves of \( q_1 \) and \( q_2 \), Fig. 3-4 show control input curves of control inputs \( \tau_1 \) and \( \tau_2 \). Fig. 5-6 show the proposed method can be depicted by designing the critic utility function \( \hat{Q} \). Fig. 7-8 shows critic RBFNN weight norm \( ||W_d|| \) and actor...
In this paper, the adaptive RBFNN control has been investigated for a class of rigid robot manipulators with uncertain dynamics to optimize control performance in discrete time. The control system is designed with the actor RBFNN and the critic RBFNN to eliminate the strategic utility function and system tracking error. Con-
control laws are real-time adaptive and are tuned online. Based on the output feedback method, the control method to compensate for the influences of dynamics uncertainties and external disturbance, not only guarantees the system is Lyapunov stability, but also achieves the optimal trajectory tracking performance.

Acknowledgements

This work was supported in part by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/L026856/2.


