Paper:

Yang, R., Yang, C., Chen, M. & Annamalai, A. (2017). Discrete-time Optimal Adaptive RBFNN Control for Robot Manipulators with Uncertain Dynamics. *Neurocomputing*
http://dx.doi.org/10.1016/j.neucom.2016.12.048

Discrete-time Optimal Adaptive RBFNN Control for Robot Manipulators with Uncertain Dynamics

Runxian Yang[a,b,c], Chenguang Yang[c,*], Mou Chen[a], Andy SK Annamalai[d]

[a]*College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China*
[b]*College of Electric and IT, Yangzhou Polytechnic Institute, Yangzhou, China*
[c]*Zienkiewicz Centre for Computational Engineering, Swansea University, Swansea, UK*
[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 performed 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 nonliearity, 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

☆This research is a collaborative effort
*Corresponding author.
Email address: cyang@theiet.org (Chenguang Yang)

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 noncausal 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,

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

where $q \in \Re^n$ denotes the joint position, and $\dot{q} \in \Re^n$ is the joint velocity, $\ddot{q} \in \Re^n$ denotes the joint acceleration, $M(q) \in \Re^{n\times n}$ is the symmetric positive definite inertia matrix, $C(q,\dot{q}) \in \Re^{n\times n}$ is the Coriolis-Centrifugal torque matrix, $G(q) \in \Re^n$ denotes the gravity torque vector; $\tau \in \Re^n$ is the vector of control input torque, $\tau_d \in \Re^n$ 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* $2C(q,\dot{q}) - \dot{M}(q) \in \Re^{n\times n}$ *is a skew-symmetric matrix, such that*

$$x^T[\dot{M}(q) - 2C(q,\dot{q})]x = 0, \forall x \in \Re^n \qquad (2)$$

Property 2. *The* $M(q)$*, a symmetric and positive definite inertia matrix, is uniformly bounded, there* $\underline{m} > 0$

Table 1: NOMENCLATURE

Notation	Description
$\|\| \cdot \|\|$	the Euclidean norm of vectors and induced norm of matrices
$a := b$	a is defined as b
$[\,]^T$	the transpose of a vector or a matrix
$[\,]^{-1}$	the inverse of a n-order reversible matrix
$\mathbf{0}_{[p]}$	p-dimensional zero vector
$\mathbf{I}_{[m]}$	m-dimensional identity matrix
W^*	the ideal neural net weight matrix at the kth step
$\hat{W}^k$	the estimate of W^*
$\tilde{W}^k$	$\hat{W}^k - W^*$, the weight estimate error
q	the n-dimensional joint position
$\dot{q}$	the n-dimensional joint velocity
$\ddot{q}$	the n-dimensional joint acceleration
q_d	the n-dimensional ideal joint position
$M(q)$	the $n \times n$ dimensional symmetric positive definite inertia matrix
$C(q, \dot{q})$	the $n \times n$ dimensional Coriolis-Centrifugal torque matrix
$G(q)$	the n-dimensional gravity torque vector
τ	the n-dimensional vector of control input torque
τ_d	the n-dimensional external force torque
T	the sampling time interval
t_k	the sampling time, and $t_k = kT$
p^k	the sampled joint angle at time t_k, and $p^k = q(t_k)$
v^k	the sampled joint velocity at time t_k, and $v^k = \dot{q}(t_k)$
τ^k	the sampled joint force at time t_k
τ_d^k	the sampled external disturbance force at time t_k
p_d^k	the sampled ideal joint position at time t_k
ξ^k	ξ^k is defined as $p^k + Tv^k$
e^k	the trajectory error, e^k is defined as $p^k - p_d^k$
τ_n^*	the ideal control input
Q^k	the strategic utility function
Γ_d	the diagonal critic learning rate matrix
Γ_τ	the diagonal action learning rate matrix
k_p	the scaling factor, the proportion parameter
k_d	the scaling factor, the integral parameter
k_{pd}	the scaling factor, k_{pd} is defined as $k_p + k_d$
e_1^k	the new error function

and $\bar{m} > 0$ are constants, and thus, $M(q)$ satisfies the following inequality

$$\underline{m} \leq \|M(q)\| \leq \bar{m} \tag{3}$$

Property 3. *The matrix $C(q, \dot{q})$ and the vector $G(q)$ are bounded by $\|C(q, \dot{q})\| \leq k_c \|\dot{q}\|$, and $\|G(q)\| \leq k_g$, respectively, where k_c and k_g are positive constants.*

2.1. RBFNN Constructure

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]:

$$F(W, z) = W^T S(z), \quad W \in \Re^{N_s \times N_o}, \quad S(z) \in \Re^{N_s} \tag{4}$$

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

$$s_i(z) = e^{-\|b_i - c_{ij}\|/2b_i^2} \tag{5}$$

where $i = 1, 2, \cdots, N_n$, $j = 1, 2, \cdots, N_s$, 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 $\varphi(z) : \Omega_z \rightarrow R$ over a compact set $\Omega_z \subset R^{N_n}$, applying RBFNN (4) to approximate $\varphi(z)$, if N_s is sufficiently large, a set of ideal bounded weights W^* exist, and we have

$$\varphi(z) = W^{*T} S(z) + \mu(z) \tag{6}$$

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

$$S(z)^T S(z) < N_s \tag{7}$$

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,

$$\varphi(z) \approx \hat{W}^T S(z) \text{ or } \hat{\varphi}(z) = \hat{W}^T S(z) \tag{8}$$

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 be T, and the sampled angle at time $t_k = kT$ is p^k for the n−DOF rigid robot manipulators in (1).
Define $p^k = q(t_k) \in \Re^n$ and $v^k = \dot{q}(t_k) \in \Re^n$, the dynamic equation (1) in the continuous-time can be discretized [29, 30, 24] as

$$(M(\xi^k)/T)(v^{k+1} - v^k) = (M(\xi^k) - M(p^k))v^k - f(p^k, v^k) + \tau^k + \tau_d^k \tag{9}$$

where $M(\xi^k) \in \Re^{n\times n}$ is an inertia matrix with $\xi^k = p^k + Tv^k \in \Re^n$, $f(p^k, v^k) = C(p^k, v^k)v^k + G(p^k) \in \Re^n$, $C(p^k, v^k) \in \Re^{n\times n}$ is Coriolis-Centrifugal torque matrix and $G(p^k) \in \Re^n$ 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 $\underline{m} \le \|M(\xi^k)\| \le \bar{m}$ with $\underline{m} > 0$ 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,

$$\begin{cases} p^{k+1} = p^k + Tv^k \\ v^{k+1} = [(1+T)\mathbf{I}_{[n]} - TM^{-1}(\xi^k)M(p^k) \\ \quad - TM^{-1}C(p^k, v^k)]v^k - TM^{-1}(\xi^k)G(p^k) \\ \quad + TM^{-1}(\xi^k)\tau^k + TM^{-1}(\xi^k)\tau_d^k \end{cases} \tag{10}$$

where $\tau^k \in \Re^n$ and $p^k \in \Re^n$ are system input and output in discrete time, respectively. τ_d^k is bounded by an unknown constant $\bar{\tau}_d$ which makes $\|\tau_d^k\| \le \bar{\tau}_d$.
It is easy to know $M^{-1}(\xi^k)$ is also bounded, there $\underline{m}^* > 0$ and $\bar{m}^* > 0$ are constants, and thus, the inequality $\underline{m}^* \le \|M^{-1}(\xi^k)\| \le \bar{m}^*$ is satisfied.
The control objective of this paper is to synthesize an adaptive RBFNN control τ^k for system (10), then, all signals of the closed-loop robot system are bounded, and the joint position output p^k well tracks a bounded, ideal, reference trajectory $p_d^k \in \Omega_{p_d}^k$, 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 τ^k, while p^{k+1} is associated with p^k and v^k.
We rewrite the first equation of the robot model (10) as $p^{k+1} - p^k - Tv^k = \mathbf{0}_{[n]}$, and v^k is designed as $v^k = \frac{1}{T}(p^{k+1} - p^k)$. For the prediction $(k+2)$ step of the robot manipulator system, we can obtain

$$\begin{aligned} p^{k+2} &= p^{k+1} + Tv^{k+1} \\ &= [(2+T)\mathbf{I}_{[n]} - TM^{-1}(\xi^k)M(p^k) \\ &\quad - TM^{-1}(\xi^k)C(p^k, v^k)]p^{k+1} \\ &\quad - [(1+T)\mathbf{I}_{[n]} - TM^{-1}(\xi^k)M(p^k) \\ &\quad - TM^{-1}(\xi^k)C(p^k, v^k)]p^k \\ &\quad - T^2M^{-1}(\xi^k)G(p^k) \\ &\quad + T^2M^{-1}(\xi^k)\tau^k + T^2M^{-1}(\xi^k)\tau_d^k \end{aligned} \tag{11}$$

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

$$\begin{aligned} p^{k+1} &= [(2+T)\mathbf{I}_{[n]} - TM^{-1}(\xi^{k-1})M(p^{k-1}) \\ &\quad - TM^{-1}(\xi^{k-1})C(p^{k-1}, v^{k-1})]p^k \\ &\quad - [(1+T)\mathbf{I}_{[n]} - TM^{-1}(\xi^{k-1})M(p^{k-1}) \\ &\quad - TM^{-1}(\xi^{k-1})C(p^{k-1}, v^{k-1})]p^{k-1} \\ &\quad - T^2M^{-1}(\xi^{k-1})G(p^{k-1}) \\ &\quad + T^2M^{-1}(\xi^{k-1})\tau^{k-1} + T^2M^{-1}(\xi^{k-1})\tau_d^{k-1} \end{aligned} \tag{12}$$

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

$$\begin{aligned} L^k &= (2+T)\mathbf{I}_{[n]} - TM^{-1}(\xi^k)M(p^k) - TM^{-1}(\xi^k)C(p^k, v^k) \\ R^k &= (1+T)\mathbf{I}_{[n]} - TM^{-1}(\xi^k)M(p^k) - TM^{-1}(\xi^k)C(p^k, v^k) \\ M_\tau^k &= T^2M^{-1}(\xi^k), \quad G^k = G(p^k) \end{aligned}$$

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

$$\begin{aligned} p^{k+2} &= (L^kL^{k-1} - R^k)p^k - L^kR^{k-1}p^{k-1} \\ &\quad - L^kM_\tau^{k-1}G^{k-1} - M_\tau^kG^k + L^kM_\tau^{k-1}\tau^{k-1} \\ &\quad + M_\tau^k\tau^k + L^kM_\tau^{k-1}\tau_d^{k-1} + M_\tau^k\tau_d^k \end{aligned} \tag{13}$$

and we can define

$$\begin{aligned} L_{\underline{p}}^k &= (L^kL^{k-1} - R^k)p^k - L^kR^{k-1}p^{k-1} + L^kM_\tau^{k-1}\tau^{k-1} \\ L_G^k &= L^kM_\tau^{k-1}G^{k-1} + M_\tau^kG^k \\ L_d^k &= L^kM_\tau^{k-1}\tau_d^{k-1} + M_\tau^k\tau_d^k \end{aligned}$$

Furthermore, we rewrite (13) as

$$\begin{aligned} p^{k+2} &= L_{\underline{p}}^k - L_G^k + M_\tau^k\tau^k + L_d^k \\ &= \psi(p^{k-1}, p^k, v^k, v^{k-1}, \tau^k, \tau^{k-1}, \tau_d^{k-1}, \tau_d^k) \end{aligned} \tag{14}$$

Noting that $\psi(\cdot,\cdot,\cdot,\cdot,0,0)$ is continuous, such that all the arguments and continuously differentiable with respect to τ^k is continuous.

Lemma 1. *According (9) and (10), M_τ^k is symmetric, positive definite matrix, there $\underline{m}_\tau = T^2\underline{m}^*$ and $\bar{m}_\tau = T^2\bar{m}^*$ are positive constants, then, M_τ^k is bounded with $\underline{m}_\tau \le \|M_\tau^k\| = T^2\|M^{-1}(\xi^k)\| \le \bar{m}_\tau$.*
Therefore, it is easy to obtain that $\|L_d^k\| \le (3 + 2T + T\bar{m}^* k_c)\bar{m}_\tau\bar{\tau}_d := \bar{\tau}_d^*$.

4. Adaptive NN Control Design

4.1. Desired Control

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

$$e^{k+2} = p^{k+2} - p_d^{k+2} = L_{\underline{p}}^k - L_G^k + M_\tau^k \tau^k + L_d^k - p_d^{k+2} \quad (15)$$

It is noted that a ideal force torque control input τ_n^{*k} [31], such that

$$L_{\underline{p}}^k - L_G^k + M_\tau^k \tau_n^{*k} - p_d^{k+2} = 0 \quad (16)$$

or

$$\tau_n^{*k} = M_\tau^{k^{-1}}(p_d^{k+2} - L_{\underline{p}}^k + L_G^k) \quad (17)$$

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

$$\|e^{k+2}\| = \|L_d^k\| \le \bar{\tau}_d^* \quad (18)$$

We know the desired control τ_n^{*k} is not obtained with the unknown $M_\tau^{k^{-1}}$, $L_{\underline{p}}^k$ and L_G^k. Applying RBFNN to approximate the desired input by adaptive learn τ_n^{*k} will make tracking error $e^{k+2} = 0$ after 2 steps, if $\tau_d^k = 0$ and $\tau_d^{k-1} = 0$ in (15).

4.2. Actor RBFNN Control

From Section 2.1, the ideal weight matrix W_τ^* exists, we use a Gaussian function $S_\tau(\bar{z}^k)$ to approximate τ_n^{*k} as follows

$$\tau_n^*(\bar{z}^k) = W_\tau^{*T} S_\tau(\bar{z}^k) + \epsilon_\tau(\bar{z}^k) \quad (19)$$

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

$$\bar{z} = [p^{k^T}, p^{k-1^T}, v^{k^T}, v^{k-1^T}, \tau^{k-1^T}, p_d^{k+2^T}]^T \in \Omega_{\bar{z}}$$

$\Omega_{\bar{z}}$ is a sufficient large compact set and corresponds to Ω_{p_d}. The number of neuron in hidden layer of RBFNN is N_τ, and the ideal weight matrix $W_\tau^* =\in \Re^{N_\tau\times n}$ is given

$$W_\tau^* = [w_1, w_2, \cdots, w_r, \cdots, w_n] = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1j} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2j} & \cdots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{i1} & w_{i2} & \cdots & w_{ij} & \cdots & w_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{N_\tau 1} & w_{N_\tau 2} & \cdots & w_{N_\tau j} & \cdots & w_{N_\tau n} \end{bmatrix}$$

where $i = 1, 2, \cdots, N^\tau$, $j = 1, 2, \cdots, n$, w_r is the weight vector from all hidden layer neurons to the rth output τ_r, $r = 1, 2, \cdots, n$, the $S_\tau(\bar{z}^k) \in \Re^{N\tau}$ is the regressor matrix, $\|\epsilon_\tau(\bar{z}^k)\| \le \epsilon_\tau^*$, $\epsilon_\tau{}^* > 0$ is an approximation error. We known the ideal control $\tau_n^*(\bar{z}^k)$ can easily be bounded.
Noticing (15) and (17), we use RBFNN as an approximation of $\tau_n^*(\bar{z}^k)$ with proportion integral (PD) control to optimize control performance. Then, the system control input is given as:

$$\begin{aligned} \tau^k &= -k_p e^k - k_d(e^k - e^{k-1}) + \hat{\tau}_n(\bar{z}^k) \\ \text{or } \tau^k &= -k_{pd} e^k + k_d e^{k-1} + \hat{\tau}_n(\bar{z}^k) \quad (20) \\ \hat{\tau}_n(\bar{z}^k) &= \hat{W}_\tau^{k^T} S_\tau(\bar{z}^k) \end{aligned}$$

where, $k_p > 0$ and $k_d > 0$ are scaling factors, $k_{pd} = k_p + k_d > 0$. $\hat{W}_\tau^k \in \Re^{N_\tau\times n}$ is used to approximate unknown function $\tau_n^*(\bar{z}^k)$ in (19) with compact set $\Omega_{\bar{z}}$.
According to the equation (16), we have $p_d^{k+2} = L_{\underline{p}}^k - L_G^k + M_\tau^k \tau_n^{*k}$.
The equation (15) is rewritten as follows

$$e^{k+2} = L_{\underline{p}}^k - p_d^{k+2} = M_\tau^k(\tau^k - \tau_n^{*k}) + L_d^k \quad (21)$$

For convenience, we define:

$$S_\tau^k = S_\tau(\bar{z}^k), \quad \epsilon_p^k = \epsilon_p(\bar{z}^k)$$

From Lemma 1, it is obvious that M_τ^k is bounded with $\underline{m}_\tau$ and $\bar{m}_\tau$. Noting $\tilde{W}_\tau^k = \hat{W}_\tau^k - W_\tau^{*k}$, and substituting (19) and (20) into (21), we obtain

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

where $\tau_{dp}^k = -M_\tau^k \epsilon_\tau^k + L_d^k$.
It is easy to show that $\|\tau_{dp}^k\| \le \|M_\tau^k \epsilon_\tau^k\| + \|L_d^k\| \le \bar{m}_\tau \epsilon_\tau^* + \bar{\tau}_d^* := \tau_{dp}^*$
The error equation in (22) can be converted to:

$$e^{k+2} + M_\tau^k k_{pd} e^k - M_\tau^k k_d e^{k-1} = M_\tau^k \tilde{W}_\tau^{k^T} S_\tau^k + \tau_{dp}^k \quad (23)$$

There defines a new error function as below

$$e_1^{k+2} = e^{k+2} + M_\tau^k k_{pd} e^k - M_\tau^k k_d e^{k-1}$$

and thus, the new error function equation is obtained

$$e_1^{k+2} = M_\tau^k \tilde{W}_\tau^{k^T} S_\tau^k + \tau_{dp}^k \tag{24}$$

To improve tracking performance, the neural net weight adaptive law $\Delta\hat{W}_\tau^k = \hat{W}^{k+1} - \hat{W}^k$ 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 mehod. And the adaptive RBFNN controller in (20) is designed for the robot manipulators in (10),

Based on tracking error $e^k = p^k - p_d^k$ and error function e_1^{k+2}, we define an utility function vector $r^k \in \Re^n$ represented the current system-performance index as

$$r^k = \beta_d e_2^{k+2} \tag{25}$$

where $e_2^{k+2} = e^{k+2} + g_1 e^k - g_2 e^{k-1}$, $g_1 > 0$ and $g_2 > 0$ are error coefficients.

The long-term system-performance measure or the strategic utility function Q^k is defined using

$$Q^k = \beta_0^{N_a} r^{k+1} + \beta_0^{N_a-1} r^{k+2} + \cdots + \beta_0^{k+1} r^{N_a} + \cdots \tag{26}$$

where $0 < \beta_0 < 1$ is a system design parameter, N_a is a horizon.

Thus, the equation (26) can also be expressed as

$$Q^k = min_{\tau^k}[\beta_0 Q^{k-1} - \beta_0^{N_a+1} r^k]$$

The RBFNN in (6) is applying to approximate the strategic utility function vector Q^k as

$$Q^k = W_d^{*k^T} S_d(\underline{z}^k) + \epsilon_d(z^k) \tag{27}$$

where $W_d^* \in \Re^{N_d \times n}$ is weight matrix, N_d is number of neuron in hidden layer of the critic RBFNN, $S_d(\underline{z}^k) \in \Re^{N_d}$ is regressor matrix, $\|\epsilon_d(\bar{z}^k)\| \le \epsilon_d^*$, $\epsilon_d^* > 0$ is a critic approximation error, and $\Omega_{\underline{z}}$ is a sufficient large compact set.

$$\underline{z}^k = [p^{k^T}, p^{k-1^T}, v^{k^T}, v^{k-1^T}, \tau^{k-1^T}]^T$$

Because there is a mapping between the states p^k, v^k and $\underline{z}^k$, such that the vector $\underline{z}^k$ is selected as the critic RBFNN input in (27). The approximation matrix $\hat{W}_d^k \in \Re^{N_d}$ of the critic weight W_d^* is estimated as

$$\hat{Q}^k = \hat{W}_d^{k^T} S_d(\underline{z}^k) \tag{28}$$

where $\hat{Q}^k \in \Re^{N_d \times n}$ is the critic signal, and we select the desired critic signal $\hat{Q}_d^k = \mathbf{0}_{[n]}$ at each step.

For convenience, $S_d^k = S_d(\underline{z}^k)$.

To further analyze the strategic utility function Q^k, we define a prediction error vector as

$$e_d^k = \hat{Q}^k - \beta_0 \hat{Q}^{k-1} + \beta_0^{N_a+1} r^k \tag{29}$$

To minimize the prediction error, we design the critic RBFNN weight matrix update rule $\Delta\hat{W}_d^k$ in (28) as follows

$$\hat{W}_d^{k+1} = \hat{W}_d^k + \Delta\hat{W}_d^k = \hat{W}_d^k - \Gamma_d S_d^k e_d^{k^T} \tag{30}$$

where $\Gamma_d = \gamma_d \mathbf{I}_{[N_d]} \in \Re^{N_d \times N_d}$ 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

$$\begin{aligned}\hat{W}_d^{k+1} &= \hat{W}_d^k + \Delta\hat{W}_d^k \\ &= \hat{W}_d^k - \Gamma_d S_d^k (\hat{W}_d^{k^T} S_d^k \\ &\quad - \beta_0 \hat{W}_d^{k-1^T} S_d^{k-1} + \beta_0^{N_a+1} r^k)^T\end{aligned} \tag{31}$$

Noting that the adaptive neural net algorithm in (24) is tracking error of the $(k+2)$th step, then, we can derive the kth step error by defining $k_2 = k - 2$ that

$$e_1^k = M_\tau^{k_2} \tilde{W}_\tau^{k_2^T} S_\tau^{k_2} + \tau_{dp}^{k_2} \tag{32}$$

where $\underline{m}_\tau \le \|M_\tau^{k_2}\| \le \bar{m}_\tau$ according Lemma 1, and

$$e_1^k = e^k + M_\tau^{k_2} k_{pd} e^{k_2} - M_\tau^{k_2} k_d e^{k_2-1}$$

Based on the error $e_d^k = Q^k - Q_d^k$ and the tracking error e_1^k, the actor RBFNN update rule for (20) is given by

$$\begin{aligned}\hat{W}_\tau^{k+1} &= \hat{W}_\tau^{k_2} + \Delta\hat{W}_\tau^{k_2} \\ &= \hat{W}_\tau^{k_2} - \Gamma_\tau S_\tau^{k_2}[e_1^k - \hat{Q}_\tau^k]^T\end{aligned} \tag{33}$$

where $\Gamma_\tau = \gamma_\tau \mathbf{I}_{[N_\tau]} \in \Re^{N_\tau \times N_\tau}$ is a diagonal action system learning rate matrix with $\gamma_\tau > 0$, and $\hat{Q}_\tau^k = \beta_d \beta_0^{N_a+1} \hat{Q}^k$.

4.4. Stability Analysis

It has been shown that there exists an ideal control input $\tau_n^*(\bar{z}^k)$, which can guarantee the predictor error $e^{k+2} = 0$, if the unknown disturbance $\tau_{dp}^k = 0$. Because all assumptions are only valid in compact set $\Omega_{\bar{z}}$, 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 τ^{k-1} are in Ω_τ, all current output p^k and all past outputs p^{k-1} are in Ω_p, all future outputs p^{k+1} are also in Ω_p, all past RBFNN weight errors $\Delta\hat{W}_d^{k-1}, \Delta\hat{W}_\tau^{k_2}$ are in Ω_{w_d} and Ω_{w_τ}, 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

$$\begin{aligned} V^k &= V_1^k + V_2^k + V_3^k \\ &= tr[\tilde{W}_d^{k^T}\Gamma_d^{-1}\tilde{W}_d^k] \\ &+ \frac{1}{\rho_d}(\tilde{W}_d^{k-1^T}S_d^{k-1})^T(\tilde{W}_d^{k-1^T}S_d^{k-1}) \\ &+ \frac{1}{\rho_\tau}\sum_{j=0}^{n} tr[\tilde{W}_\tau^{k-2+i^T}\Gamma_\tau^{-1}\tilde{W}_\tau^{k-2+i}] \end{aligned} \tag{34}$$

where Γ_d and Γ_τ are diagonal learning rate matrices for critic RBFNN and actor RBFNN in (30) and (33), respectively. $\tilde{W}_d^k = \hat{W}_d^k - W_d^{*k}$, $\tilde{W}_\tau^k = \hat{W}_\tau^k - W_\tau^{*k}$, ρ_d and ρ_τ 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 kth 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

$$\Delta V^k = \Delta V_1^k + \Delta V_2^k + \Delta V_3^k \tag{35}$$

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

$$\begin{aligned} \Delta V_1^k &= tr[\tilde{W}_d^{k+1^T}\Gamma_d^{-1}\tilde{W}_d^{k+1} - \tilde{W}_d^{k^T}\Gamma_d^{-1}\tilde{W}_d^k] \\ &= tr[-2\tilde{W}_d^{k^T}S_d^kS_d^{k^T}\hat{W}_d^k \\ &+ 2\tilde{W}_d^{k^T}S_d^k\beta_0S_d^{k-1^T}\hat{W}_d^{k-1} \\ &- 2\tilde{W}_d^{k^T}S_d^k\beta_0^{N_l+1}r^k \\ &- 2\hat{W}_d^{k^T}S_d^kS_d^{k^T}\Gamma_dS_d^k\beta_0S_d^{k-1^T}\hat{W}_d^{k-1} \\ &+ 2\hat{W}_d^{k^T}S_d^kS_d^{k^T}\Gamma_dS_d^k\beta_0^{N_m+1}r^{k^T} \\ &- 2\beta_0\hat{W}_d^{k-1^T}S_d^{k-1}S_d^{k^T}\Gamma_dS_d^k\beta_0^{N_m+1}r^{k^T} \\ &+ \hat{W}_d^{k^T}S_d^kS_d^{k^T}\Gamma_dS_d^kS_d^{k^T}\hat{W}_d^k \\ &+ \beta_0\hat{W}_d^{k-1^T}S_d^{k-1}S_d^{k^T}\Gamma_dS_d^k\beta_0S_d^{k-1^T}\hat{W}_d^{k-1} \\ &+ \beta_0^{N_m+1}rkS_d^{k^T}\Gamma_dS_d^k\beta_0^{N_m+1}r^{k^T}] \end{aligned} \tag{36}$$

where $\hat{W}_d^k = \tilde{W}_d^k + W_d^*$ and $\hat{W}_d^{k-1} = \tilde{W}_d^{k-1} + W_d^*$.
For convenience to analyse, we define

$$\begin{aligned} &\mathcal{A}^k = \tilde{W}_d^{k^T}S_d^k, \;\; \mathcal{B}^k = W_d^{*^T}S_d^k - \beta_0W_d^{*^T}S_d^{k-1} \\ &\mathcal{C}^k = \beta_0\tilde{W}_d^{k-1^T}S_d^{k-1}, \;\; \mathcal{D}^k = M_\tau^{k_2}\tilde{W}_\tau^{k_2^T}S_\tau^{k_2} \\ &\mathcal{E}^k = \beta_d\beta_0^{N_a+1}, \;\; \mathcal{F}^k = \mathcal{E}^ke_2^{k+2}, \;\; S_d^{k^T}\Gamma_dS_d^k = c \end{aligned} \tag{37}$$

then, the equation (36) is rewritten as

$$\begin{aligned} \Delta V_1^k &= -2\mathcal{A}^k(\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - \mathcal{C}^k)^T \\ &+ c(\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - \mathcal{C}^k)^T \\ &\times (\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - \mathcal{C}^k) \\ &\leq -(1-c)(\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - \mathcal{C}^k)^T \\ &\times (\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - \mathcal{C}^k) - \mathcal{A}^{k^T}\mathcal{A}^k \\ &+ 3\mathcal{B}^{k^T}\mathcal{B}^k + 3\mathcal{C}^{k^T}\mathcal{C}^k + 3\mathcal{F}^{k^T}\mathcal{F}^k \end{aligned} \tag{38}$$

According the equation (25), we have

$$\mathcal{F}^k = \mathcal{E}^ke_2^{k+2} = \mathcal{E}^k(e^{k+2} + g_1e^k - g_2e^{k-1})$$

Noting that

$$\mathcal{F}^k = \mathcal{E}^ke_2^k = \mathcal{E}^k(e^k + g_1e^{k_2} - g_2e^{k_2-1})$$

Considering (32), we define a new vector $\mathcal{F}_e^k \in \Re^n$ as

$$\mathcal{F}_e^k = \mathcal{E}^ke_1^k = \mathcal{E}^k(e^k + M_\tau^{k_2}k_{pd}e^{k_2} - M_\tau^{k_2}k_de^{k_2-1}) \tag{39}$$

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

$$\begin{aligned} \mathcal{F}^{k^T}\mathcal{F}^k &= \mathcal{E}^{k^2}e_2^{k^T}e_2^k \leq \mathcal{E}^{k^2}(3g_1^2e^{k_2^T}e^{k_2} \\ &+ 3g_2^2e^{k_2-1^T}e^{k_2-1} + 3e^{k^T}e^k) \\ \mathcal{F}_e^{k^T}\mathcal{F}_e^k &= \mathcal{E}^{k^2}e_1^{k^T}e_1^k \\ &\leq \mathcal{E}^{k^2}(3k_{pd}^2e^{k_2^T}M_e^{k_2}e^{k_2} \\ &+ 3k_d^2e^{k_2-1^T}M_e^{k_2}e^{k_2-1} + 3e^{k^T}e^k) \end{aligned} \tag{40}$$

where $M_e^{k_2} = M_\tau^{k_2^T}M_\tau^{k_2}$.
We see $M_e^{k_2}$ is a symmetric positive definite matrix, and it is bounded with $\underline{m}_\tau^2 \leq \|M_e^{k_2}\| \leq \bar{m}_\tau^2$.

Theorem 1. *According the properties of symmetric positive definite matrix, the eigenvalues λ_i^e of $M_e^{k_2}$ are positive values, let us define $\lambda_{max}^e = max(\lambda_i^e)$ and $\lambda_{min}^e = min(\lambda_i^e)$, $i = 1, 2, \cdots, n$. According matrix norm property, we have $n\lambda_{min}^{e^2} \leq \|M_e^{k_2}\|^2 = \sum_{i=1}^n \lambda_i^1(M_e^{k_2^T}M_e^{k_2}) \leq n\lambda_{max}^{e^2}$*
Define

$$G_1^{k_2} = k_{pd}^2M_e^{k_2} - g_1^2\mathbf{I}_{[n]}, \;\; G_2^{k_2} = k_d^2M_e^{k_2} - g_2^2\mathbf{I}_{[n]}$$

It is noted that $\mathcal{F}^{k^T}\mathcal{F}^k \le \mathcal{F}_e^{k^T}\mathcal{F}_e^k$ can be satisfied, when the error coefficients g_1 and g_2 are given

$$g_1^2 \le \frac{k_{pd}^2 \underline{m}_\tau}{\sqrt{n}},\ g_2^2 \le \frac{k_d^2 \underline{m}_\tau}{\sqrt{n}}$$

Substituting (32) to $\mathcal{F}_e^{k^T}\mathcal{F}_e^k = \mathcal{E}^{k^2} e_1^{k^T} e_1^k$ in (40), we get

$$\begin{aligned}\mathcal{F}^{k^T}\mathcal{F}^k &\le \mathcal{F}_e^{k^T}\mathcal{F}_e^k = \mathcal{E}^{k^2} e_1^{k^T} e_1^k = \mathcal{E}^{k^2}\mathcal{D}^{k^T}\mathcal{D}^k \\ &+ 2\mathcal{E}^{k^2}\mathcal{D}^{k^T}\tau_{dp}^{k_2} + \mathcal{E}^{k^2}\tau_{dp}^{k_2^T}\tau_{dp}^{k_2} \\ &\le 2\mathcal{E}^{k^2}\mathcal{D}^{k^T}\mathcal{D}^k + 2\mathcal{E}^{k^2}\tau_{dp}^{*^2}\end{aligned} \tag{41}$$

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

$$\begin{aligned}\Delta V_1^k &\le -(1-c)(\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - C^k)^T \\ &\times (\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - C^k) - \mathcal{A}^{k^T}\mathcal{A}^k \\ &+ 3\mathcal{B}^{k^T}\mathcal{B}^k + 3C^{k^T}C^k \\ &+ 6\mathcal{E}^{k^2}\mathcal{D}^{k^T}\mathcal{D}^k + 6\mathcal{E}^{k^2}\tau_{dp}^{*^2}\end{aligned} \tag{42}$$

Taking the second term difference ΔV_2^k of (35), we get

$$\begin{aligned}\Delta V_2^k &= \frac{1}{\rho_d}[(\tilde{W}_d^{k^T} S_d^k)^T \times (\tilde{W}_d^{k^T} S_d^k) \\ &- (\tilde{W}_d^{k-1^T} S_d^{k-1})^T \times (\tilde{W}_d^{k-1^T} S_d^{k-1})] \\ &= \frac{1}{\rho_d}(\mathcal{A}^{k^T}\mathcal{A}^k - \frac{1}{\beta_0^2}C^{k^T}C^k)\end{aligned} \tag{43}$$

The third difference ΔV_3^k of (35) along (28), (30) and (33) is given by

$$\begin{aligned}\Delta V_3^k &= -\frac{2}{\rho_\tau}(e_1^k - \hat{Q}_\tau^k)^T \tilde{W}_\tau^{k_2^T} S_\tau^{k_2} \\ &+ \frac{b}{\rho_\tau}(e_1^k - \hat{Q}_\tau^k)^T(e_1^k - \hat{Q}_\tau^k)\end{aligned} \tag{44}$$

where $b = S_\tau^{k_2^T}\Gamma_\tau S_\tau^{k_2}$. It is noted that

$$\hat{Q}_\tau^k = \beta_d\beta_0^{N_a+1}\hat{Q}^k = \mathcal{E}^k \tilde{W}_d^{k^T} S_d^k + \mathcal{E}^k W_d^{*^T} S_d^k$$

Defining $\mathcal{U}^k = \tau_{dp}^{k_2} - \mathcal{E}^k W_d^{*^T} S_d^k$ and substituting (32) into (44), we have

$$\begin{aligned}\Delta V_3^k &\le -\frac{1}{\rho_\tau}(\mathcal{D}^k + \mathcal{U}^k - \mathcal{E}^k\mathcal{A}^k)^T \\ &\times (M_\tau^{k_2^{-1}} - b\mathbf{I}_{[n]}) \\ &\times (\mathcal{D}^k + \mathcal{U}^k - \mathcal{E}^k\mathcal{A}^k) \\ &- \mathcal{D}^{k^T}\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}}\mathcal{D}^k \\ &+ 2\mathcal{U}k^T\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}}\mathcal{U}^k \\ &+ 2\mathcal{E}^{k^2}\mathcal{A}^{k^T}\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}}\mathcal{A}^k\end{aligned} \tag{45}$$

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

$$\begin{aligned}\Delta V^k &\le -\mathcal{A}^{k^T}(\mathbf{I}_{[n]} - \frac{1}{\rho_d}\mathbf{I}_{[n]} - 2\mathcal{E}^{k^2}\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}})\mathcal{A}^k \\ &- (\frac{1}{\rho_d\beta_0^2} - 3)C^{k^T}C^k - \mathcal{D}^{k^T}(\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}} \\ &- 6\mathcal{E}^{k^2}\mathbf{I}_{[n]})\mathcal{D}^k \\ &- (1-c)(\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - C^k)^T \\ &\times (\mathcal{A}^k + \mathcal{B}^k + \mathcal{F}^k - C^k) \\ &- \frac{1}{\rho_\tau}(\mathcal{D}^k + \mathcal{U}^k - \mathcal{E}^k\mathcal{A}^k)^T(M_\tau^{k_2^{-1}} - b\mathbf{I}_{[n]}) \\ &\times (\mathcal{D}^k + \mathcal{U}^k - \mathcal{E}^k\mathcal{A}^k) + \|\mathcal{J}\|^2\end{aligned} \tag{46}$$

where

$$\|\mathcal{J}^k\|^2 = 3\mathcal{B}^{k^T}\mathcal{B}^k + 6\mathcal{E}^{k^2}\tau_{dp}^{*^2} + 2\mathcal{U}^{k^T}\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}}\mathcal{U}^k$$

We see that

$$\begin{aligned}c &= S_d^{k^T}\Gamma_d S_d^k = \gamma_d S_d^{k^T} S_d^k < \gamma_d N_d \\ b &= S_\tau^{k^T}\Gamma_\tau S_\tau^k = \gamma_\tau S_\tau^{k^T} S_\tau^k < \gamma_\tau N_\tau\end{aligned}$$

And thus,

$$\begin{aligned}\|\mathcal{J}^k\|^2 \le \|\mathcal{J}_\tau^k\|^2 &= (6 + 6\beta_0^2 + \frac{4\mathcal{E}^{k^2}\beta_d^2}{\rho_\tau \underline{m}_\tau^*})\|W_d^*\|^2 N_d \\ &+ \frac{4\tau_{dp}^{*^2}}{\rho_\tau \underline{m}_\tau^*} + 6\mathcal{E}^{k^2}\tau_{dp}^{*^2}\end{aligned}$$

Note Theorem 1 and Lemma 2, we know $M_\tau^{k_2^{-1}}$ is also a symmetric positive definite matrix, and is bounded with $\underline{m}_\tau^* \le \|M_\tau^{k_2^{-1}}\| \le \bar{m}_\tau^*$. Then, we define the eigenvalues of $M_\tau^{k_2^{-1}}$ are $\lambda_i^{k_2}$, $i = 1, 2, \cdots, n$. It is obvious that $\lambda_i^{k_2} > 0$. We define $\lambda_{max}^{k_2} = max(\lambda_i^{k_2})$ and $\lambda_{min}^{k_2} = min(\lambda_i^{k_2})$, $i = 1, 2, \cdots, n$, then, $n\lambda_{min}^{k_2} \le \|M_\tau^{k_2^{-1}}\|^2 = \sum_{i=1}^n \lambda_i(M_\tau^{k_2^{-1^T}} M_\tau^{k_2^{-1}}) \le n\lambda_{max}^{k_2}$.
Define

$$\begin{aligned}\mathcal{H}^k &= (1 - \frac{1}{\rho_d})\mathbf{I}_n - 2\mathcal{E}^{k^2}\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}} \\ \mathcal{I}^k &= \frac{1}{\rho_\tau}M_\tau^{k_2^{-1}} - 6\mathcal{E}^{k^2}\mathbf{I}_{[n]} \\ O^k &= M_\tau^{k_2^{-1}} - b\mathbf{I}_{[n]}\end{aligned}$$

The matrices $\mathcal{H}^k$, $\mathcal{I}^k$ and O^k are symmetric positive definite, and they need satisfied the following conditions:

$$(1-\frac{1}{\rho_d})-2\mathcal{E}^{k^2}\frac{1}{\rho_\tau}\frac{\bar{m}_\tau^*}{\sqrt{n}}>0$$
$$\frac{1}{\rho_\tau}\frac{m_\tau^*}{\sqrt{n}}-6\mathcal{E}^{k^2}>0,\quad 1-b\frac{\bar{m}_\tau^*}{\sqrt{n}}>0$$

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:*

$$\begin{cases} 0<\rho_d\leq\frac{1}{3\beta_0^2} \\ 0<\beta_0<\frac{\sqrt{3}}{3} \\ \frac{2\mathcal{E}^{k^2}\bar{m}_\tau^*}{(1-\frac{1}{\rho_d})\sqrt{n}}<\rho_\tau<\frac{m_\tau^*}{6\mathcal{E}^{k^2}\sqrt{n}} \\ 0<\gamma_d<\frac{1}{N_d} \\ 0<\gamma_\tau<\frac{\sqrt{n}}{N_\tau\bar{m}_\tau^*} \end{cases} \tag{47}$$

Assuming that the condition set above are satisfied, we have

$$\begin{aligned}\Delta V^k \leq & -\mathcal{A}^{k^T}((1-\frac{1}{\rho_d})\mathbf{I}_{[n]}-2\mathcal{E}^{k^2}\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}})\mathcal{A}^k \\ & -(\frac{1}{\rho_d\beta_0^2}-3)C^{k^T}C^k-\mathcal{D}^{k^T}(\frac{1}{\rho_\tau}M_\tau^{k_2^{-1}} \\ & -6\mathcal{E}^{k^2}\mathbf{I}_{[n]})\mathcal{D}^k+\|\mathcal{J}_\tau\|^2\end{aligned} \tag{48}$$

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

$$\begin{aligned}&\|\mathcal{J}_\tau^k\|^2<(\mathcal{P}_H\mathcal{A}^k)^T(\mathcal{P}_H\mathcal{A}^k)\\&\|\mathcal{A}^k\|^2>\|\mathcal{J}_\tau^k\|\|\mathcal{P}_H\|^{(-1)}\\&\qquad\text{or}\\&\|\mathcal{J}_\tau^k\|^2<(\mathcal{P}_I\mathcal{D}^k)^T(\mathcal{P}_I\mathcal{D}^k)\\&\|\mathcal{D}^k\|^2>\|\mathcal{J}_\tau^k\|\|\mathcal{P}_I\|^{(-1)}\end{aligned} \tag{49}$$

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

$$e^k=(\mathbf{I}_{[n]}+M_\tau^k k_{pd}z^{k-2}-M_\tau^k k_d z^{k-3})^{-1}e_1^k \tag{50}$$

Noting (49) and (50), we know there exists a finite running step K_τ, which makes $\|\mathcal{A}^k\|^2\leq\|\mathcal{J}_\tau^k\|\|\mathcal{P}_H\|^{(-1)}$ or $\|\mathcal{D}^k\|^2\leq\|\mathcal{J}_\tau^k\|\|\mathcal{P}_I\|^{(-1)}$, and makes $\|e^k\|\leq\|e_1^k\|$ under $(\mathbf{I}_{[n]}+M_\tau^k k_{pd}z^{k-2}-M_\tau^k k_d z^{k-3})^{-1}$ being Hurwitz-stable for all $k>K_\tau$.

From the definition of $\mathcal{A}^k$ and $\mathcal{D}^k$, the boundedness of $\tilde{W}_d^{k^T}S_d^k$ and $\tilde{W}_\tau^{k^T}S_\tau^k$ can be deduced. $\tilde{W}_d^{*^T}S_d^k$ and $\tilde{W}_\tau^{*^T}S_\tau^k$ are bounded, then, $\hat{W}_d^{k^T}S_d^k$ and $\hat{W}_\tau^{k^T}S_\tau^k$ are also bounded. We see that the boundedness of $\hat{W}_d^{k^T}S_d^k$ and $\hat{W}_\tau^{k^T}S_\tau^k$ further implies that $\hat{W}_d^k$ and $\hat{W}_\tau^k$ are bounded. with the boundedness of M_τ^k and τ_{dp}^*, we know the tracking error e_1^k is bounded as

$$\begin{aligned}\|e_1^k\|^2 &= e_1^{k^T}e_1^k\leq 2\mathcal{D}^{k^T}\mathcal{D}^k+2\tau_{dp}^{k_2^T}\tau_{dp}^{k_2}\\&<2\|\mathcal{J}_\tau^k\|\|\mathcal{P}_I\|^{(-1)}+2\tau_{dp}^{*^2}\end{aligned} \tag{51}$$

or, we can get

$$\|e^k\|\leq\|e_1^k\|<\sqrt{2\|\mathcal{J}_\tau^k\|\|\mathcal{P}_I\|^{(-1)}+2\tau_{dp}^{*^2}} \tag{52}$$

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 $m_1=m_2=1.0kg$, the length are $l_1=l_2=0.2m$, the inertia are $I_1=I_2=0.003kgm^2$, the distance are $l_{c1}=l_{c2}=0.1m$.
The dynamics of the robot manipulator with $G(q)=\mathbf{0}_{[2]}$ is given as follows:

$$\begin{aligned}M(q)&=\begin{bmatrix}M_{11}&M_{12}\\M_{21}&M_{22}\end{bmatrix}\\C(q,\dot{q})&=\begin{bmatrix}C_{11}&C_{12}\\C_{21}&C_{22}\end{bmatrix}\end{aligned} \tag{53}$$

where

$$\begin{aligned}&M_{11}=m_1l_{c1}^2+m_2(l_1^2+l_{c2}^2+2l_1l_{c2}cos(q_2))+I_1+I_2\\&M_{12}=M_{21}=m_2(l_{c2}^2+l_1l_{c2}cos(q_2)+I_2\\&M_{22}=m_2l_{c2}^2+I_2\\&C_{11}=-m_2l_1l_{c2}sin(q_2)\dot{q}_2\\&C_{12}=-m_2l_1l_{c2}sin(q_2)(\dot{q}_1+\dot{q}_2)\\&C_{21}=m_2l_1l_{c2}sin(q_2)\dot{q}_1\quad C_{22}=0\end{aligned}$$

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

$$\tau_{ds}=[0.05cos(0.01t)cos(q_1),0.05cos(0.01t)cos(q_2)]^T$$
$$\tau_{db}=[40cos(0.01t)cos(q_1),40cos(0.01t)cos(q_2)]^T$$

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 $\hat{W}_d^{k^T} S_d^k$ 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 $\hat{W}_\tau^{k^T} S_\tau^k$ approximating the system tracking error using $N_\tau = 4096$ with all the centres of Gaussian function evenly in $[-1; 1]$, and the widths=1. The design parameters are chosen as $\gamma_d = 0.0005$, $\gamma_\tau = 0.0001$, $\beta_d = 0.8$, $N_a = 3$, $\beta_0 = 0.5$, $k_p = 0.5$, $k_d = 120$, The initial weights $\hat{W}_d(0) = \mathbf{0}_{[2\times N_d]}$, $\hat{W}_\tau(0) = \mathbf{0}_{[2\times N_\tau]}$, and we choose the controller sampling interval $T = 0.01s$..

To show the effectiveness, we have done the relevant comparative analysis for trajectory tracking accuracy and capability for the robot manipulator with τ_{ds} and τ_{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 $\tau^k = sat(K_1 e^k + K_2 \hat{f}(k))$ is applied , where K_1 and K_2 are gain matrices, and $\hat{f}(k)$ is the estimation of all uncertain terms $f(k)$. The parameters $K_1 = [K_1 1 K_1 2]$ and K_2 obtained by using LMIs theory as $K_{11} = [-10 -5; -5 -22]$, $K_{12} = [-10 -5; -2 -10]$, and $K_2 = [0.0150.025; 0.0350.05]$, respectively.

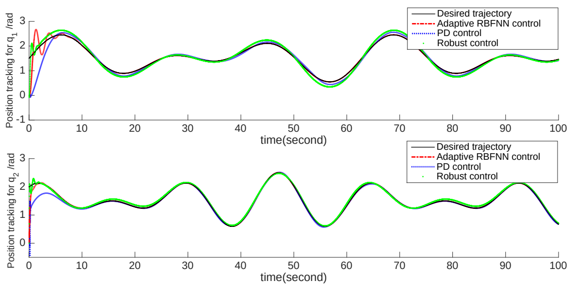

Figure 1: Position trajectory tracking curves of q_1 and q_2 for a small disturbance

Fig.1-2 show tracking curves of joint positions q_1 and q_2, Fig.3-4 show control input curves of control inputs τ_1 and τ_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

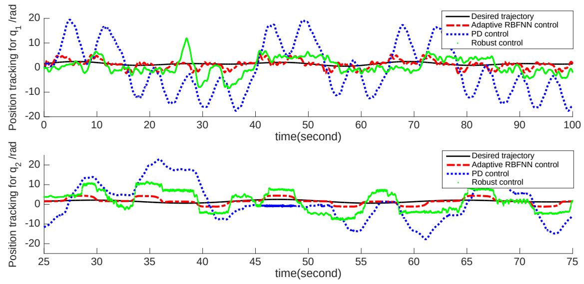

Figure 2: Position trajectory tracking curves of q_1 and q_2 for a large disturbance

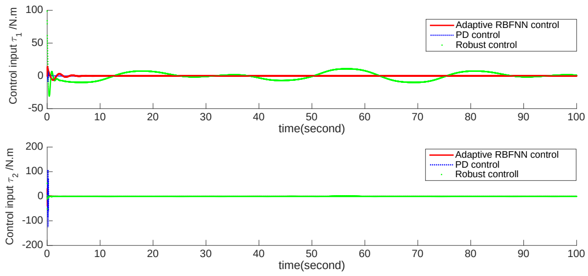

Figure 3: Control input curves of τ_1 and τ_2 for a small disturbance

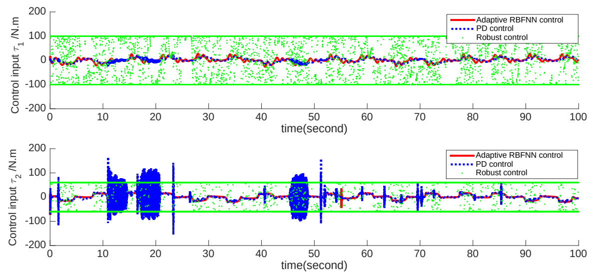

Figure 4: Control input curves of τ_1 and τ_2 for a large disturbance

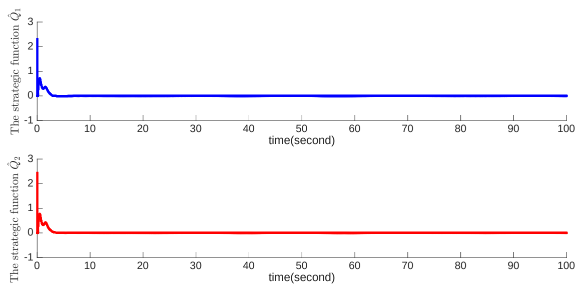

Figure 5: The strategic utility functions $\hat{Q}_1$ and $\hat{Q}_2$ for a small disturbance

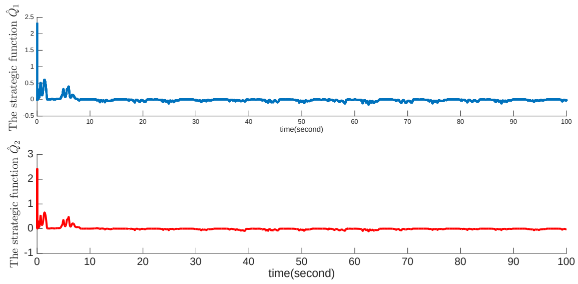

Figure 6: The strategic utility funcitons $\hat{Q}_1$ and $\hat{Q}_2$ for a large disturbance

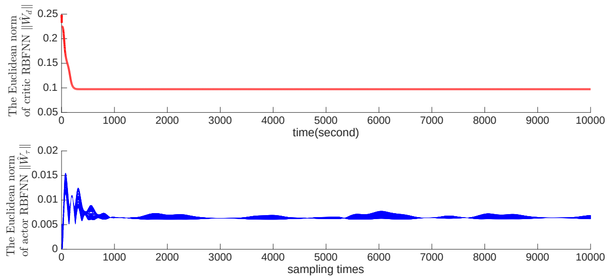

Figure 7: The Euclidean norm curves of critic RBFNN $||W_d||$ and actor RBFNN $||W_\tau||$ for a small disturbance

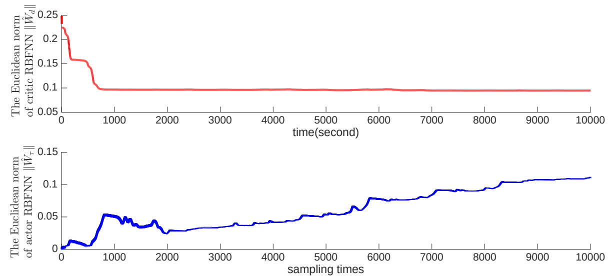

Figure 8: The Euclidean norm curves of critic RBFNN weight $||W_d||$ and actor RBFNN weight $||W_\tau||$ for a large disturbance

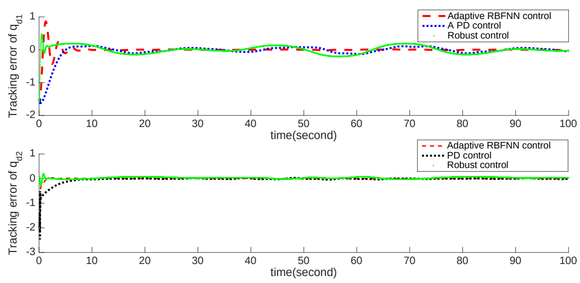

Figure 9: Tracking error curves of e_1 and e_2 for a small disturbance

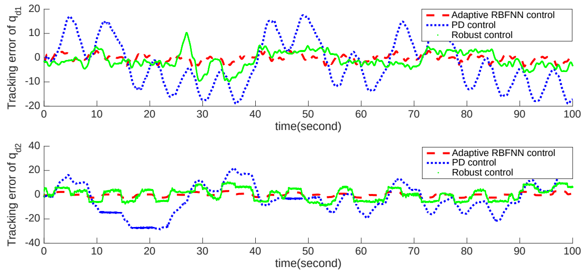

Figure 10: Tracking error curves of e_1 and e_2 for a large disturbance

RBFNN weight $||W_\tau||$, and Fig. 9-10 show tracking error curves of $e_1 = q_1 - q_{d1}$ and $e_2 = q_2 - q_{d2}$ with a small external disturbance τ_{ds} or a large external disturbance τ_{db}, respectively.

Analyzing all above simulation results, the proposed optimal adaptive RBFNN control, in comparison to the PD control and the robust control, is applied for robot trajectory tracking.

It is obvious that the first joint has an small initial error, and the position value q_1 deviates from the desired joint position trajectory q_{1d} for less than $5s$, but the controller is able to regulates quickly the joint position q_1 to track the desired trajectory q_{1d}, and make overall control optimization by using the proposed control system. The second joint has the excellent performance for a small external disturbance torque in Fig. 1. The method also is able to achieve satisfied control performance for existing a large external disturbance in Fig 2.

The control inputs τ_1 and τ_2 are well bounded in Fig. 3-4, and the critic utility function $\hat{Q}$ converge to a small of neighbourhood of zero in Fig. 5-6.

Most of the adaptive RBFNN weight norm $||\hat{W}_\tau||$ estimates remain in or converge to a small neighbourhood of zero for a small disturbance, and also can gradually converge to 0.1 for a large disturbance in Fig. 7-8.

Therefore, the effectiveness and optimal performance of the proposed control algorithm has been successfully demonstrated despite the presence of the external force torque and a wide range of uncertain disturbances.

6. Conclusion

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. Control laws are real-time adaptive 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.

[1] F. L. Lewis, D. M. Dawson, C. T. Abdallah, Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles, 2rd. Marcel Dekker, Inc., New York, USA, 2004.

[2] F. L. Lewis, S. Jagannathan, A. Yesildirak, Neural Network Control of Robot Manipulators and Non-linear Systems, Taylor Francis: London, UK:1 Gunpowder Square, EC4A 3DE, 1999.

[3] X. Ren, A. B. Rad, F. L. Lewis, Neural network-based compensation control of robot manipulators with unknown dynamics, American control conference (2007) 13–18.

[4] G. Palli, C. Melchiorri, A. D. Luca, On the feedback linearization of robots with variable joint stiffness, IEEE International Conference, Robotics and Automation (2008) 19–23.

[5] N. S. Bedrossian, M. W. Spong, Feedback linearization of robot manipulators and riemannian curvature, Journal of Robotic Systems 12 (8) (1995) 541–552.

[6] C. Pukdeboon, Lyapunov Optimizing Sliding Mode Control for Robot Manipulators, Applied Mathematical Sciences 7 (63) (2013) 3123–3139.

[7] J. M. Yang, J. H. Kim, Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots, Robotics and Automation, IEEE Trans. 15 (3) (1999) 3123–3139.

[8] H. B. Guo, Y. G. Liu, G. R. Liu, H. R. Li, Cascade control of a hydraulically driven 6-dof parallel robot manipulator based on a sliding mode, Control Engineering Practice 16 (9) (2008) 1055–1068.

[9] Y. J. Pi, X. Y. Wang, Trajectory tracking control of a 6-dof hydraulic parallel robot manipulator with uncertain load disturbances, Control Engineering 19 (2) (2011) 185–193.

[10] M. N. Mahyuddin, G. Herrmann, S. G. Khan, A novel adaptive control algorithm in application to a humanoid robot arm, Advances in Autonomous Robotics 7429 (2012) 25–36.

[11] T. F. Pazelli, M. H. Terra, A. A. Siqueira, Experimental investigation on adaptive robust controller designs applied to a free-floating space manipulator, Control Engineering Practice 19 (4) (2011) 395–408.

[12] N. Mendes, P. Neto, Indirect adaptive fuzzy control for industrial robots: A solution for contact applications, Expert Systems With Applications 42 (22) (2015) 8929–8935.

[13] J. S. Park, G. S. Han, H. S. Ahn, D. H. Ki, Adaptive Approaches on the Sliding Mode Control of Robot Manipulators, Control, Automation and Systems Engineering, Trans. 3 (1) (2001) 15–20.

[14] A. B. B. Sharkawy, S. A. Salman, An Adaptive Fuzzy Sliding Mode Control Scheme for Robotic Systems, Intelligent Control and Automation 2 (2011) 299–309.

[15] X. Li, C. Cheah, Adaptive Neural Network Control of Robot Based on a Unified Objective Bound, Control Systems Technology, IEEE Trans. 22 (3) (2014) 1032–1043.

[16] C. L. P. Chen, G. X. Wen, Y. J. Liu, F. Y. Wang, Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks, Neural Networks and Learning Systems, IEEE Trans. 25 (6) (2014) 1217–1226.

[17] J. K. Liu, Y. Lu, Adaptive rbf neural network control of robot with actuator nonlinearities, Journal of Control Theory and Applications 8 (2) (2010) 249–256.

[18] S. Jagannathaan, F. L. Lewis, Multilayer discrete-time neural-net controller with guaranteed performance, Neural Networks, IEEE Trans. 7 (1) (1996) 107–130.

[19] M. M. Fateh, S. M. Ahmadi, S. Khorashadizadeh, Adaptive RBF network control for robot manipulators, Journal of AI and Data Mining 2 (2) (2014) 159–166.

[20] W. H. Zhang, X. P. Ye, L. H. Jiang, Y. M. Fang, Robust Control for Robotic Manipulators Base on Adaptive Neural Network, The Open Mechanical Engineering Journal 8 (2014) 497–502.

[21] S. G. Tzafestas, Dynamic modelling and adaptive control of industrial robots: The state-of-art, Systems Analysis Modelling Simulation 1 (1989) 243–266.

[22] L. F. Zhai, T. Y. Chai, S. Ge, T. H. Lee, Stable Adaptive Neural Network Control of MIMO Nonafne Nonlinear Discrete-Time Systems, Decision and Control, 47th IEEE Conference (2008) 9–11.

[23] Y. N. Li, C. G. Yang, S. S. Ge, T. H. Lee, Adaptive output feedback nn control of a class of discrete-time mimo nonlinear systems with unknown control directions, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans. 41 (2) (2011) 507–517.

[24] F. C. Sun, L. Li, H. X. Li, H. P. Liu, Neuro-fuzzy dynamic-inversion-based adaptive control for robotic manipulatorsdiscrete time case, Industrial Electronics, IEEE Trans. 54 (3) (2007) 1342–1351.

[25] M. M. Fateh, S. Azargoshasb, Discrete time robust control of robot manipulators in the task space using adaptive fuzzy estimator, Journal of AI and Data Mining 3 (1) (2015) 113–120.

[26] L. F. Zhai, C. G. Yang, S. S. Ge, T. Y. Chai, T. H. Lee, Direct adaptive nn control of mimo nonlinear discrete-time systems using discrete nussbaum gain, The International Federation of Automatic Control, 17th World Congress (2008) 6508–6512.

[27] B. Xu, C. Y. Yang, Z. K. Shi, Reinforcement Learning Output Feedback NN Control Using Deterministic Learning Technique, Neural Networks and Learning Systems, IEEE Trans. 25 (3) (2014) 635–640.

[28] S. L. Dai, C. Wang, F. Luo, Identification and Learning Control of Ocean Surface Ship Using Neural Networks, Industrial Informatics, IEEE Trans. 8 (4) (2012) 801–809.

[29] F. C. Sun, H. X. Li, L. Li, Robot discrete adaptive control based on dynamic-inversion using dynamical neural networks, Automatica 38 (11) (2002) 1977–1983.

[30] A. Zagorianos, S. G. Tzafestas, G. S. Stavrakakis, On line discretetime control of industrial robots, Robotics and Autonomous Systems 14 (4) (1995) 289–299.

[31] C. G. Yang, S. S. Ge, C. Xiang, T. Y. Chai, T. H. Lee, Output feedback nn control for two classes of discrete-time systems with unknown control directions in a unified approach, Neural Networks, IEEE Trans. 19 (11) (2008) 1873–1886.