Conference contribution:
Yang, R., Yang, C., Chen, M. & Na, J. (2016). *RBFNN based adaptive control of uncertain robot manipulators in discrete time.* (pp. 1-6). Belfast, Northern Ireland, UK:
http://dx.doi.org/10.1109/CONTROL.2016.7737617

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/
RBFNN Based Adaptive Control of Uncertain Robot Manipulators in Discrete Time

Runxian Yang 1,2,3, Chenguang Yang2∗, Mou Chen1 and Jing Na4
1 College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China
2 Zienkiewicz Centre for Computational Engineering, Swansea University, Swansea, UK
3 College of Electric and IT, Yangzhou Polytechnic Institute, Yangzhou, China
4 Department of Mechanical Engineering, University of Bristol, Bristol, UK

Abstract—The trajectory tracking control problem for a class of n-degree-of-freedom (n-DOF) rigid robot manipulators is studied in this paper. A novel adaptive radial basis function neural network (RBFNN) control is proposed in discrete time for multiple-input multiple-output (MIMO) robot manipulators with nonlinearity and time-varying uncertainty. The high order discrete-time robot model is transformed to facilitate digital implementation of controller, and the output-feedback form is derived to avoid potential noncausal problem in discrete time. Furthermore, the desired controller based on RBFNN is designed to compensate for effect of uncertainties, and the RBFNN is trained using tracking error, such that the stability of closed-loop robot system has been well guaranteed, the high-quality control performance has been well satisfied. The RBFNN weight adaptive law is designed and the semi-global uniformly ultimate boundedness (SGUUB) is achieved by Lyapunov based on control synthesis. Comparative simulation studies show the proposed control scheme results in supreme performance than conventional control methods.

I. INTRODUCTION

With advances of technologies, robot applications in industry and our daily life become increasingly popular, the relevant research works have been an attractive topic. However, most robot manipulators are usually subject to unmodelled dynamics and various uncertainties in practice [2]–[4], an ideal control design for a class of robot manipulators is challenging.

Various approaches for trajectory tracking control of robot manipulators have been proposed. Feedback linearization methods [5], [6], sliding mode and other robust control methods [1], [7]–[9] have all been extensively investigated for robot control, and global tracking error convergence are able to be guaranteed. Furthermore, the advanced intelligent methods and relevant research results have been well applied to robot control, e.g., adaptive control [10], [11], adaptive-fuzzy control [12], adaptive-sliding control [13], and complex adaptive control based on fuzzy and sliding-mode theories for robot manipulators [14], function approximators have also been utilized. In order to compensate for uncertainties of robot manipulators, adaptive neural network (ANN) techniques have been popular in recent years [15]–[17]. ANNs have universal approximation capability for nonlinear functions. An adaptive RBFNN algorithm guaranteeing closed-loop stability has been proposed for robot manipulator systems in [18]. A novel RBFNN estimator has been designed to compensate for uncertainties in [19], [20]. These approaches are able to guarantee UUB of closed-loop system of robot manipulators. But the digital implementation of robot controllers and network communication of high-speed computers are becoming increasingly popular and powerful. Thus, the increasing research works for robot manipulators have now been carried out in discrete time.

Discrete-time robot manipulator models and discrete-time control methods are used in [18], [21], the discrete-time controllers applied to on-line robot control provides convenience for implementation. In [22], a ANN controller has been proposed based on combining one-step-ahead control with ANN control for a class of MIMO discrete-time systems with nonaffine nonlinearity. In [28], a class of MIMO nonlinear systems with block triangular structure can be decomposed in discrete time, by applying pure-feedback method, an ANN control has been presented based on all subsystems with couplings and unknown directions. These discrete-time approaches perform well to guarantee robust stability of nonlinear robot systems. However, these research works only guarantee stability of closed-loop robot manipulator systems, while realizing trajectory tracking control is seldom in discrete time. Thus, a novel control scheme proposed for a class of robot manipulators with uncertainty in discrete time is the main research objectives of this paper.

Aiming to address satisfied trajectory tracking performance based on stable closed-loop robot system, we develop a discrete-time novel RBFNN based adaptive control for uncertain robot manipulators.

The following notations are employed in this paper.

- $\| \cdot \|$ represents the Euclidean norm of vectors and induced norm of matrices.
- $b := a$ denotes $b$ is defined as $a$.
- $\| \cdot \|^T$ represents the transpose of a vector or a matrix.
- $\| \cdot \|^{-1}$ represents the inverse of a n-order reversible matrix.
• \( \theta_p \) denotes the dimension of zero vector is \( p \)-dimension.
• \( I_m \) stands for \( m \)-dimension unit matrix.
• \( \hat{W}^p \) represents the idea neural net weight matrix.
• \( \hat{W}^k \) represents the estimate value matrix of neural net idea weight \( \hat{W}^k \) at the \( k \)-th step.
• \( \hat{W} = \hat{W}^k - \hat{W}^\tau \) denotes the weight estimate error.

II. DISCRETIZING FOR ROBOT MODEL

The dynamic model of general \( n \)-DOF nonlinear rigid robot manipulators can be described using ordinary differential equation as

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

where \( q \in \mathbb{R}^n \) is the joint position, and \( \dot{q} \in \mathbb{R}^n \) is the joint velocity, \( \ddot{q} \in \mathbb{R}^n \) is the joint acceleration, \( M(q) \in \mathbb{R}^{n \times n} \) is the symmetric and 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 control input torque vector, \( \tau_d \in \mathbb{R}^n \) is the external force torque vector.

According to [2], the following properties hold for rigid robot manipulators in (1):

**Property 1**: \( M(q) \) is uniformly bounded, and satisfies the following inequality

\[
m \leq ||M(q)|| \leq \hat{m}
\]

**Property 2**: 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.

It is very important and meaningful to design robot controller in discrete time. For a class of \( n \)-DOF rigid nonlinear robot manipulators with uncertainty in (1), which can be discretized by using discretization theory with a small sampling time interval \( T \).

The sampled joint angle is \( q^k = q^k_0 \), the sampled joint angle velocity is \( \dot{q}^k = \dot{q}^k_0 \), the control torque is \( \tau^k = \tau^k_0 \), and the external disturbance torque is \( \tau_d^k = \tau_d^k_0 \) at the sampling time interval \( T_k = kT \), respectively.

Define \( p^k = q^k \in \mathbb{R}^n \) and \( v^k = \dot{q}^k \in \mathbb{R}^n \), then, the equivalent dynamics form in discrete time can be obtained [23], [26], [27] as

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

where \( M(\xi^k) \in \mathbb{R}^{n \times n} \) is also the inertia matrix with \( \xi^k \cong p^k + T\ddot{v}^k \in \mathbb{R}^n \), \( f(p^k, v^k) = C(p^k, v^k)\dot{v}^k + G(p^k) \in \mathbb{R}^n \), \( C(p^k, v^k) \in \mathbb{R}^{n \times n} \) is Coriolis-Centrifugal torque matrix and \( G(p^k) \in \mathbb{R}^n \) is gravitational synthetic vector in discrete time, respectively.

According to Property 1, \( M(\xi^k) \) is also symmetric, positive definite and bounded, satisfying \( \hat{m} \leq ||M(\xi^k)|| \leq \tilde{m} \) with known constants \( \hat{m} > 0 \) and \( \tilde{m} > 0 \).

III. TRANSFERING TO FEEDBACK SYSTEM

To avoid possible noncausal problem in control design, we extend our previous research works [24], [29], [30] to a class of nonlinear time-varying MIMO robot manipulators with uncertainty in discrete time.

The discrete-time dynamics in (3) can be transferred into the output-feedback control system [31] as

\[
\begin{aligned}
\hat{p}^{k+1} &= p^k + T\hat{v}^k \\
\hat{v}^{k+1} &= [(1 + T)I_n] - TM^{-1}(\xi^k)M(p^k) \\
&- TM^{-1}(\xi^k)C(p^k, \dot{v}^k)\dot{v}^k - TM^{-1}(\xi^k)G(p^k) \\
&+ TM^{-1}(\xi^k)\tau^k + TM^{-1}(\xi^k)\tau_d^k
\end{aligned}
\]

where \( \tau_d^k \) is bounded as \( ||\tau_d^k|| \leq \tilde{\tau}_d \) with an known constant \( \tilde{\tau}_d \).

It is easily known that \( M^{-1}(\xi^k) \) is also bounded, satisfying \( \tilde{m}^* \leq ||M^{-1}(\xi^k)|| \leq \hat{m}^* \) with known constants \( \hat{m}^* > 0 \) and \( \tilde{m}^* > 0 \).

The control objective is to synthesize an adaptive RBFNN control input \( \tau^k \) for robot system (4), not only all signals of closed-loop robot system are bounded, but also the joint position signal \( p^k \) is able to well track the ideal trajectory signal of robot manipulators \( p^k_c \in \Omega_{pc} \), finally, the satisfied control performance is able to be obtained, where \( \Omega_{pc} \) is a compact set.

It is noted that \( v^{k+1} \) depends on control output \( \hat{v}^k \), while \( \hat{p}^{k+1} \) is associated with \( \hat{p}^k \) and \( \hat{v}^k \) at the \( (k + 1) \)-th step in (4).

We can rewrite the first equation of the system (4) as \( \hat{p}^{k+1} = p^k + T\hat{v}^{k+1} \), \( p^k - T\hat{v} = \theta_0[n] \), and \( v^k \) is designed as \( \hat{v}^k = 1/2(p^{k+1} - p^k) \).

To predict the \((k + 2)\)th step of robot manipulators, we have

\[
\begin{aligned}
\hat{p}^{k+2} &= p^{k+1} + T\hat{v}^{k+1} \\
&= [(2 + T)I_n] - TM^{-1}(\xi^k)M(p^k) \\
&- TM^{-1}(\xi^k)C(p^k, \dot{v}^k)\dot{v}^k \\
&- [(1 + T)I_n] - TM^{-1}(\xi^k)G(p^k) \\
&+ TM^{-1}(\xi^k)\tau^k + TM^{-1}(\xi^k)\tau_d^k
\end{aligned}
\]

Furthemore, we need to move (5) back to the \((k + 1)\)-th step, the output-feedback method is applied to get the \( p^{k+1} \) as

\[
\begin{aligned}
p^{k+1} &= [(2 + T)I_n] - TM^{-1}(\xi^k)M(p^{k-1}) \\
&- TM^{-1}(\xi^k)C(p^{k-1}, \dot{v}^{k-1})\dot{v}^{k-1} \\
&- [(1 + T)I_n] - TM^{-1}(\xi^k)M(p^{k-1}) \\
&- TM^{-1}(\xi^k)C(p^{k-1}, \dot{v}^{k-1})\dot{v}^{k-1} \\
&+ T^2M^{-1}(\xi^k)\tau^k + T^2M^{-1}(\xi^k)\tau_d^{k-1}
\end{aligned}
\]

Substituting (6) to (5), we note that there is no more explicit future outputs and input signals. For convenience, let us define

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

Considering equation (6), we know that future state at the \((k + 1)\)-th step is able to be obtained by getting values of the
current \( k \)-th step and the past \((k-1)\)-th step. Then, the output \( p_{k+2} \) is obtained as
\[
p_{k+2} = (L_kL_{k-1} - R_k)p_k - L_kR_{k-1}p_{k-1}
- L_kM_{k-1}G_{k-1} - M_kG_k + L_kM_k^{-1}r_{k-1} + M_k^{-1}r_k
\]
and we further define
\[
\begin{align*}
L_k &= (L_kL_{k-1} - R_k)p_k - L_kR_{k-1}p_{k-1} + L_kM_k^{-1}r_{k-1} - M_k^{-1}r_k \\
L_G &= L_kM_k^{-1}G_{k-1} + M_kG_k \\
\tau_d &= L_kM_k^{-1}r_{k-1} + M_k^{-1}r_k
\end{align*}
\]
Thus, equation (7) can be rewritten as
\[
p_{k+2} = L_k^2 - L_kL_k^{-1} - R_k)p_k + L_kM_k^{-1}r_{k-1} + M_k^{-1}r_k
= \psi(p_{k-1}, p_k, \tau_k, r_{k-1}, r_k)
\]
It is easily known that the function \( \psi(\cdot, \cdot, \cdot, 0, 0) \) in (8) is continuous for all the arguments and continuously differentiable.

**Lemma 1:** \( M_k \) is symmetric positive definite matrix, and is bounded as \( m_r \leq M_k \leq M_r \) with \( m_r = T^2 \bar{m} \) and \( \bar{m} = T^{-2} \tilde{m} \).

According to Lemma 1, we know that \( L_d \) is bounded and \( \|L_d\| \leq (3 + 2T + T\bar{m}k_c)\bar{m} + \tau_d := \tau_d^* \).

**IV. ADAPTIVE RBFNN CONTROLLER DESIGN**

**A. RBFNN Approximation**

The RBFNN can approximate any nonlinear function \( F(z) \), which can be expressed as [25]:
\[
F(z) = W^T \Sigma(z), \quad W \in \mathbb{R}^{N_o \times N_o}, \quad \Sigma(z) \in \mathbb{R}^{N_o}
\]
where \( z = [z_1, z_2, \ldots, z_{N_o}] \in \mathbb{R}^{N_o} \) in \( \Omega_z \) is the input vector of RBFNN, \( N_o \) is neuron node number, \( N_o \) is output dimension of RBFNN, \( W \) is weight matrix, \( \Sigma(z) = [s_1(z), s_2(z), \ldots, s_{N_o}(z)]^T \) is hidden layer output function of RBFNN, and \( s_i(z) \) is the \( i \)-th neuron output function, the Gaussian RBFNN function is chosen as follows
\[
s_i(z) = e^{-|z_i - c_{ij}|^2/2\bar{\sigma}_i^2}
\]
where \( i = 1, 2, \ldots, N_o, \quad j = 1, 2, \ldots, N_o, \quad c_{ij} \) is the center of the \( j \)-th neuron node for the \( i \)-th input signal, \( b_i \) is the width of the \( j \)-th neuron.

A number of research results have shown that for any continuous smooth function \( \varphi(z) : \Omega_z \rightarrow R \) over a compact set \( \Omega_z \subset R^{N_o} \) [32], [33], we can apply RBFNN (9) to approximate \( \varphi(z) \). In particular, if \( N_o \) is chosen a sufficiently large value, such that the ideal bounded weight \( W^* \) exists, we have
\[
\varphi(z) = W^*T \Sigma(z) + \mu(z)
\]
(11)
where \( \mu(z) \) is the approximation error, which is bounded as \( |\mu(z)| < \mu^* \) with a given small constant \( \mu^* \).

RBFNN in (9) or in (11) has the following property, which will be used in the control design:
\[
S(z)^T S(z) < N_o
\]
(12)
Noting the ideal RBFNN weight \( W^* \) is unknown in practice, we often use \( \hat{W} \) as estimate weight of ideal weight \( W^* \) to approximate the unknown nonlinear function \( \varphi(z) \). By designing an appropriate learning rule, the estimate \( \hat{W} \) can be renewed. Then, equation (11) can be rewritten as
\[
\varphi(z) \approx \hat{W}^T S(z)
\]
(13)

**B. Desired Control**

The ideal system tracking output is \( p_d \). The dynamics of tracking error \( e_{k+2} \) can be obtained as
\[
e_{k+2} = p_{k+2} - p_d = L_k^2 - L_kG + M_k^{-1}r_k + L_d - \tau_d^+ = 0
\]
(14)
where \( p_{k+2} \) is defined in (8).

There exists a continuous ideal control input \( \tau_n \) [29], such that
\[
L_k - L_G + M_k^{-1} \tau_n - \tau_d^+ = 0
\]
(15)

**Lemma 2:** There are positive constants \( m_r = 1/\bar{m} \) and \( \bar{m} = 1/M_r \), and \( M_r^{-1} \) is bounded as \( \tilde{m}^* \leq \|M_r^{-1}\| \leq \tilde{m}^* \).

The predictor for two-step trajectory error \( e_{k+2} \) can be constrained as
\[
\|e_{k+2}\| \leq \tilde{m}^*
\]
(16)
It is noted that the desired control \( \tau_n \) is not obtained with the unknown \( M_k^{-1}, L_k, \) and \( L_G \). We apply the adaptive RBFNN to learn and to approximate the desired input \( \tau_n \), such that tracking error \( e_{k+2} = 0 \) after 2 steps can be achieved, if \( \tau_d = 0 \) and \( \tau_k = 0 \) in (14).

**C. RBFNN Based Control**

From Section IV-A, an ideal weight matrix \( W^*_k \) exists, we apply RBFNN Gaussian function \( S_r(z_k) \) to approximate the ideal control input \( \tau_n \) as follows
\[
\tau_n = W^T S_r(z_k) + \epsilon_r(z_k)
\]
(17)
where \( S_r(z_k) \in \mathbb{R}^{N_r} \) is the regression matrix, \( N_r \) is neuron node number, \( \|\epsilon_r(z_k)\| \leq \epsilon_r^* \) with \( \epsilon_r^* > 0 \) is the approximation error, the ideal weight matrix \( W^* \) is chosen as
\[
\tilde{m} = [p, p^{-1}, v, v^{-1}, r, r^{-1}, v, v^{-1}, p, p^{-1}, v, v^{-1}, r, r^{-1}] \in \Omega_z
\]
where \( \Omega_z \) is a sufficient large compact set corresponding to \( \Omega_{\mu} \). It is easy to verify the ideal control \( \tau_n(z_k) \) is bounded.

According to (14) and (15), we apply RBFNN to approximate the ideal control input \( \tau_n(z_k) \), and introduce PD method to improve control performance, the system control input is designed as:
\[
\tau_k = -k_p \dot{z}_k + k_d z_k + \tau_n(z_k)
\]
(18)
where \( k_p = k_p + k_d > 0 \).
According to equation (15), we have \( p_{k+2}d = L_k - L_k^G + M_k^T \tau_{kn} \). Then, equation (14) is rewritten as follows
\[
e^{k+2} = L_k^G - e^{k+2} = M_k^T (\tau_k - \tau_{kn}) + L_d^k
\]
(19)
For convenience, we define:
\[
S_k^T = S_r(\omega_k), \quad e_k^* = \epsilon_r(\omega_k)
\]
From Lemma 1, it is obvious that \( M_k^T \) is bounded with \( m_r \) and \( m_r \). Noting \( \hat{W}_r^k = W_r^k - W_r^{k-1} \), we substitute (17) and (18) into (19), then,
\[
e^{k+2} = M_k^T (-k_p e_k^* + k_d e^{k-1}) + M_k^T \hat{W}_r^{k-1} S_k^T + \tau_k^k
\]
where \( \tau_{dp}^k = -M_k^T e_k^* + L_d^k \).
It is easy to show that \( ||e_{dp}^k|| \leq ||M_k^T e_k^*|| + ||L_d^k|| \leq m_r \epsilon^*_r + \tau^*_d \).
Then, the error equation in (20) can be converted as:
\[
e^{k+2} = M_k^T k_p e_k^* - M_k^T k_d e^{k-1} + M_k^T \hat{W}_r^{k-1} S_k^T + \tau_{dp}^k
\]
(21)
We define a new error function as follows
\[
e_1^{k+2} = e^{k+2} + M_k^T k_p e_k^* - M_k^T k_d e^{k-1}
\]
(22)
Substituting (20) into (22), the error function \( e_1^{k+2} \) is rewritten as
\[
e_{1}^{k+2} = M_k^T \hat{W}_r^{k-1} S_k^T + \tau_{dp}^k
\]
(23)
It is noted that the error function based on the adaptive RBFNN algorithms (23) is the \((k + 2)\)th step error for robot system, then, we can obtain the \(k\)th step system error by defining \( k_2 = k - 2 \)
\[
e_{1}^k = M_k^T \hat{W}_r^{k-2} S_k^T + \tau_{dp}^2
\]
(24)
where \( m_r \leq ||M_k^T|| \leq m_r \) according to Lemma 1, and
\[
e_{1}^k = e^k + M_k^T k_p e^k - M_k^T k_d e^{k-1}
\]
Based on system tracking error \( e_{1}^k \), RBFNN update rule \( \hat{W}_r^{k+1} \) for (18) is given by
\[
\Delta \hat{W}_r^{k+1} = -\Gamma_r S_r T_\tau \tau_{k}^T
\]
\[
\hat{W}_r^{k+1} = \hat{W}_r^k + \Delta \hat{W}_r^{k+1}
\]
(25)
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 \( \gamma_r > 0 \).

D. Stability Analysis

It has been shown that an ideal control input \( \tau_r^*(\omega_k) \) exists and can guarantee \( e^{k+2} = 0 \), if the unknown disturbance \( \tau_{dp}^k \) is 0. Based on above all assumptions are only valid in compact set \( \Omega_s \), the system all outputs and inputs signal must be prove remain in corresponding compact sets.
A positive definite Lyapunov function \( V_k \) for the system (8) is chosen as
\[
V^k = \sum_{j=0}^{n} tr \left[ \hat{W}_r^{k-2+i} T_\tau \Gamma_r^{-1} \hat{W}_r^{k-2+i} \right]
\]
where \( \hat{W}_r^k = \hat{W}_r^k - W_r^k \).
Note the error function in (24), it is obvious that the Lyapunov function \( V_k \) contains system tracking error, strategic signal error and parameter adaptation for RBFNN weights. The difference of (26) is given by
\[
\Delta V_k = -e^{kT} T_\tau \hat{W}_r^{k-2} S_k^T + b_1 e^T \epsilon_k^1
\]
(27)
where \( b = S_r T_\tau \Gamma_r S_k^T. \)
Defining \( A_k^w = M_k^T \hat{W}_r^{k-2} S_k^T \) and substituting (24) into (27), we have
\[
\Delta V_k = -2(A_k^w + \tau_{dp}^k) T_\tau \hat{W}_r^{k-2} A_k^w
\]
\[
+ b(A_k^w + \tau_{dp}^k) T_\tau (A_k^w + \tau_{dp}^k) \leq -\left( A_k^w + \tau_{dp}^k \right) T_\tau (A_k^w + \tau_{dp}^k)
\]
\[
\times (A_k^w + \tau_{dp}^k) - A_w^T M_k^T A_k^w
\]
\[
+ ||J_k^T||^2
\]
(28)
where \( ||J_k^T||^2 = 2\tau_{dp}^k T_\tau \hat{W}_r^{k-2} \tau_{dp}^k \leq ||J^k||^2 = 2\tau_{dp}^k T_\tau \).
According to Lemma 2, it is easy to know that \( M_k^T \tau_{kn} \) is symmetric positive definite matrix, and it can be bounded with \( m_r \leq ||M_k^T \tau_{kn}|| \leq m_r \). If the eigenvalues of \( M_k^T \tau_{kn} \) are \( \lambda_{kn}^i, i = 1, 2, \ldots, n \), it is obvious that \( \lambda_{kn}^i > 0 \).
We further define \( \lambda_{max} = \max (\lambda_{kn}^1), \lambda_{min} = \min (\lambda_{kn}^1) \), then, \( n \lambda_{min}^i \leq ||M_k^T \tau_{kn}||^2 \) \( = \sum_{i=1}^{n} \lambda_i (M_k^T \tau_{kn})^2 \leq n \lambda_{max}^i \), \( i = 1, 2, \ldots, n \).
For convenience, we define \( P_k^w = M_k^T \tau_{kn} - \lambda_0 \). The matrix \( P_k^w \) being symmetric positive definite can be satisfied under following condition:
\[
1 - b \frac{m_r^*}{\sqrt{n}} > 0
\]
According to the property in (12), it is obvious that \( P_k^w = S_r T_\tau \Gamma_r S_k^T, \gamma_r S_k^T S_k^T < \gamma_r N_r. \) Analyze the difference of Lyapunov function in (28), the design parameter of controller are selected as
\[
0 < \gamma_r < \sqrt{n} N_r m_r^*
\]
(29)
Furthermore, the following theorem is presented to analyze the stability of the system in (8), such that the closed-loop system stability and the trajectory tracking performance can be guaranteed by choosing appropriate parameters and adaptive weight gain of the controller.

Theorem 1: Assume that the conditions set above are satisfied, and define \( B_k^w = A_k^w + \tau_{dp}^k \), then, we have
\[
\Delta V_k \leq -B_k^w T_\tau P_k^w B_k^w + ||J_k^T||^2
\]
(30)
Proof. There exists an invertible matrix \( Q_k^w \) so that \( P_k^w = Q_k^w \). Accordingly, \( \Delta V_k \leq 0 \) can be satisfied under following conditions:
\[
||B_k^w||^2 > ||J_k^T||^2 ||Q_k^w||^{-1}
\]
(31)
A discrete-time delay factor \( z^{-1} \) is introduced in (23), we have
\[
e^k = (I_n + M_k^T k_p z^{-2} - M_k^T k_d z^{-3})^{-1} e_1^k
\]
(32)
According to (31) and (32), we know there exists a finite running step \( K_\tau \), which makes \( ||B_{\tau k^2}w||^2 \leq | ... 1-3. \)

Consider the boundedness of \( M_{\tau k}^2 \) and \( \tau_{dp}^* \), \( B_{\tau k}^2 = A_{\tau k}^2 + I_{dp} \), such that the error \( e_{\tau 1}^k \) is bounded as

\[
||e_{\tau 1}^k||^2 = \|e_{\tau 1}^k\|^2 \leq 2A_{\tau k}^T A_{\tau k} + 2I_{dp} \tau_{dp}^2 < 4|J_{\tau 0}^d|||Q_{\tau}^0||^{-1} + 6\tau_{dp}^2
\]

(33)

or, we can get

\[
||e_{\tau 1}^k||^2 \leq ||e_{\tau 1}^k|| < \sqrt{4|J_{\tau 0}^d|||Q_{\tau}^0||^{-1}} + 6\tau_{dp}^2
\]

(34)

the proof is complete.

V. SIMULATION STUDIES

To verify the above developed adaptive RBFNN control approach, a testing example, 2-DOF robot manipulator interacting is used in this section.

A. Robot Manipulator Dynamics Model

The following parameters of robot manipulator are specified.
The mass are \( m_1 = m_2 = 1.0 \text{kg} \), the length are \( l_1 = l_2 = 0.2 \text{m} \), the inertia are \( I_1 = I_2 = 0.003 \text{kgm}^2 \), the distance are \( l_{c1} = l_{c2} = 0.1 \text{m} \).

Then, dynamics of the robot manipulator with \( G(q) = 0|q| \) is given as

\[
M(q) = [M_{11} M_{12}; M_{21} M_{22}]
\]

\[
C(q) = [C_{11} C_{12}; C_{21} C_{22}]
\]

(35)

where

\[
M_{11} = m_1l_{c1}^2 + m_2(l_{c1}^2 + l_{c2}^2 + 2l_{c1}l_{c2}\cos(q_2)) + I_1 + I_2
\]

\[
M_{12} = M_{21} = m_2l_{c2}^2 + l_1l_{c2}\cos(q_2) + I_2
\]

\[
M_{22} = m_2l_{c2}^2 + I_2
\]

\[
C_{11} = -m_2l_{c2}\sin(q_2)q_2
\]

\[
C_{12} = -m_2l_{c2}\sin(q_2)(q_1 + \dot{q}_2)
\]

\[
C_{21} = m_2l_{c2}\sin(q_2)q_1, \quad C_{22} = 0
\]

The external force torque may be caused by disturbance, a smaller and a larger amplitude force torque \( \tau_{ds} \) and \( \tau_{db} \) are assumed as, respectively,

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

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

Two different types of desired trajectory \( q_{dd} \) and \( q_{dd} \) are assumed as

\[
q_{dd} = [q_{dd,1}, q_{dd,2}]^T = \begin{bmatrix} 1.5 + 0.5\sin(0.3t) + \sin(0.2t) \\ 1.5 + 0.5\cos(1.3t) + \sin(0.2t) \end{bmatrix}
\]

\[
q_{dg} = [q_{dg,1}, q_{dg,2}]^T = \begin{bmatrix} 0.6\text{sign}\left(\cos\left(\pi t/200\right)\right) + 0.4 \\ 0.5\text{sign}\left(\sin\left(\pi t/200\right)\right) \end{bmatrix}
\]

B. Test Results

The initial states of robot manipulator in (35) are \( q(0) = [0, 0]^T \) and \( \dot{q}(0) = [0, 0]^T \). We construct the adaptive RBFNN \( \hat{W}_K \) \( S_k \), which approximates system tracking error using \( N_x = 4096 \) with all the centres of Gaussian function evenly in \([-1, 1]\) and all the widths \( b = 1 \). The design parameters are chosen as \( \gamma_\tau = 0.01, k_p = 6.5, k_d = 115 \),

The initial weights \( W_\tau(0) = 0|2xN_x| \). Simulation results are presented with the controller sampling interval \( T = 0.01s \).

To show the effectiveness, we use the above same design parameters, and compare the position trajectory accuracy and capability between the adaptive RBFNN control and traditional PD control \( \tau^* = -k_p\dot{q} - k_d(q - \dot{q}) \) for the robot manipulator (4) with \( \tau_{ds} \) and \( \tau_{db} \) in Figs. 1-3.

Fig.1-2 show trajectory tracking trajectories and error trajectories of \( q_1 \) and \( q_2 \) for the desired \( q_{dd} \) with added
 Comparing with a traditional PD control based on the above simulation results with a small disturbance signal $\tau_{ds}$, the first joint of the proposed discrete-time adaptive RBFNN control has an initial error and deviates from the desired trajectory for less than 8s, but it can adjust itself quickly to achieve the desired trajectory; and the second joint using the proposed control has also an excellent tracking performance than the traditional PD control. Furthermore, a large disturbance signal $\tau_{db}$ added to test tracking performance of the proposed control for the desired trajectory $q_{db}$, the simulation results are given in Fig.3, which shows that two joints of robot manipulator have achieved satisfied tracking trajectories using adaptive RBFNN control.

VI. CONCLUSION

An discrete-time adaptive RBFNN has been developed for a class of uncertain robot manipulators to achieve precise tracking control performance. The adaptive RBFNN controller is designed to estimate system error, where the control law is adaptively tuned online. Two different types of given trajectory and two kinds of external disturbances are used to test the performance of the proposed approach in the simulation. The proposed discrete-time adaptive RBFNN control is able to overcome effects of external disturbances and internal uncertainties. Not only the closed-loop system stability is guaranteed via Lyapunov stability analysis, but also excellent tracking performance is achieved.

REFERENCES


