
Peer reviewed version

Link to published version (if available):
10.1109/IECON.2016.7792980

Link to publication record in Explore Bristol Research
PDF-document

This is the accepted author manuscript (AAM). The final published version (version of record) is available online via IEEE at DOI: 10.1109/IECON.2016.7792980. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
http://www.bristol.ac.uk/pure/about/ebr-terms
Abstract—In this paper, a radial basis function neural network (RBFNN) based adaptive control is designed for nonlinear robot manipulators. Barrier Lyapunov function (BLF) technique and terminal sliding mode (TSM) technique are seamlessly integrated to achieve finite time convergence of both tracking performance and NN learning performance. BLF is employed to ensure position tracking error converge to a specified small bound in a finite time, and TSM is used to guarantee finite-time convergence of neural learning error to a small bound as well. Extensive simulation studies are performed to illustrate the effectiveness and efficiency of the proposed control method.

Index Terms—Robot manipulator; Adaptive control; Finite-time convergence; BLF, Neural Networks

I. INTRODUCTION

With the increasing needs of robot by our modern society and industry, the research of robot control technologies have attracted enormous attention [1], [2], [3], [4]. In the recent decades, many robotic researchers focus on study of control design in the presence of various constraints since the violation of these constraints may cause collisions and threaten the safety of surrounding environment and the robot itself. In [5], an adaptive controller was developed for robot manipulators to constrained the operation in an circular area to guarantee the safety. A robust adaptive position/force control scheme was proposed to deal with the holonomic constraints of the mobile robots in [6]. Recently, BLFs have been developed in nonlinear control design to deal with the state and output constrains [7], [8], [9], [10], [11], [12]. By adding constraints to the behavior of the state variables or system’s outputs, tracking errors are indirectly constrained with the BLF constraint control method. A BLF-based controller was developed to control a robot manipulator with uncertain dynamics and joint space constraints [7].

The integral BLFs were synthesize in controller to prevent the movement of joint to violate the predefined constraints. In [9], BLFs were incorporated in the adaptive neural network control for a class of nonlinear systems in the presence of unknown functions. In [10], by applying a error transformation, a convenient BLF was constructed in a robust position controller to achieve prescribed performance constraints for a strict feedback nonlinear multiple-input-multiple-output (MIMO) dynamic system. A BLF is employed to deal with the tracking control with full-state constrains for a n-link robot with uncertain dynamics [11]. While in [12], an asymmetric time-varying BLF was presented to ensure the control of strict feedback nonlinear systems to satisfy prescribed constraints. In practice, the transient performance is very important for robot systems. This is because the transient characteristics (e.g. overshoot and convergence rate of tracking errors, amplitudes and frequency of control signals) could greatly influence the system performance.

It is known that finite-time stabilization of dynamical systems may give rise to a high-precision performance besides finite-time convergence to the equilibrium. This can be achieved by some continuous nonsmooth feedback controllers in [16], the approach has been applied to control robot manipulators in [17]. Applying RBFNN control method in infinite-time for robotic system needs learning or renewing the weight terms, which can be considered as the unknown parameter for robotic system. The adaptive parameter estimation schemes are proposed in [18], [19], which exponential and finite-time error convergence are proved without using the derivative of the system states. In [20], [21] neural networks were incorporated into the TSM control design to relax the requirement of system model knowledge and achieve FT error convergence. However, it is noted that the parameter estimation was not addressed in the aforementioned schemes.

Motivated by the above mentioned work, in this paper, we combine BLF and TSM techniques together to design RBFNN based adaptive control for robot manipulators with unknown dynamics. A novel controller is developed with guaranteed tracking performance in both transient and steady state stages, and with finite-time convergence of neural learning performance.

C. Yang is with Centre for Robotics and Neural Systems, Plymouth University, UK.
R. Yang is with College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China.
J. Na is with Department of Mechanical Engineering, University of Bristol, UK.
F. Chen is within the Department of Advanced Robotics, Istituto Italiano di Tecnologia, Italy.
II. PROBLEM FORMULATION AND MODEL DYNAMICS

A. Problem Formulation

The control objective of this paper is to design a robot controller such that the end-effector position $q$ could track a desired trajectory $q_d$ specified in the joint space, while guarantee (i) the tracking errors could achieve predefined transient performances, (ii) all the signals in the robot system remain bounded.

B. Manipulator Dynamics

The dynamic equation of an $n$-link robot manipulator can be described as follows:

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

where $M(q) \in \mathbb{R}^{n \times n}$, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ and $G(q) \in \mathbb{R}^n$ are the inertial matrix, Coriolis and centrifugal matrix and gravitational force vector, respectively, $n$ is the number of robotic joints; $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$ and $\ddot{q} \in \mathbb{R}^n$ are the vectors of the robot arm’s joint position, joint velocity and joint acceleration, respectively and $\tau \in \mathbb{R}^n$ is torque applied on the joints. The following properties will be used in the control design and performance analysis. [15]

Property 1: The inertia matrix $M(q)$ is symmetric and positive definite.

Property 2: The $\dot{M}(q) + 2C(q, \dot{q})$ is a skew symmetric matrix, i.e.,

$$v^T \left( \dot{M}(q) + 2C(q, \dot{q}) \right) v = 0 \quad \forall v \in \mathbb{R}^n$$

C. Preliminaries

In this paper, we use the RBFNN to approximate continues function $F(z) : \mathbb{R}^m \to \mathbb{R}$ as follows,

$$F_{nn}(z) = \sum_{i=1}^{N} w_i s_i(z) = W^T S(z)$$

where $Z \in \Omega_Z \subset \mathbb{R}^m$ is the input vector, $W^T \in \mathbb{R}^{N \times n}$ is the weight vector, $N$ is the number of RBFNN nodes, and $S(z) = [s_1, s_2, \ldots, s_N]^T$ is the regressor vector with $s_i(\cdot)$ being a radial basis function. The most commonly used Gaussian radial basis functions is used as follows:

$$s_i(\|Z - c_i\|) = \exp \left[ -\frac{(Z - c_i)^T(Z - c_i)}{b_i^2} \right]$$

and $b_i$ are distinct points in state space, and $b_i = [b_{i1}, b_{i2}, \ldots, b_{in}]^T$ is the center of the receptive field and $c_i$ is the width of the Gaussian function, $i = 1, \ldots, N$. It has been proven that, with sufficiently large node number, RBFNN (3) can approximate any continuous function $F(z)$ over a compact set $\Omega_Z$ to arbitrary accuracy as

$$f(Z) = W^* S(Z) + \epsilon(Z), \quad \forall Z \in \Omega_Z$$

where $W^*$ is the ideal constant weight vector, $\epsilon(Z)$ is the approximation error such that $|\epsilon(Z)| < \epsilon^*$ with constant $\epsilon^* > 0$ for all $Z \in \Omega_Z$.

Definition 1: [22] A vector $S$ is persistently excited (PE) if there exist $T > 0$, $\iota > 0$ such that $\int_{t}^{t+T} S^T S \geq \iota$.

Lemma 1: [23] If a function $V(t) \geq 0$ with initial value $V(0) > 0$ satisfies the following condition

$$\dot{V} \leq -\kappa V^p, \quad 0 < p < 1.$$  

Then, $V(t) \equiv 0, \forall t \geq t_c$, for a certain $t_c$ that satisfies

$$t_c \leq \frac{V^{-1-p}(0)}{\kappa(1-p)}$$

III. CONTROL DESIGN

Let us defined the tracking error signals of the robot manipulator as

$$\zeta_e = q - q_d$$

$$\zeta_v = \dot{q} - \alpha$$

where $\alpha$ is a virtual controller will be designed latter.

Then, the error equation can be derived from the robot dynamics 1 and (6) as

$$M\dot{\zeta}_v + C\zeta_v = \tau + F_1(z)$$

where $F_1(z) = -(M\alpha + C\alpha + G)$ with $z = [q^T, \dot{q}^T, \alpha^T, \dot{\alpha}^T]^T$, $G$ and $C$ are the abbreviation of $G(q), C(q, \dot{q})$, respectively. It should be noted that $F_1(z) \in \mathbb{R}^n$ is an unknown function vector as the matrices $M, C$ and the vectors $G$ are unavailable. Therefore, the function $F_1(z)$ can not be directly applied in the controller design.

The following assumption is given

Assumption 1: The desired trajectory $q_d$ is chosen so that the $S(z)$ is PE.

Define the symbols $i = 1, 2, \ldots, n$ and $j = 1, 2, 3$ in all following contents.

To formulate the system (7), let us define three alternative vectors as

$$\begin{cases}
F_1(z) = -(M\alpha + C\alpha + G) \\
F_2(z) = M\zeta_v \\
F_3(z) = -M\zeta_v + C\zeta_v
\end{cases}$$

where $F_j(z) \in \mathbb{R}^n$.

Thus, the system (7) can be rewritten by

$$F_2(z) + F_3(z) - F_1(z) = \tau$$

It is well known that RBFNN (3) is applied to approximate the unknown dynamics function, an adaptive parameter estimation method are designed [24].

Let us define alternative vectors as

$$\begin{cases}
F_1(z) = W_{F_1}^T S_1(z) + \varepsilon_1 \\
F_2(z) = W_{F_2}^T S_2(z) + \varepsilon_2 \\
F_3(z) = W_{F_3}^T S_3(z) + \varepsilon_3
\end{cases}$$

where $W_{F_j}^* \in \mathbb{R}^{N_j \times n}$ is optimal weigh matrix; $S_j(z) = [s_{j1}^T, s_{j2}^T, \ldots, s_{jN_j}^T] \in \mathbb{R}^{N_j}$ are the corresponding regression vectors in (4); $\varepsilon_j = [\varepsilon_{j1}, \varepsilon_{j2}, \ldots, \varepsilon_{jn}]^T \in \mathbb{R}^n$ are the approximation error vectors, and $||\varepsilon_j|| \leq \varepsilon^*$ with a positive constant $\varepsilon^*$; $N_j$ are the numbers of neural node of the RBFNN $F_j(z)$.

We can define three new RBFNN functions $\tilde{S}_1(z) = [s_1^T(z), 0_{N_2}^T, 0_{N_3}^T]^T$, $\tilde{S}_2(z) = [0_{N_1}^T, s_2^T(z), 0_{N_3}^T]^T$, $\tilde{S}_3(z) =$
\[
\begin{bmatrix}
    0^T_{N_1}, 0^T_{N_2}, S^T_f(z)\end{bmatrix}^T \in \mathbb{R}^{N_n}, \quad N_n = N_1 + N_2 + N_3 \; \text{and define a new RBFNN weight matrix } W^*(z) \in \mathbb{R}^{N_n \times n} \text{ as}
\]
\[
W^* = \begin{bmatrix}
    W^T_1 & W^T_2 & \cdots & W^T_n
\end{bmatrix} = \begin{bmatrix}
    W^*_{F_1} & W^*_{F_2} & \cdots & W^*_{F_n}
\end{bmatrix}
\]
\[
= \begin{bmatrix}
    W^*_{F_1,1} & W^*_{F_1,2} & \cdots & W^*_{F_1,n}
    W^*_{F_2,1} & W^*_{F_2,2} & \cdots & W^*_{F_2,n}
    W^*_{F_3,1} & W^*_{F_3,2} & \cdots & W^*_{F_3,n}
\end{bmatrix}
\]
\]
where \( W^*_{F,j} = [W^T_{F_1,j}, W^T_{F_2,j}, W^T_{F_3,j}]^T \in \mathbb{R}^{N_n} \) with \( W^*_{F,j} \in \mathbb{R}^{N_j} \) is the \( j \)th column of the \( j \)th RBFNN optimal weight matrix \( W^*_{F,j} \).

Then, the equation (10) can be further formulated as
\[
\begin{bmatrix}
    F_1(z) = W^T \bar{S}_1(z) + \varepsilon_1
    F_2(z) = W^T \bar{S}_2(z) + \varepsilon_2
    F_3(z) = W^T \bar{S}_3(z) + \varepsilon_3
\end{bmatrix}
\]
Consequently, substituting (12) into (9), we have as
\[
W^T \bar{S}(z) = \tau + \bar{\varepsilon}
\]
where \( \bar{\varepsilon} = \varepsilon_1 - \varepsilon_3 \) is RBFNN construction error vector with \( \|\bar{\varepsilon}\| < \varepsilon^*, \varepsilon^* \) is a positive constant, \( \bar{S}(z) = \bar{S}_2(z) + \bar{S}_3(z) - \bar{S}_1(z) = [-S^T_1(z), S^T_2(z), S^T_3(z)]^T \in \mathbb{R}^{N_n} \) is a new RBFNN basic function vector.

Consequently, using RBFNN method, the system (13) can be divided into \( n \) subsystems as
\[
W^T \bar{S}(z) = \tau + \bar{\varepsilon}
\]
where \( \tau_i \) and \( \bar{\varepsilon}_i \) are control input and RBFNN approximation error of the \( i \)th subsystem, respectively.
\[
\tau_i = -k_{2i} \zeta \varepsilon_i - \bar{\gamma}_i \varepsilon_i - k_{3i} \zeta \frac{\varepsilon_i}{\zeta} - W^T \bar{S}_i(z)
\]
where \( k_{2i} \) and \( k_{3i} \) are designed positive constant, \( \bar{\gamma}_i \) is designed in (38), \( \zeta = [\zeta_1, \zeta_2, \cdots, \zeta_n]^T \) and \( \zeta = [\zeta_1, \zeta_2, \cdots, \zeta_n]^T \) are defined in (6), \( \bar{W}_i \) is the estimate of \( W^* \).

To design the optimal adaptive estimation law of weight vectors \( \bar{W}_i \), a novel adaptive parameter estimation in are introduced [24].

We first design the following filters
\[
\begin{bmatrix}
    k \bar{S}_{i,1} + \bar{S}_{i,1} = \bar{S}_1, \quad S_{1, t=0} = 0_{[N_n]} \\
    k \bar{S}_{i,2} + \bar{S}_{i,2} = \bar{S}_2, \quad S_{2, t=0} = 0_{[N_n]} \\
    k \bar{S}_{i,3} + \bar{S}_{i,3} = \bar{S}_3, \quad S_{3, t=0} = 0_{[N_n]} \\
    k \hat{\tau}_i + \tau_i = \tau, \quad \tau_{f, t=0} = 0
\end{bmatrix}
\]
where, \( k > 0 \) is a filter parameter, \( \bar{S}_{i,j} = \bar{S}_{i,j}(z) \in \mathbb{R}^{N_n} \) and \( \tau_{f} \in \mathbb{R} \) are the filtered variables, respectively.

The filter operations are applied to the equation (14), such that a corresponding equation can be obtained as follows
\[
W^* \bar{S}_f = \tau_i + \bar{\varepsilon}_f
\]
where \( \bar{S}_f = \frac{S_2 - S_1}{k} + \frac{S_3 - S_1}{k} \in \mathbb{R}^N \) is a new RBFNN function vector, \( \bar{\varepsilon}_f \) can only be used for analysis from \( k \hat{\varepsilon}_f + \bar{\varepsilon}_f = \bar{\varepsilon}_f \) with \( \bar{\varepsilon}_f(0) = 0 \). It is clear that the \( W^* \) can be considered as unknown parameters in (17), which needs be estimated as \( \hat{W} \) during control designation.

To accommodate parameter estimation, the matrix \( P \in \mathbb{R}^{N_n \times N_n} \) and vector \( Q_i \in \mathbb{R}^{1 \times N_n} \) are defined as follows
\[
\begin{bmatrix}
    \dot{\hat{P}} = -\kappa_i P + \bar{S}_i \bar{S}_f^T, \quad P(0) = 0_{N_n \times N_n} \\
    \dot{\hat{Q}}_i = -\kappa_i Q_i + \tau_i \bar{S}_f, \quad Q_i(0) = 0_{N_n}
\end{bmatrix}
\]
Considering (17), it is clear that \( Q_i = P^T \bar{W}_i - \mu_{\varepsilon_i} \) with \( \mu_{\varepsilon_i} = \int_0^T e^{-\kappa_i(t-r)} \bar{\varepsilon}_f \bar{S}_f(r) dr \in \mathbb{R}^{N_n} \) in (18), such that the parameter
\[
R_i = P^T \bar{W}_i - Q_i
\]
\[
= P^T \bar{W}_i + \mu_{\varepsilon_i}
\]
where \( \mu_{\varepsilon_i} \) is bounded, definition (1) implies \( \bar{S}_i \), bounded, and \( v_{\varepsilon_i} \) is bounded according to (13), then, we have \( \|\mu_{\varepsilon_i}\| \leq \zeta_\varepsilon^* \) for a constant \( \zeta_\varepsilon^* > 0 \).

**Lemma 2:** [18] The matrix \( P \) is positive definite satisfying \( \lambda_{\min}(P(t)) > \delta_p \) for \( t > T \) and \( \sigma > 0 \), \( T > 0 \), provided the NN function \( \bar{S}(z) \) is PE in Definition (1).

The RBFNN weight estimation \( \hat{W}_i \) in (18) can be obtained by designing the following adaptive law
\[
\hat{W}_i = \Gamma_i \left( \zeta v_\varepsilon \bar{S}_1 - \frac{P^T R_i}{\|R_i\|} \right)
\]
where \( \Gamma_i \in \mathbb{R}^{N_n \times N_n} \) is a positive definitive matrix, and \( \gamma_i \) is a positive constant.

**A. Predefined Tracking Performance**

As mention above, for tracking errors \( \zeta_e \) in (2), our controller design objective is to make \( q(t) \) track a predefined trajectory \( q_0(t) \) while guarantee \( \zeta_e(t) \) satisfying the predefined transient performance. At first, let us define a smooth decreasing performance function which could describe the transient performance of tracking errors as \( \phi(t) = [\phi_1(t), \phi_2(t), \cdots, \phi_n(t)]^T \)
\[
\phi(t) = \begin{bmatrix}
    (\rho_{10} - \rho_{\infty}) e^{-\alpha_1 t} + \rho_{\infty} \\
    (\rho_{20} - \rho_{\infty}) e^{-\alpha_2 t} + \rho_{\infty} \\
    \vdots \\
    (\rho_{n0} - \rho_{\infty}) e^{-\alpha_n t} + \rho_{\infty}
\end{bmatrix}
\]
where \( \rho_{10}, \rho_{20}, \cdots, \rho_{n0}, \rho_{\infty}, \rho_{\infty}, \rho_{\infty}, \cdots, \rho_{\infty} \), and \( \alpha_1, \alpha_2, \cdots, \alpha_n \) are properly chosen positive constants. The performance functions \( \phi_1, \phi_2, \cdots, \phi_n \) are smooth, bounded and positive functions with \( \lim_{t \to \infty} \phi_i(t) = \rho_{\infty} \).

To ensure the tracking error satisfying the prescribed tracking transient, we depicted the bounds of the tracking errors as \(-\beta_1 \phi(t) < \zeta_e < \beta_2 \phi(t)\), where \( \beta_1 \) and \( \beta_2 \) are positive design constants. The functions \( \beta_1 \phi(t) \) and \( -\beta_2 \phi(t) \) describe the tracking transient performance with \( \alpha_i \) regulates the lower bounded of the required convergence rate of tracking errors, while \( \beta_1 \rho_{\infty} \) and \( -\beta_2 \rho_{\infty} \) define the maximum overshoot and undershoot of the tracking errors. Thus we can regulate the transient performance and the steady-state stages by properly select the function \( \phi_i \) and the designed parameters \( \beta_1, \beta_2 \).
B. Stability Analysis

Theorem 1: Consider the robot manipulator (1) with the tracking error (6), employ the global NN controller design (15) with the NN weight adaptive law (20) and the prescribed transient performance (21), then, we have all the tracking signals are UUB and the tracking error coverage to a small neighborhood of zero; the predefined transient and tracking performance is guaranteed.

Considering the following Lyapunov function

\[ V = V_1 + V_2 + V_3 \]  

(22)

\[ V_1 = \sum_{i=1}^{n} \left( \frac{h_i}{2} \ln \frac{1}{1 - \xi_{bi}^2} + \frac{1 - h_i}{2} \ln \frac{1}{1 - \xi_{ai}^2} \right) \]  

(23)

where \( \xi_{ai} \) and \( \xi_{bi} \) are designed by applying coordinate transformations on the tracking error \( \xi_r \), we have

\[ \xi_a = \begin{bmatrix} \xi_{11} & \xi_{12} & \cdots & \xi_{1n} \\ \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1n} \end{bmatrix} \]

\[ \xi_b = \begin{bmatrix} \xi_{21} & \xi_{22} & \cdots & \xi_{2n} \\ \varphi_{21} & \varphi_{22} \end{bmatrix} \]  

(24)

\[ \xi = h_i(\xi_{ai})\xi_{bi} + (1 - h_i(\xi_{ai}))\xi_{ai} \]  

(25)

where \( \varphi_{ai}(t) = -\beta_i(\phi_i(t), \varphi_{ai}(t)) \) and \( \varphi_{bi}(t) = \beta_i(\phi_i(t), \varphi_{bi}(t)) \) are the ith element of the vectors \( \xi_a \), \( \xi_b \), respectively, and \( h_i(\xi_{ai}) \) is defined as

\[ h_i = \begin{cases} 1 & \xi_{ai} \geq 0 \\ 0 & \xi_{ai} < 0 \end{cases} \]  

(26)

Then, we design a virtual controller as

\[ \sigma_i(t) = q_{di} - k_{1i}\zeta_{ei} + \sigma_i(t)\zeta_{ei} \]  

(27)

where

\[ \sigma_i(t) = \sqrt{\left( \frac{\hat{\varphi}_{ei}}{\varphi_{ei}} \right)^2 + \left( \frac{\hat{\varphi}_{2i}}{\varphi_{2i}} \right)^2 + k_{ai}} \]  

(28)

\( k_{ai} \) and \( k_{1i} \) are designed positive constants. Notice that the following inequality holds

\[ \sigma_i(t) + h_i(\varphi_{ei}) + (1 - h_i(\varphi_{ei})) \geq 0 \]  

(29)

Substituting (32) into (31) yields

\[ \dot{V}_1 = \sum_{i=1}^{n} \left( \left( \frac{h_i(\xi_{ei})}{(1 - \xi_{ei}^2)\varphi_{ei}} + \frac{(1 - h_i(\xi_{ai}))}{(1 - \xi_{ai}^2)\varphi_{ei}} \right) \right) \]  

(30)

\[ \xi_{ei}(t) + \xi_{ei}(t) = \xi_{ei}(t) + \xi_{ei}(t) \]  

(31)

And in terms of (34), we have

\[ \dot{V}_1 \leq \sum_{i=1}^{n} \left( \left( \frac{h_i}{(1 - \xi_{ei}^2)\varphi_{ei}} + \frac{(1 - h_i)}{(1 - \xi_{ai}^2)\varphi_{ei}} \right) \right) \]  

(32)

Noting that the following inequality exists,

\[ \frac{\xi_{ei}^2}{(1 - \xi_{ei}^2)} \geq \ln \frac{1}{(1 - \xi_{ei}^2)} \quad \forall |\xi_{ei}| < 1 \]  

(33)

Substituting (37) into (36), and using the \( \varphi_{ei} \) to represent \( h_i/(\varphi_{ei}^2 - \xi_{ei}^2) + (1 - h_i)/(\varphi_{ei}^2 - \xi_{ei}^2) \), we can obtain that

\[ \dot{V}_1 \leq \sum_{i=1}^{n} \left( \frac{k_{1i}}{(1 - \xi_{ei}^2)} + \varphi_{ei}(t) \right) \]  

(34)

Let us take the derivative of \( V_2 \) in (26), substitute (1) and (6) into (39), we can obtain that

\[ \dot{V}_2 = \frac{1}{2} \zeta_{ei}^T M \zeta_{ei} + \zeta_{ei}^T M \dot{\zeta}_{ei} \]  

(35)

[\text{where } F_{ei}(z) \text{ have be described in (7).}]

Substituting the control input (15) into (39) as

\[ \dot{V}_2 = \sum_{i=1}^{n} \left( \eta_i + \zeta_{ei} \left( \dot{W}_i^T S_1(z) - W_i^* T S(z) + \varepsilon_i \right) \right) \]  

(36)

where \( \eta_i = -k_{ei} \zeta_{ei}^2 - k_{2i} \zeta_{ei}^2 - 2 \zeta_{ei} \eta \zeta_{ei} \zeta_{ei} \).

According to the Young’s inequality, the following relation can be easily obtained

\[ \zeta_{ei} \leq \frac{1}{2} \zeta_{ei}^2 + \frac{1}{2} \zeta_{ei}^2 \]  

(37)

\[ \dot{V}_3 = \frac{1}{2} \zeta_{ei}^T M \zeta_{ei} + \zeta_{ei}^T M \dot{\zeta}_{ei} \]  

(38)

\[ \dot{V}_3 = \sum_{i=1}^{n} \left( \eta_i + \zeta_{ei} \left( \dot{W}_i^T S_1(z) - W_i^* T S(z) + \varepsilon_i \right) \right) \]  

(39)

\[ \zeta_{ei} \leq \frac{1}{2} \zeta_{ei}^2 + \frac{1}{2} \zeta_{ei}^2 \]  

(40)
Substituting (41) into (40), and considering $\|\varepsilon_j\| < \varepsilon^*$, we have
\[
V_2 \leq \sum_{i=1}^{n} \left( \eta_i - \zeta_{vi} \bar{W}_i^{T} S_1 + \frac{1}{2} \zeta_{vi}^2 + \frac{1}{2} \varepsilon^* \right) \tag{42}
\]
Differentiating the second equation of (26) with respect to time, we can obtain $\dot{V}_3$ as
\[
\dot{V}_3 = \sum_{i=1}^{n} \left( R_i^{T} P^{-T} \Gamma_{i}^{-1} \frac{\partial (P^{-T} R_i)}{\partial t} \right) \tag{43}
\]
According to (18) and (19), we have
\[
\frac{\partial (P^{-T} R_i)}{\partial t} = \frac{\partial (W_i + P^{-T} \mu_{xi})}{\partial t} = \dot{W}_i - P^{-T} \dot{P} \dot{P} P^{-T} \mu_{xi} + P^{-T} \mu_{xi}
\]
\[
= \dot{W}_i + \bar{\mu}_{xi} = \dot{W}_i + \bar{\mu}_{xi}
\]
where $\bar{\mu}_{xi} = P^{-T} \dot{\mu}_{xi} - P^{-T} \dot{P} \mu_{xi} \in \mathbb{R}^{n_x}$.
Substituting (44) into (43), and considering the equation (19) differentiation of $V_3$ can be written as
\[
\dot{V}_3 = \sum_{i=1}^{n} \left( R_i^{T} P^{-T} \Gamma_{i}^{-1} (\dot{W}_i + \bar{\mu}_{xi}) \right)
= \sum_{i=1}^{n} \left( R_i^{T} P^{-T} \Gamma_{i}^{-1} (\Gamma_i (\zeta_{vi} S_1 - \gamma_i \frac{PR_i}{\|R_i\|}) + \bar{\mu}_{xi}) \right)
= \sum_{i=1}^{n} \left( R_i^{T} P^{-T} \zeta_{vi} S_1 - \gamma_i \frac{R_i^{T} P^{-T} PR_i}{\|R_i\|} \right)
+ \sum_{i=1}^{n} R_i^{T} P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}
\leq \sum_{i=1}^{n} \left( \zeta_{vi} \bar{W}_i^{T} S_1 + |\zeta_{vi}||\mu_{xi}|P^{-T} S_1 \right)
- \sum_{i=1}^{n} \left( (\gamma_i - \|P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}\|)|R_i| \right) \tag{45}
\]
Let us combine (38), (42) with (45),
\[
\dot{V}_1 + \dot{V}_2 + \dot{V}_3 \leq \sum_{i=1}^{n} \left( - \frac{k_{ii}}{1 - \xi_i^{o}} - k_{ii} \xi_{vi} + \frac{1}{2} \zeta_{vi}^2 + \frac{1}{2} \varepsilon^* \right)
+ \sum_{i=1}^{n} \left( - k_{3i} \zeta_{vi}^2 + |\zeta_{vi}||\mu_{xi}|P^{-T} S_1 \right)
- \sum_{i=1}^{n} \left( \gamma_i - \|P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}\|)|R_i| \right)
\leq - \sum_{i=1}^{n} \left( (k_{2i} - \frac{1}{2}) \zeta_{vi}^2 \right)
- \sum_{i=1}^{n} \left( \frac{k_{ii}}{1 - \xi_i^{o}} - \frac{1}{2} \varepsilon^* \right)
- \sum_{i=1}^{n} \left( \gamma_i - \|P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}\|)|R_i| \right)
\leq - \sum_{i=1}^{n} \left( (k_{3i} - \gamma_i)|\zeta_{vi}| \right)
- \sum_{i=1}^{n} \left( (k_{3i} - \|P^{-T} \zeta_{vi} S_i\|)|\zeta_{vi}| \right)
\tag{46}
\]
According assumption (1), and noting that $\varepsilon_j$ and $\bar{S}_j$ are bounded with $\|\varepsilon_j\| \leq \varepsilon^*$ and $\bar{S}_j = t, T > 0$, then, $\mu_{xi}$ and $\bar{\mu}_{xi}$ are bounded in finite-time interval. Considering the lemma (2), $P$ is bounded in magnitude, thus $\|\mu_{xi}|P^{-T} S_1\|$ and $\|P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}\|$ exist and are bounded as long as all closed-loop system parameters are suitably chosen. Then, the equation (22) is semiglobal stability for enough large $k_{3}, \gamma_i$, and $k_{3i}$ and $k_{3i} \geq \frac{1}{2}$, we have
\[
\dot{V} \leq \sum_{i=1}^{n} \left( k_{2i} - \frac{1}{2} \zeta_{vi}^2 \right) \leq 0 \tag{47}
\]
The inequation (47) further implies $\lim_{t \to \infty} \zeta_{vi} = 0$. Thus, the control error $\zeta_{vi}$ converges to zero and all other signals in the closed-loop are bounded.
To further prove finite-time convergence, we substitute (26) into $V_{23} = V_2 + V_3$, thus
\[
V_{23} = \frac{1}{2} \zeta_{vi}^{T} M \zeta_{vi} + \frac{1}{2} R_i^{T} P^{-T} \Gamma_{i}^{-1} P^{-T} R_i \tag{48}
\]
Then, the time differentiation of (48) is written as
\[
\dot{V}_{23} \leq \sum_{i=1}^{n} \left( -k_{2i} \zeta_{vi}^2 - k_{3i} \zeta_{vi}^2 + \gamma_i \bar{\zeta}_{vi} \bar{S}_j + \bar{\zeta}_{vi} \bar{S}_j \right)
+ \sum_{i=1}^{n} \left( \zeta_{vi} |\mu_{xi}|P^{-T} S_1 \right) - \sum_{i=1}^{n} \left( (\gamma_i - \|P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}\|)|R_i| \right)
\leq - \sum_{i=1}^{n} \left( k_{3i} - \gamma_i |\zeta_{vi}| \right)
\tag{49}
\]
The bounded analysing for $\|\mu_{xi}|P^{-T} S_1\|$ and $\|P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}\|$ represented above as long as all closed-loop system parameters are suitably chosen. Then, $V_{23}$ is semiglobal stability for enough large $\gamma_i$ and $k_{3i}$, the semiglobal stability of (49) follows such that
\[
V_{23} \leq - \sum_{i=1}^{n} k_{2i} \zeta_{vi}^2 = - \bar{\zeta}_{vi}^{T} k_{2i} \zeta_{vi} \leq 0 \tag{50}
\]
with $k_{2i} = [k_{21}, k_{22}, \ldots, k_{2n}]$.
The inequation (49) can be represented as
\[
\dot{V}_{23} \leq - \sum_{i=1}^{n} \left( (k_{3i} - \gamma_i |\zeta_{vi}|) - \varepsilon^* - \|\mu_{xi}|P^{-T} S_1\| \right) \tag{51}
\]
According Lemma (2) and the Lyapunov function $V_{23}$ in (48), the inequation (51) can be rewritten as
\[
\dot{V}_{23} \leq \mathcal{K} V_{23} \tag{52}
\]
with $l = 1/2$ and
\[
\mathcal{K} = \sum_{i=1}^{n} \left( \min \{k_{3i}, 3 \gamma_i \} - \varepsilon^* - \|\mu_{xi}|P^{-T} S_1\| \right) \times \sqrt{2/\lambda_{max} (M)}.
\]
\[
(\gamma_i - \|P^{-T} \Gamma_{i}^{-1} \bar{\mu}_{xi}\|) \delta \sqrt{2/\lambda_{i}(\Gamma_{i}^{-1})} \}
\]
Noting the inequation (52) and applying Lemma 1, we have \( V_{23} \equiv 0, \forall t \geq t_c \) with the finite-time
\[
t_c \leq 2KV_{23}^{1/2}(0)
\] (53)
Consequently, combination \( V \) (22), \( \dot{V} \) (47), \( V_{23} \) (48) and \( \dot{V}_{23} \) (52), finite-time convergence of the tracking error \( \zeta_c, \zeta_v \) and \( R \) to zero is guaranteed, which implies \( \lim_{t \to \infty} \dot{W}TP = \mu_c \). This complete the proof.

IV. SIMULATION STUDIES

In this section, simulation studies are carried out to illustrate the effectiveness of the proposed adaptive RBFNN control algorithm (15). In the simulation, we employ a 2-link manipulator model whose dynamics is given by [11]:
\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau
\] (54)
where \( q = [q_1, q_2] \) is a vector of joint variables, and
\[
G(q) = \begin{bmatrix} G_{11} \\ G_{21} \end{bmatrix}
\] (55)
\[
M(q) = \begin{bmatrix} M_{11} & M_{12} \\ m_{21} & M_{22} \end{bmatrix}
\] (56)
\[
C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}
\] (57)
The plant parameters are chosen as follows:
\[
G_{11} = (m_1l_2 + m_2l_1)g \cos q_1 + m_2l_2g \cos(q_1 + q_2)
G_{21} = m_2l_2g \cos(q_1 + q_2)
M_{11} = m_1l_2^2 + m_2(l_1^2 + l_2^2 + 2l_1l_2 \cos q_2) + I_1 + I_2
M_{12} = m_2(l_2^2 + l_1l_2 \cos q_2) + I_2
M_{22} = m_2l_2^2 + I_2
C_{11} = -m_2l_1l_2(\dot{q}_1 + \dot{q}_2) \sin q_2
C_{12} = -m_2l_1l_2(\dot{q}_1 + \dot{q}_2) \sin q_2
C_{21} = m_2l_1l_1 \dot{q}_1 \sin q_2
C_{22} = 0
\]
with \( m_1 = 2kg, m_2 = 0.85kg, l_1 = 0.35m, l_2 = 0.31m, I_1 = 0.061kgm^2 \) and \( I_2 = 0.020kgm^2 \), and \( m_i \) and \( l_i \) are the mass and length of link \( i \), \( l_{ci} \) is the distance between \( i \) - th joint and the \( j \) th link’s mass center, \( i = 1, 2 \). And \( I_i \) is the inertia of link \( i \).

The reference trajectory \( q_d \) is chosen as \( q_d = [\sin(0.5t), 2 \cos(0.5t)]^T \), where \( t \in [0, t_f] \) and \( t_f = 15s \). The initial state are set as \( q = [-1, 3] \). While to guarantee the transient performance, the prescribed performance functions are designed as \( \phi_1(t) = (1 - 0.05)e^{-t} + 1, \phi_2(t) = (1 - 0.03)e^{-t} + 1 \), i.e. the tracking error are bounded by
\[
-\beta_1\phi_1(t) \leq \zeta_{11}(t) < -\beta_2\phi_1(t) \quad i = 1, 2
\] (58)
with \( \beta_1 = 1, \beta_2 = 1 \). The control gains are selected as \( k_1 = [20, 20]^T, k_2 = [10, 10]^T, k_3 = [30; 30] \), the gains of NN adaptive laws are chosen as \( \Gamma_1 = 0.01I_{N_x \times N_x}, \Gamma_2 = 0.01I_{N_x \times N_x} \), and the parameter \( \gamma = [1; 1] \). The simulation results are shown in Figs.1-4. As shown in Fig.1 and Fig.3, we see clearly that the \( q_1, q_2, \dot{q}_1 \) and \( \dot{q}_2 \) could effectively follow the reference trajectories, which means that the proposed controller can achieve a good tracking in the presence of unknown manipulator dynamics. While Fig.2 shows the tracking errors \( \zeta_{c1}, \zeta_{c2} \) coverage to a small value close to zero quickly. The simulation results in Fig.2 also illustrate that our proposed adaptive RBFNN with time-vaey-BLF controller has guaranteed the tracking errors always remain in the predefined region and the prescribed transient performances are never violated. While the corresponding control input is depicted as shown in Fig.4.

V. CONCLUSION

In this paper, we have investigated the adaptive neural network control for robot manipulators with unknown dynamics. Controller designed using adaptive RBFNN and time-varying BLF techniques achieves predefined transient performance and guarantee finite time convergence of RBFNN learning. Leakage terms, functions of the estimation error, are incorporated into the adaptation laws to avoid windup of the adaptation
algorithms. Simulation results have demonstrated the effectiveness and efficiency of the proposed control scheme.

REFERENCES


