INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2017; **27**:875–893
Published online 26 July 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.3604

$\mathcal{L}_2$ performance control of robot manipulators with kinematics, dynamics and actuator uncertainties

Liang Xu[1], Qinglei Hu[2,*,†] and Youmin Zhang[3]

[1]*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore*
[2]*School of Automation Science and Electrical Engineering, Beihang University, Beijing, China*
[3]*Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Quebec, Canada*

SUMMARY

This paper deals with the task-space trajectory tracking control problem of robot manipulators. An improved adaptive backstepping controller is proposed to deal with the uncertainties in kinematics, dynamics, and actuator modeling. To avoid the explosion of computation in conventional backstepping techniques, a modified dynamic surface control algorithm is proposed, which guarantees the asymptotic convergence rather than the uniformly ultimately boundedness of tracking errors in conventional dynamic surface control methods. Furthermore, the expression of the $\mathcal{L}_2$ norm of tracking errors is explicitly derived in relation to the controller parameters, which provides instructions on tuning controller parameters to adjust the system performance. Moreover, the passivity structure of the designed adaptation law is thoroughly analyzed. Simulation of a free-floating space robot is used to verify the effectiveness of the proposed control strategy in comparison with the conventional tracking control schemes.

Received 19 July 2015; Revised 22 February 2016; Accepted 17 June 2016

KEY WORDS: robot manipulators; tracking control; adaptive dynamic surface control; uncertainties

1. INTRODUCTION

The design of a robust control system for robot manipulators in the presence of model uncertainties is one of the most challenging tasks for control engineers. In the past decades, several control algorithms are proposed to fulfill this objective (for example, References [1, 2]). The design of those controllers relies on the assumption that the exact kinematics (i.e., the Jacobian matrix) of robot manipulators is known. Unfortunately, in real world, physical parameters cannot be measured precisely. Furthermore, when the robot picks up objects of different lengths or unknown orientations, the overall kinematics is changing and therefore difficult to derive the mathematical model exactly. All these difficulties suggest that the kinematic uncertainties must be taken into consideration when designing task-space tracking controllers for robot manipulators.

To solve the regulation problem in the presence of kinematic uncertainties, proportional-differential (PD) [3, 4], adaptive Jacobian [5, 6], transpose Jacobian [7], and inverse Jacobian [8] control laws have been proposed for fixed-base manipulators to achieve task-space regulation objectives. Actually, both the latter two strategies are dual, as stated in Reference [9]. Amplitude-limited torque input controller [10] and adaptive set point controller [11] are proposed with an emphasis on the actuator constraints. Task-space tracking control problem, on the other hand, is also highly concerned in the robot research community. Adaptive Jacobian [12–17], inverse kinematics [18, 19], and prediction error-based [20, 21] control algorithms are proposed to cope

*Correspondence to: Qinglei Hu, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China.
†E-mail: huql_buaa@buaa.edu.cn

with robot tracking control problems in the presence of both kinematic and dynamic uncertainties. In addition, both neural [22] and adaptive Jacobian controllers [23] are proposed when taking uncertain actuator models into consideration. However, in the aforementioned control schemes, the linear-in-parameters assumption for uncertainties in kinematics is needed. Unfortunately, for some specific robot manipulator systems, such as the free-floating space manipulators, owing to the second-order non-holonomic constraints caused by the conservation of angular momentum, the generalized Jacobian matrix [24] of the free-floating space robot is no longer linear with respect to a set of physical parameters [25].

Backstepping technique provides a systematic approach for construction of Lyapunov functions to a broad class of strict-feedback nonlinear systems. Because of its simple configuration and ease of implementation, this control strategy has been widely applied to various control systems, such as attitude control systems [26], robot manipulators [27], unmanned aerial vehicles [28], and multi-agent systems [29]. However, there is one main issue with this method, that is, the explosion of complexity caused by the need of the derivatives of the designed virtual input. It requires also the known system functions to be C^n smooth when the nonlinear system has a relative degree of n. To overcome the aforementioned drawbacks of the backstepping design, a new technique named dynamic surface control (DSC) was proposed [30, 31]. By introducing a low-pass filter to prevent the derivative of nonlinear functions at each design step, the phenomenon of explosion of complexity is eliminated. Thus, the dynamic surface control methods are widely developed [32–35]. However, the results of those existing research works only guarantee the uniform and ultimate boundedness of the tracking error. For more accurate control purpose, the asymptotic stability of the tracking error should be achieved.

This paper deals with the task-space trajectory tracking control of robot manipulators. An improved intelligent dynamic surface controller is proposed to deal with the kinematic, dynamic, and actuator uncertainties. The proposed controller does not require the kinematics to be linear with respect to a set of physical parameters. Besides, unlike conventional dynamic surface control methods, the asymptotic stabilization of the tracking error can be achieved. Furthermore, the explicit relation among the $\mathcal{L}_2$ transient performance of the tracking error and the controller gains is deduced, and thus, the control performance can be guaranteed by appropriately selecting controller parameters.

The paper is organized as follows: In Section 2, a formal problem statement accompanied by all the governing equations is presented. In Section 3, the control scheme and the associated stability analysis for the resulting closed-loop dynamics are investigated. Section 4 provides the passivity analysis of the designed controller. Numerical simulation results are presented in Section 5, and the paper is ended with some concluding remarks.

Notions: Throughout the paper, let $\mathbb{R}$ be the real number and $\mathbb{R}^n$ denote the space of real n-dimensional vector. $\mathbf{1}^n$ is an n-dimensional vector with all elements equal to one. $\boldsymbol{A}^{\mathrm{T}}$ and $\boldsymbol{A}^{-1}$ denote the transpose and inverse/pseudo-inverse of the matrix/vector $\boldsymbol{A}$. The norm of vector $\boldsymbol{x}$ is defined as $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$. The matrix norm is defined as $\|\boldsymbol{A}\| = \sqrt{\lambda_{\max}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}$, with $\lambda_{\max}(\cdot)$ denoting the maximum eigenvalue of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$. The Frobenius norm of matrix is defined as the root of the squared sum of all diagonal elements, that is, $\|\boldsymbol{A}\|_F^2 = \mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$, where the trace $\mathrm{tr}(\boldsymbol{A})$ satisfies $\mathrm{tr}(\boldsymbol{A}) = \mathrm{tr}(\boldsymbol{A}^{\mathrm{T}})$, $\mathrm{tr}(\boldsymbol{BC}) = \mathrm{tr}(\boldsymbol{CB})$ for any $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{m\times n}$, and $\boldsymbol{C} \in \mathbb{R}^{n\times m}$. For piecewise continuous, squared, and integrable function $\boldsymbol{f} : [0,\infty) \to \mathbb{R}^n$, the $\mathcal{L}_2$ norm is defined by $\|\boldsymbol{f}\|_{\mathcal{L}_2} = \sqrt{\int_0^\infty \boldsymbol{f}(t)^{\mathrm{T}}\boldsymbol{f}(t)\mathrm{d}t}$. We say $\boldsymbol{f}$ is bounded (i.e., $\boldsymbol{f} \in \mathcal{L}_\infty[0,\infty)$) if $\|\boldsymbol{f}(t)\| < \infty$ for all $t \in [0,\infty)$.

2. MODELING AND PROBLEM FORMULATION

The dynamics of a n-link, serially connected, revolute rigid robot manipulator is given as follows [2]

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{V}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{1}$$

DOI: 10.1002/rnc

where $\boldsymbol{q} \in \mathbb{R}^n$ denotes the manipulator joint angles; $\boldsymbol{B}(\boldsymbol{q})$ represents the positive-definite and symmetric robot inertial matrix; $\boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ contains the nonlinear centripetal, Coriolis and gravitational forces and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the torque applied on manipulators.

Let $\boldsymbol{x} \in \mathbb{R}^m$ represent a task-space vector relating to the position and orientation of manipulator end-effectors, then the motion rate of the end-effector is related to that of joint angles by

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2}$$

The matrix $\boldsymbol{J} \in \mathbb{R}^{m\times n}$ is the Jacobian matrix. For simplicity, the manipulator is assumed to be driven by armature-controlled direct current (DC) motors. The dynamics of this type of motor is described as follows [22]

$$\boldsymbol{\tau} = \boldsymbol{k}_\tau \boldsymbol{i}, \ \boldsymbol{L}\dot{\boldsymbol{i}} + \boldsymbol{R}\boldsymbol{i} + \boldsymbol{k}_e\dot{\boldsymbol{q}} = \boldsymbol{u} \tag{3}$$

where $\boldsymbol{i}, \boldsymbol{u} \in \mathbb{R}^n$ denote the armature current and the voltage vector, respectively; $\boldsymbol{k}_\tau \in \mathbb{R}^{n\times n}$ is the positive definite constant diagonal matrix characterizing electro-mechanical conversion between current and torque; $\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{k}_e \in \mathbb{R}^{n\times n}$ are positive-definite constant diagonal matrices representing the circuit inductance, resistance, and back electromotive force constant of the motor, respectively.

This paper concerns with the inertially referenced task-space trajectory tracking control problems in the presence of kinematics, dynamics, and actuator uncertainties. For general robot manipulators, the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$ is linear with respect to a set of physical parameters. The adaptive Jacobian scheme proposed in [14, 23] can be applied to solved this problem. However, for some special manipulator systems, such as the free-floating space manipulators, the Jacobian cannot be written in a linear form with respect to a set of physical parameters [25]. It is thus impossible to directly extend the existing adaptive Jacobian schemes to free-floating space manipulators, and this fact motivated us to provide a potential solution to such a control design problem. By exploiting the unique learning and nonlinearities handling capability of neural networks, together with first-order low-pass filter with time-varying gain techniques, an improved intelligent/adaptive dynamic surface controller is proposed for solving task-space trajectory tracking control problem of such robotic systems in this paper as to be presented subsequently.

3. MAIN RESULTS

Careful inspection of the dynamics (1), (2) and (3) suggests a cascade structure that is suitable for a backstepping control design. A revised adaptive/intelligent dynamic surface control technique is therefore proposed in this section to avoid the explosion of complexity and also to achieve asymptotic convergence of the tracking error.

3.1. Controller synthesis

Based on the estimated physical parameters of the robot manipulator system, the nominal values for system matrices can be calculated and nominated here as $\boldsymbol{J}_0, \boldsymbol{B}_0, \boldsymbol{V}_0, \boldsymbol{L}_0, \boldsymbol{R}_0, \boldsymbol{k}_{\tau 0}, \boldsymbol{k}_{e0}$. The subsequent development is based on the assumption that all kinematic singularities associated with $\boldsymbol{J}_0(\boldsymbol{q})$ are assumed to be avoided. Then the backstepping design procedure is stated as follows:

Step (1): Define the error variable $\boldsymbol{z}_1 = \boldsymbol{x} - \boldsymbol{x}_d$, where $\boldsymbol{x}_d$ is the desired inertially referenced trajectory and its time derivatives up to the third order are assumed to be continuously differentiable and bounded for all $t > 0$. The derivative of $\boldsymbol{z}_1$ from (2) is

$$\dot{\boldsymbol{z}}_1 = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_\mathrm{d} = \boldsymbol{J}_0\dot{\boldsymbol{q}} + \Xi_1 - \dot{\boldsymbol{x}}_\mathrm{d}$$

with the lump uncertainty Ξ_1 defined as $\Xi_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{J}\dot{\boldsymbol{q}} - \boldsymbol{J}_0\dot{\boldsymbol{q}}$, which will be estimated with a radial basis function (RBF) neural network.

From the neural network function approximation theory [36], there always exists an optimal RBF neural network that can approximate the unknown function Ξ_1 with arbitrary accuracy, that is, there exists an optimal weight matrix $\boldsymbol{W}_1$ and a residual $\boldsymbol{\varepsilon}_1$, such that $\Xi_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{W}_1^\mathrm{T}\boldsymbol{\Theta}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\varepsilon}_1$, where $\boldsymbol{\Theta}_1(\cdot)$ are the basis functions of the neural networks. In the following

DOI: 10.1002/rnc

deduction, the upper bounds of $\boldsymbol{W}_1$, $\boldsymbol{\varepsilon}_1$ are denoted as Δ_{W_1}, Δ_{ε_1}, that is, $\|\boldsymbol{W}_1\|_F \leqslant \Delta_{W_1}$, $\|\boldsymbol{\varepsilon}_1\| \leqslant \Delta_{\varepsilon_1}$.

Following the philosophy of the task-space inverse dynamics controller design [2], the virtual control vector for $\dot{\boldsymbol{q}}$ is designed as

$$\bar{\dot{\boldsymbol{q}}} = \boldsymbol{J}_0^{-1}\left(-k_1 z_1 + \dot{\boldsymbol{x}}_{\mathrm{d}} - \hat{\Xi}_1\right) \tag{4}$$

where k_1 is a positive constant to be determined and $\hat{\Xi}_1$ is the function approximation of Ξ_1 using RBF neural network.

The filtered virtual control input $\dot{\boldsymbol{q}}_f$ is obtained from the following time-varying first-order low-pass filter

$$\epsilon_2(t)\ddot{\boldsymbol{q}}_f + \dot{\boldsymbol{q}}_f = \bar{\dot{\boldsymbol{q}}}, \quad \dot{\boldsymbol{q}}_f(0) = \bar{\dot{\boldsymbol{q}}}(0) \tag{5}$$

where $\epsilon_2(t)$ will be designed in the following parts.

Step (2): Define the second error vector as $z_2 = \dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_f$, and the derivative of z_2 from (1) then can be written as

$$\dot{z}_2 = \ddot{\boldsymbol{q}} - \ddot{\boldsymbol{q}}_f = \boldsymbol{B}_0^{-1}(\boldsymbol{\tau} - \boldsymbol{V}_0) + \Xi_2 - \ddot{\boldsymbol{q}}_f$$

where $\Xi_2(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \ddot{\boldsymbol{q}} + \boldsymbol{B}_0^{-1}(\boldsymbol{V}_0 - \boldsymbol{B}\ddot{\boldsymbol{q}} - \boldsymbol{V})$ and will be approximated with RBF neural network, that is, $\Xi_2(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{W}_2{}^{\mathrm{T}}\boldsymbol{\Theta}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) + \boldsymbol{\varepsilon}_2$ with $\boldsymbol{W}_2$, $\boldsymbol{\varepsilon}_2$ similarly defined and $\|\boldsymbol{W}_2\|_F \leqslant \Delta_{W_2}$, $\|\boldsymbol{\varepsilon}_2\| \leqslant \Delta_{\varepsilon_2}$. Similar to the inverse dynamics control law design [2], the virtual control variable for $\boldsymbol{\tau}$ is designed to be

$$\bar{\boldsymbol{\tau}} = \boldsymbol{V}_0 + \boldsymbol{B}_0\left(-k_2 z_2 - \boldsymbol{J}_0{}^{\mathrm{T}} z_1 + \varrho_1 \boldsymbol{J}_0{}^{\mathrm{T}}\boldsymbol{J}_0 \boldsymbol{y}_2 - \varrho_1 \boldsymbol{J}_0{}^{\mathrm{T}}\hat{\boldsymbol{s}} - \hat{\Xi}_2 + \ddot{\boldsymbol{q}}_f\right) \tag{6}$$

where k_2, ϱ_1 are both positive constants to be determined; $\hat{\Xi}_2$ is the function approximation of Ξ_2 with an RBF neural network; $\boldsymbol{y}_2$ is the difference between the filtered and designed virtual control input of $\dot{\boldsymbol{q}}$, that is, $\boldsymbol{y}_2 = \dot{\boldsymbol{q}}_f - \bar{\dot{\boldsymbol{q}}}$; and the task-space sliding variable $\hat{\boldsymbol{s}}$ is defined as

$$\hat{\boldsymbol{s}} = \hat{\dot{\boldsymbol{x}}} - \dot{\boldsymbol{x}}_d + k_1(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{d}}) = \boldsymbol{J}_0\dot{\boldsymbol{q}} + \hat{\Xi}_1 - \dot{\boldsymbol{x}}_{\mathrm{d}} + k_1(\boldsymbol{x} - \boldsymbol{x}_d) \tag{7}$$

The filtered virtual control input $\boldsymbol{\tau}_f$ is obtained from the following time-varying first-order filter

$$\epsilon_3(t)\dot{\boldsymbol{\tau}}_f + \boldsymbol{\tau}_f = \bar{\boldsymbol{\tau}}, \quad \boldsymbol{\tau}_f(0) = \bar{\boldsymbol{\tau}}(0) \tag{8}$$

where the filter gain $\epsilon_3(t)$ will be designed in the following parts.

Step (3): Define the third error variable as $z_3 = \boldsymbol{\tau} - \boldsymbol{\tau}_f$. The derivative of z_3 from (3) is given by

$$\dot{z}_3 = \dot{\boldsymbol{\tau}} - \dot{\boldsymbol{\tau}}_f = \boldsymbol{k}_{\tau_0}\boldsymbol{L}_0^{-1}\left(\boldsymbol{u} - \boldsymbol{R}_0\boldsymbol{k}_{\tau_0}^{-1}\boldsymbol{\tau} - \boldsymbol{k}_{e0}\dot{\boldsymbol{q}}\right) + \Xi_3 - \dot{\boldsymbol{\tau}}_f$$

where $\Xi_3(\boldsymbol{\tau}, \dot{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{\tau}} + \boldsymbol{k}_{\tau_0}\boldsymbol{L}_0^{-1}\left(\boldsymbol{R}_0\boldsymbol{k}_{\tau_0}^{-1}\boldsymbol{\tau} + \boldsymbol{k}_{e0}\dot{\boldsymbol{q}} - \boldsymbol{L}\boldsymbol{k}_{\tau}^{-1}\dot{\boldsymbol{\tau}} - \boldsymbol{R}\boldsymbol{k}_{\tau}^{-1}\boldsymbol{\tau} - \boldsymbol{k}_e\dot{\boldsymbol{q}}\right)$ and will be approximated with an RBF neural network, that is, $\Xi_3(\boldsymbol{\tau}, \dot{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}) = \boldsymbol{W}_2^{\mathrm{T}}\boldsymbol{\Theta}(\boldsymbol{\tau}, \dot{\boldsymbol{\tau}}, \dot{\boldsymbol{q}}) + \boldsymbol{\varepsilon}_3$ with $\|\boldsymbol{W}_3\|_F \leqslant \Delta_{W_3}$, $\|\boldsymbol{\varepsilon}_3\| \leqslant \Delta_{\varepsilon_3}$.

The actual control signal $\boldsymbol{u}$ is finally designed as

$$\boldsymbol{u} = \boldsymbol{R}_0\boldsymbol{k}_{\tau_0}^{-1}\boldsymbol{\tau} + \boldsymbol{k}_{e0}\dot{\boldsymbol{q}} + \boldsymbol{L}_0\boldsymbol{k}_{\tau 0}^{-1}\left(-\hat{\Xi}_3 - k_3 z_3 - (\boldsymbol{B}_0{}^{-1})^{\mathrm{T}} z_2 + \dot{\boldsymbol{\tau}}_f\right) \tag{9}$$

with $\hat{\Xi}_3$ representing function approximation of Ξ_3 using RBF neural network, that is, $\hat{\Xi}_3 = \hat{\boldsymbol{W}}_3^{\mathrm{T}}\boldsymbol{\Theta}_3 + \hat{\boldsymbol{\varepsilon}}_3$.

In the following, the approximation errors of RBF neural networks are denoted as $\tilde{\Xi}_i \triangleq \Xi_i - \hat{\Xi}_i = (\boldsymbol{W}_i - \hat{\boldsymbol{W}}_i)^{\mathrm{T}}\boldsymbol{\Theta}_i + (\boldsymbol{\varepsilon}_i - \hat{\boldsymbol{\varepsilon}}_i) \triangleq \tilde{\boldsymbol{W}}_i^{\mathrm{T}}\boldsymbol{\Theta}_i + \tilde{\boldsymbol{\varepsilon}}_i, i = 1, 2, 3.$

DOI: 10.1002/rnc

3.2. Stability analysis

Define $\boldsymbol{y}_3$ as the differences between $\boldsymbol{\tau}_f$ and $\bar{\boldsymbol{\tau}}$, that is, $\boldsymbol{y}_3 = \boldsymbol{\tau}_f - \bar{\boldsymbol{\tau}}$. Define the variables $\boldsymbol{P}_2\left(z_1, z_2, \boldsymbol{y}_2, \boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d, \dot{\hat{\boldsymbol{W}}}_1\right) \triangleq -\dot{\bar{\boldsymbol{q}}}$, $\boldsymbol{P}_3\left(z_1, z_2, z_3, \boldsymbol{y}_2, \boldsymbol{y}_3, \boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d, \dot{\hat{\boldsymbol{W}}}_1, \dot{\hat{\boldsymbol{W}}}_2\right) \triangleq -\dot{\bar{\boldsymbol{\tau}}}$. Using $z_1, z_2, z_3, \boldsymbol{y}_2, \boldsymbol{y}_3$ as new system coordinates, with the designed control input (4), (6), and (9), the error dynamics for states z_1, z_2, z_3 can be expressed as

$$\begin{cases} \dot{z}_1 = -k_1 z_1 + \boldsymbol{J}_0 z_2 + \boldsymbol{J}_0 \boldsymbol{y}_2 + \tilde{\Xi}_1 \\ \dot{z}_2 = -\boldsymbol{J}_0^{\mathrm{T}} z_1 - k_2 z_2 + \boldsymbol{B}_0^{-1} z_3 + \varrho_1 \boldsymbol{J}_0{}^{\mathrm{T}} \boldsymbol{J}_0 \boldsymbol{y}_2 + \boldsymbol{B}_0{}^{-1} \boldsymbol{y}_3 - \varrho_1 \boldsymbol{J}_0^{\mathrm{T}} \hat{\boldsymbol{s}} + \tilde{\Xi}_2 \\ \dot{z}_3 = -(\boldsymbol{B}_0{}^{-1})^{\mathrm{T}} z_2 - k_3 z_3 + \tilde{\Xi}_3 \end{cases} \tag{10}$$

The dynamics for $\boldsymbol{y}_i$ $(i = 2, 3)$ can be formulated as

$$\begin{cases} \dot{\boldsymbol{y}}_2 = -\dfrac{\boldsymbol{y}_2}{\epsilon_2(t)} + \boldsymbol{P}_2 \\ \dot{\boldsymbol{y}}_3 = -\dfrac{\boldsymbol{y}_3}{\epsilon_3(t)} + \boldsymbol{P}_3 \end{cases} \tag{11}$$

Here, the following lemma is needed in the stability analysis and is stated in the following.

Lemma 1
There exist time-varying positive functions $\delta(t)$ converging to zero as $t \to \infty$ and satisfying

$$\lim_{t\to\infty} \int_0^t \delta(\omega) d\omega = \rho < \infty \ \text{ and } \ \dot{\delta}(t) = -l(t)\delta(t)$$

with finite constants ρ and functions $l(t) > 0$.

Remark 1
Note that there are many choices for $\delta(t)$ that satisfy Lemma 1. For example, $e^{-l_1 t}$ $(l_1 > 0)$, $(1+t)^{-l_2}$ $(l_2 > 1)$.

Define the bound for the nominal function of $\boldsymbol{J}_0$ and $\boldsymbol{B}_0^{-1}$ as Δ_{J_0} and $\Delta_{B_0^{-1}}$, that is, $\|\boldsymbol{J}_0\| \leqslant \Delta_{J_0}$, $\left\|\boldsymbol{B}_0{}^{-1}\right\| \leqslant \Delta_{B_0^{-1}}$. The main result of this paper is stated in the following theorem.

Theorem 1
For the robot manipulator system represented by (1), (2), and (3), if the control law is defined as (4), (6), and (9) and the parameter updating laws are chosen as

$$\begin{cases} \dot{\hat{\boldsymbol{W}}}_1 = \dfrac{1}{\beta_1} \boldsymbol{\Theta}_1 (z_1 + 2\varrho_1 \boldsymbol{s} - 2\varrho_1 \boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} - \dfrac{\alpha_1}{\beta_1} \|z_1\|^2 \hat{\boldsymbol{W}}_1 \\ \dot{\hat{\boldsymbol{\varepsilon}}}_1 = \dfrac{1}{\mu_1} (z_1 + 2\varrho_1 \boldsymbol{s} - 2\varrho_1 \boldsymbol{J}_0 \boldsymbol{y}_2) - \dfrac{\nu_1}{\mu_1} ||z_1||^2 \hat{\boldsymbol{\varepsilon}}_1 \end{cases} \tag{12}$$

$$\begin{cases} \dot{\hat{\boldsymbol{W}}}_i = \dfrac{1}{\beta_i} \boldsymbol{\Theta}_i z_i{}^{\mathrm{T}} - \dfrac{\alpha_i}{\beta_i} \|z_i\|^2 \hat{\boldsymbol{W}}_i \\ \dot{\hat{\boldsymbol{\varepsilon}}}_i = \dfrac{1}{\mu_i} z_i - \dfrac{\nu_i}{\mu_i} ||z_i||^2 \hat{\boldsymbol{\varepsilon}}_i \end{cases} \quad (i = 2, 3) \tag{13}$$

where $\boldsymbol{s} = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_{\mathrm{d}} + k_1(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{d}})$; $\alpha_i, \beta_i, \mu_i, \nu_i, \varrho_1$ $(i = 1, 2, 3)$ are arbitrary positive constants; the controller parameters are designed to satisfy

DOI: 10.1002/rnc

$$
\begin{aligned}
k_1 &> \frac{\Delta_{J_0}}{2} + \alpha_1 \frac{{\Delta_{W_1}}^2}{4} + \nu_1 \frac{{\Delta_{\varepsilon_1}}^2}{4} \\
k_2 &> \frac{\Delta_{B_0^{-1}}}{2} + \alpha_2 \frac{{\Delta_{W_2}}^2}{4} + \nu_2 \frac{{\Delta_{\varepsilon_2}}^2}{4} \\
k_3 &> \alpha_3 \frac{{\Delta_{W_3}}^2}{4} + \nu_3 \frac{{\Delta_{\varepsilon_3}}^2}{4}
\end{aligned} \tag{14}
$$

and the filter gains for (5), (8) are selected to be

$$
\epsilon_i(t) = \varrho_i \delta_i(t) \;\; i = 2, 3 \tag{15}
$$

where $\delta_i(t)$ are positive functions satisfying Lemma 1 with $\int_0^\infty \delta_i(t)\mathrm{d}t = \rho_i \;\; i = 2, 3$ and

$$
\begin{aligned}
0 < \varrho_2 &\leqslant \frac{2}{\delta_2(0) + \Delta_{J_0}} \\
0 < \varrho_3 &\leqslant \frac{2}{\delta_3(0) + \Delta_{B_0^{-1}}}
\end{aligned} \tag{16}
$$

then given any positive number p, for all initial conditions satisfying $\mathcal{S} : V_1(0) + V_2(0) \leqslant 2p$, with V_1, V_2 defined later, the following results for the closed-loop system can be obtained

(i) The task-space tracking error converges asymptotically to zero, that is, $\lim_{t\to\infty} \|z_1\| = 0$.
(ii) All the signals in the closed-loop system are bounded.
(iii) The following $\mathcal{L}_2$ transient tracking performance bounds can be guaranteed.

$$
\|\boldsymbol{x} - \boldsymbol{x}_d\|_{\mathcal{L}_2} \leqslant \left(\kappa_p V(0) + \frac{\rho_2 {\Delta_{P_2}}^2 + \rho_3 {\Delta_{P_3}}^2}{2} \right)^{1/2} \tag{17}
$$

$$
\|\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d\|_{\mathcal{L}_2} \leqslant \left(\kappa_d V(0) + \left(4\Delta_{J_0}^2 + 2{k_1}^2 + 2 \right) \left(\rho_2 {\Delta_{P_2}}^2 + \rho_3 {\Delta_{P_3}}^2 \right) \right)^{1/2} \tag{18}
$$

with $\kappa_p = \frac{1}{\eta_1}, \kappa_d = \frac{4{k_1}^2}{\eta_1} + \frac{4\Delta_{J_0}^2}{\eta_2} + \frac{4\Delta_{J_0}^2}{\eta_4} + \frac{4}{\varrho_1}$.

Proof

(i) : Define candidate Lyapunov functions

$$
V_1 = \frac{1}{2}\sum_{i=1}^{3} \left[z_i^{\mathrm{T}} z_i + \beta_i \mathrm{tr}\left\{ \tilde{\boldsymbol{W}}_i^{\mathrm{T}} \tilde{\boldsymbol{W}}_i \right\} + \mu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \tilde{\boldsymbol{\varepsilon}}_i \right], \quad V_2 = \frac{1}{2}\sum_{i=2}^{3} \delta_i(t) \boldsymbol{y}_i^{\mathrm{T}} \boldsymbol{y}_i
$$

The derivatives of V_1 along system (10) is

$$
\begin{aligned}
\dot{V}_1 = &\sum_{i=1}^{3} \left[z_i{}^{\mathrm{T}} \tilde{\Xi}_i - \beta_i \mathrm{tr}\left\{ \tilde{\boldsymbol{W}}_i^{\mathrm{T}} \dot{\hat{\boldsymbol{W}}}_i \right\} - \mu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \dot{\hat{\boldsymbol{\varepsilon}}}_i \right] + z_1{}^{\mathrm{T}} \left(-k_1 z_1 + \boldsymbol{J}_0 z_2 + \boldsymbol{J}_0 \boldsymbol{y}_2 \right) \\
&+ z_2{}^{\mathrm{T}} \left(-\boldsymbol{J}_0{}^{\mathrm{T}} z_1 - k_2 z_2 + \boldsymbol{B}_0^{-1} z_3 + \varrho_1 \boldsymbol{J}_0{}^{\mathrm{T}} \boldsymbol{J}_0 \boldsymbol{y}_2 + \boldsymbol{B}_0{}^{-1} \boldsymbol{y}_3 - \varrho_1 \boldsymbol{J}_0{}^{\mathrm{T}} \hat{\boldsymbol{s}} \right) \\
&+ z_3{}^{\mathrm{T}} \left(-\left(\boldsymbol{B}_0{}^{-1} \right)^{\mathrm{T}} z_2 - k_3 z_3 \right) \\
= &\sum_{i=1}^{3} \left[\tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \left(z_i - \mu_i \dot{\hat{\varepsilon}}_i \right) + \mathrm{tr}\left\{ \tilde{\boldsymbol{W}}_i^{\mathrm{T}} \boldsymbol{\Theta}_i z_i{}^{\mathrm{T}} - \beta_i \tilde{\boldsymbol{W}}_i^{\mathrm{T}} \dot{\hat{\boldsymbol{W}}}_i \right\} \right] + \Xi
\end{aligned}
$$

with $\Xi = z_1{}^{\mathrm{T}} \left(-k_1 z_1 + \boldsymbol{J}_0 \boldsymbol{y}_2 \right) + z_2{}^{\mathrm{T}} \left(-k_2 z_2 + \varrho_1 \boldsymbol{J}_0{}^{\mathrm{T}} \boldsymbol{J}_0 \boldsymbol{y}_2 + \boldsymbol{B}_0{}^{-1} \boldsymbol{y}_3 - \varrho_1 \boldsymbol{J}_0{}^{\mathrm{T}} \hat{\boldsymbol{s}} \right) - k_3 z_3{}^{\mathrm{T}} z_3$.

DOI: 10.1002/rnc

If the parameter updating law is chosen according to (13), $\dot{V}_1$ is

$$\dot{V}_1 = \sum_{i=2}^{3} \|z_i\|^2 \left[\alpha_i \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \hat{\boldsymbol{W}}_i\right\} + \nu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \hat{\boldsymbol{\varepsilon}}_i\right] + \left[z_1{}^{\mathrm{T}} \tilde{\Xi}_1 - \beta_1 \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}} \dot{\hat{\boldsymbol{W}}}_1\right\} - \mu_1 \tilde{\boldsymbol{\varepsilon}}_1{}^{\mathrm{T}} \dot{\hat{\boldsymbol{\varepsilon}}}_1\right] + \Xi \tag{19}$$

Because $\boldsymbol{J}_0 z_2 = \boldsymbol{J}_0 \left(\dot{\boldsymbol{q}} - \boldsymbol{y}_2 - \dot{\bar{\boldsymbol{q}}}\right) = \hat{\boldsymbol{s}} - \boldsymbol{J}_0 \boldsymbol{y}_2$, one has

$$\begin{aligned}\dot{V}_1 = &\sum_{i=2}^{3} \|z_i\|^2 \left[\alpha_i \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \hat{\boldsymbol{W}}_i\right\} + \nu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \hat{\boldsymbol{\varepsilon}}_i\right] + \left[z_1{}^{\mathrm{T}} \tilde{\Xi}_1 - \beta_1 \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}} \dot{\hat{\boldsymbol{W}}}_1\right\} - \mu_1 \tilde{\boldsymbol{\varepsilon}}_1{}^{\mathrm{T}} \dot{\hat{\boldsymbol{\varepsilon}}}_1\right] - k_3 z_3{}^{\mathrm{T}} z_3 \\ &+ z_1{}^{\mathrm{T}} \left(-k_1 z_1 + \boldsymbol{J}_0 \boldsymbol{y}_2\right) + z_2{}^{\mathrm{T}} \left(-k_2 z_2 + \boldsymbol{B}_0{}^{-1} \boldsymbol{y}_3\right) - \varrho_1 (\hat{\boldsymbol{s}} - \boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} (\hat{\boldsymbol{s}} - \boldsymbol{J}_0 \boldsymbol{y}_2)\end{aligned}$$

Because $\boldsymbol{s} - \hat{\boldsymbol{s}} = \tilde{\Xi}_1$, it can be obtained that

$$\begin{aligned}\dot{V}_1 = &\sum_{i=2}^{3} \|z_i\|^2 \left[\alpha_i \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \hat{\boldsymbol{W}}_i\right\} + \nu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \hat{\boldsymbol{\varepsilon}}_i\right] + \left[z_1{}^{\mathrm{T}} \tilde{\Xi}_1 - \beta_1 \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}} \dot{\hat{\boldsymbol{W}}}_1\right\} - \mu_1 \tilde{\boldsymbol{\varepsilon}}_1{}^{\mathrm{T}} \dot{\hat{\boldsymbol{\varepsilon}}}_1\right] \\ &+ z_1{}^{\mathrm{T}} \left(-k_1 z_1 + \boldsymbol{J}_0 \boldsymbol{y}_2\right) + z_2{}^{\mathrm{T}} \left(-k_2 z_2 + \boldsymbol{B}_0{}^{-1} \boldsymbol{y}_3\right) - k_3 z_3{}^{\mathrm{T}} z_3 \\ &- \varrho_1 (\boldsymbol{s} - \boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} (\boldsymbol{s} - \boldsymbol{J}_0 \boldsymbol{y}_2) - \varrho_1 \tilde{\Xi}_1^{\mathrm{T}} \tilde{\Xi}_1 + 2\varrho_1 (\boldsymbol{s} - \boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} \tilde{\Xi}_1 \\ = &\sum_{i=2}^{3} \|z_i\|^2 \left[\alpha_i \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \hat{\boldsymbol{W}}_i\right\} + \nu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \hat{\boldsymbol{\varepsilon}}_i\right] - \left[\beta_1 \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}} \dot{\hat{\boldsymbol{W}}}_1\right\} + \mu_1 \tilde{\boldsymbol{\varepsilon}}_1{}^{\mathrm{T}} \dot{\hat{\boldsymbol{\varepsilon}}}_1\right] \\ &+ z_1{}^{\mathrm{T}} \left(-k_1 z_1 + \boldsymbol{J}_0 \boldsymbol{y}_2\right) \quad + z_2{}^{\mathrm{T}} \left(-k_2 z_2 + \boldsymbol{B}_0{}^{-1} \boldsymbol{y}_3\right) - k_3 z_3{}^{\mathrm{T}} z_3 \\ &- \varrho_1 (\boldsymbol{s} - \boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} (\boldsymbol{s} - \boldsymbol{J}_0 \boldsymbol{y}_2) - \varrho_1 \tilde{\Xi}_1^{\mathrm{T}} \tilde{\Xi}_1 + (z_1 + 2\varrho_1 \boldsymbol{s} - 2\varrho_1 \boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} \tilde{\Xi}_1 \\ \leq &\sum_{i=2}^{3} \|z_i\|^2 \left[\alpha_i \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \hat{\boldsymbol{W}}_i\right\} + \nu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \hat{\boldsymbol{\varepsilon}}_i\right] - \left[\beta_1 \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}} \dot{\hat{\boldsymbol{W}}}_1\right\} + \mu_1 \tilde{\boldsymbol{\varepsilon}}_1{}^{\mathrm{T}} \dot{\hat{\boldsymbol{\varepsilon}}}_1\right] \\ &+ z_1{}^{\mathrm{T}} \left(-k_1 z_1 + \boldsymbol{J}_0 \boldsymbol{y}_2\right) \quad + z_2{}^{\mathrm{T}} \left(-k_2 z_2 + \boldsymbol{B}_0{}^{-1} \boldsymbol{y}_3\right) - k_3 z_3{}^{\mathrm{T}} z_3 \\ &- \varrho_1 \tilde{\Xi}_1^{\mathrm{T}} \tilde{\Xi}_1 + (z_1 + 2\varrho_1 \boldsymbol{s} - 2\varrho_1 \boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} \tilde{\Xi}_1\end{aligned}$$

If the parameter estimation law is chosen as (12), the derivative of V_1 is

$$\begin{aligned}\dot{V}_1 \leq &- \varrho_1 \tilde{\Xi}_1^{\mathrm{T}} \tilde{\Xi}_1 + \sum_{i=1}^{3} \|z_i\|^2 \left[\alpha_i \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \hat{\boldsymbol{W}}_i\right\} + \nu_i \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} \hat{\boldsymbol{\varepsilon}}_i\right] - \left(k_1 - \frac{\Delta_{J_0}}{2}\right) z_1{}^{\mathrm{T}} z_1 \\ &- \left(k_2 - \frac{\Delta_{B_0{}^{-1}}}{2}\right) z_2{}^{\mathrm{T}} z_2 - k_3 z_3{}^{\mathrm{T}} z_3 + \frac{\Delta_{J_0}}{2} \boldsymbol{y}_2{}^{\mathrm{T}} \boldsymbol{y}_2 + \frac{\Delta_{B_0{}^{-1}}}{2} \boldsymbol{y}_3{}^{\mathrm{T}} \boldsymbol{y}_3\end{aligned}$$

In view of properties of the Frobenius norm [37], one has

$$\mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \hat{\boldsymbol{W}}_i\right\} = \mathrm{tr}\left\{\tilde{\boldsymbol{W}}_i^{\mathrm{T}} \boldsymbol{W}_i\right\} - \left\|\tilde{\boldsymbol{W}}_i\right\|_F{}^2 \leq \left\|\tilde{\boldsymbol{W}}_i\right\|_F \Delta_{W_i} - \left\|\tilde{\boldsymbol{W}}_i\right\|_F \tag{20}$$

Because $\tilde{\boldsymbol{\varepsilon}}_i{}^T \hat{\boldsymbol{\varepsilon}}_i = \tilde{\boldsymbol{\varepsilon}}_i{}^{\mathrm{T}} (\boldsymbol{\varepsilon}_i - \tilde{\boldsymbol{\varepsilon}}_i) \leq \|\tilde{\boldsymbol{\varepsilon}}_i\| \Delta_{\varepsilon_i} - \|\tilde{\boldsymbol{\varepsilon}}_i\|^2$, one obtains

$$\begin{aligned}\dot{V}_1 \leq &\sum_{i=1}^{3} \alpha_i \|z_i\|^2 \left[\frac{\Delta_{W_i}{}^2}{4} - \left(\left\|\tilde{\boldsymbol{W}}\right\|_F - \frac{\Delta_{W_i}}{2}\right)^2\right] + \sum_{i=1}^{3} \nu_i ||z_i||^2 \left[\frac{\Delta_{\varepsilon_i}{}^2}{4} - \left(||\tilde{\boldsymbol{\varepsilon}}_i|| - \frac{\Delta_{\varepsilon_i}}{2}\right)^2\right] \\ &- \varrho_1 \tilde{\Xi}_1^{\mathrm{T}} \tilde{\Xi}_1 - \left(k_1 - \frac{\Delta_{J_0}}{2}\right) z_1{}^{\mathrm{T}} z_1 - \left(k_2 - \frac{\Delta_{B_0{}^{-1}}}{2}\right) z_2{}^{\mathrm{T}} z_2 - k_3 z_3{}^{\mathrm{T}} z_3\end{aligned}$$

DOI: 10.1002/rnc

$$
\begin{aligned}
&+\frac{\Delta_{J_0}}{2}{\boldsymbol{y}_2}^{\mathrm{T}}\boldsymbol{y}_2+\frac{\Delta_{B_0^{-1}}}{2}{\boldsymbol{y}_3}^{\mathrm{T}}\boldsymbol{y}_3\\
\leqslant&-\varrho_1\tilde{\Xi}_1^{\mathrm{T}}\tilde{\Xi}_1-\left(k_1-\frac{\Delta_{J_0}}{2}-\alpha_1\frac{{\Delta_{W_1}}^2}{4}-\nu_1\frac{{\Delta_{\varepsilon_1}}^2}{4}\right){z_1}^{\mathrm{T}}z_1\\
&-\left(k_2-\frac{\Delta_{B_0^{-1}}}{2}-\alpha_2\frac{{\Delta_{W_2}}^2}{4}-\nu_2\frac{{\Delta_{\varepsilon_2}}^2}{4}\right){z_2}^{\mathrm{T}}z_2+\frac{\Delta_{J_0}}{2}{\boldsymbol{y}_2}^{\mathrm{T}}\boldsymbol{y}_2\\
&-\left(k_3-\alpha_3\frac{{\Delta_{W_3}}^2}{4}-\nu_3\frac{{\Delta_{\varepsilon_3}}^2}{4}\right){z_3}^{\mathrm{T}}z_3+\frac{\Delta_{B_0^{-1}}}{2}{\boldsymbol{y}_3}^{\mathrm{T}}\boldsymbol{y}_3
\end{aligned}
$$

Also, the derivative of V_2 can be calculated as

$$
\begin{aligned}
\dot{V}_2=&-\frac{l_2(t)\delta_2(t)}{2}{\boldsymbol{y}_2}^{\mathrm{T}}\boldsymbol{y}_2-\frac{l_3(t)\delta_3(t)}{2}{\boldsymbol{y}_3}^{\mathrm{T}}\boldsymbol{y}_3\\
&+\delta_2(t){\boldsymbol{y}_2}^{\mathrm{T}}\left[-\frac{\boldsymbol{y}_2}{\epsilon_2(t)}+\boldsymbol{P}_2\right]+\delta_2(t){\boldsymbol{y}_3}^{\mathrm{T}}\left[-\frac{\boldsymbol{y}_3}{\epsilon_3(t)}+\boldsymbol{P}_3\right]
\end{aligned}
$$

Because for any $p>0$, the set $\mathcal{S}$ is compact. Therefore, $\|\boldsymbol{P}_i\|$ has a maximum Δ_{P_i} on $\mathcal{S}$. Thus, it can be concluded that

$$
\begin{aligned}
\dot{V}_2\leqslant&-\left(\frac{l_2(t)\delta_2(t)}{2}+\frac{\delta_2(t)}{\epsilon_2(t)}\right){\boldsymbol{y}_2}^{\mathrm{T}}\boldsymbol{y}_2+\Delta_{P_2}\delta_2(t)\,\|\boldsymbol{y}_2\|\\
&-\left(\frac{l_3(t)\delta_3(t)}{2}+\frac{\delta_3(t)}{\epsilon_3(t)}\right){\boldsymbol{y}_3}^{\mathrm{T}}\boldsymbol{y}_3+\Delta_{P_3}\delta_3(t)\,\|\boldsymbol{y}_3\|\\
\leqslant&-\left(\frac{l_2(t)\delta_2(t)}{2}+\frac{\delta_2(t)}{\epsilon_2(t)}-\frac{\delta_2(t)}{2}\right){\boldsymbol{y}_2}^{\mathrm{T}}\boldsymbol{y}_2\\
&-\left(\frac{l_3(t)\delta_3(t)}{2}+\frac{\delta_3(t)}{\epsilon_3(t)}-\frac{\delta_3(t)}{2}\right){\boldsymbol{y}_3}^{\mathrm{T}}\boldsymbol{y}_3+\frac{\delta_2(t)\Delta_{P_2}^2+\delta_3(t)\Delta_{P_3}^2}{2}
\end{aligned}
$$

Define the candidate Lyapunov function $V=V_1+V_2$ for the entire closed-loop system, one obtains

$$
\begin{aligned}
\dot{V}\leqslant&-\varrho_1\tilde{\Xi}_1^{\mathrm{T}}\tilde{\Xi}_1-\left(k_1-\frac{\Delta_{J_0}}{2}-\alpha_1\frac{{\Delta_{W_1}}^2}{4}-\nu_1\frac{{\Delta_{\varepsilon_1}}^2}{4}\right){z_1}^{\mathrm{T}}z_1+\frac{\delta_2(\mathrm{t})\Delta_{P_2}+\delta_3^2(\mathrm{t})\Delta_{P_3}^2}{2}\\
&-\left(k_2-\frac{\Delta_{B_0^{-1}}}{2}-\alpha_2\frac{{\Delta_{W_2}}^2}{4}-\nu_2\frac{{\Delta_{\varepsilon_2}}^2}{4}\right){z_2}^{\mathrm{T}}z_2-\left(k_3-\alpha_3\frac{{\Delta_{W_3}}^2}{4}-\nu_3\frac{{\Delta_{\varepsilon_3}}^2}{4}\right){z_3}^{\mathrm{T}}z_3\\
&-\left(\frac{l_2(t)\delta_2(t)}{2}+\frac{\delta_2(t)}{\epsilon_2(t)}-\frac{\delta_2(t)}{2}-\frac{\Delta_{J_0}}{2}\right){\boldsymbol{y}_2}^{\mathrm{T}}\boldsymbol{y}_2-\left(\frac{l_3(t)\delta_3(t)}{2}+\frac{\delta_3(t)}{\epsilon_3(t)}-\frac{\delta_3(t)}{2}-\frac{\Delta_{B_0^{-1}}}{2}\right){\boldsymbol{y}_3}^{\mathrm{T}}\boldsymbol{y}_3
\end{aligned}
$$

If the parameter of the first-order low-pass filter is chosen as (15), then

$$
\begin{aligned}
\dot{V}\leqslant&-\varrho_1\tilde{\Xi}_1^{\mathrm{T}}\tilde{\Xi}_1-\left(k_1-\frac{\Delta_{J_0}}{2}-\alpha_1\frac{{\Delta_{W_1}}^2}{4}-\nu_1\frac{{\Delta_{\varepsilon_1}}^2}{4}\right){z_1}^{\mathrm{T}}z_1+\frac{\delta_2(\mathrm{t}){\Delta_{P_2}}^2+\delta_3(\mathrm{t}){\Delta_{P_3}}^2}{2}\\
&-\left(k_2-\frac{\Delta_{B_0^{-1}}}{2}-\alpha_2\frac{{\Delta_{W_2}}^2}{4}-\nu_2\frac{{\Delta_{\varepsilon_2}}^2}{4}\right){z_2}^{\mathrm{T}}z_2-\left(k_3-\alpha_3\frac{{\Delta_{W_3}}^2}{4}-\nu_3\frac{{\Delta_{\varepsilon_3}}^2}{4}\right){z_3}^{\mathrm{T}}z_3\\
&-\left(\frac{1}{\varrho_2}-\frac{\delta_2(0)}{2}-\frac{\Delta_{J_0}}{2}\right){\boldsymbol{y}_2}^{\mathrm{T}}\boldsymbol{y}_2-\left(\frac{1}{\varrho_3}-\frac{\delta_3(0)}{2}-\frac{\Delta_{B_0^{-1}}}{2}\right){\boldsymbol{y}_3}^{\mathrm{T}}\boldsymbol{y}_3
\end{aligned}
\tag{21}
$$

DOI: 10.1002/rnc

If controller parameter are chosen to satisfy (14), (16) and further define

$$\eta_1 = k_1 - \frac{\Delta_{J_0}}{2} - \alpha_1 \frac{\Delta_{W_1}{}^2}{4} - \nu_1 \frac{\Delta_{\varepsilon_1}{}^2}{4}, \quad \eta_2 = k_2 - \frac{\Delta_{B_0{}^{-1}}}{2} - \alpha_2 \frac{\Delta_{W_2}{}^2}{4} - \nu_2 \frac{\Delta_{\varepsilon_2}{}^2}{4}$$
$$\eta_3 = k_3 - \alpha_3 \frac{\Delta_{W_3}{}^2}{4} - \nu_3 \frac{\Delta_{\varepsilon_3}{}^2}{4}, \quad \eta_4 = \frac{1}{\varrho_2} - \frac{\delta_2(0)}{2} - \frac{\Delta_{J_0}}{2}, \quad \eta_5 = \frac{1}{\varrho_3} - \frac{\delta_3(0)}{2} - \frac{\Delta_{B_0{}^{-1}}}{2}$$
$$\eta_6 = \varrho_1, \quad \eta_m = \min\{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6\}$$

then (21) can be summarized as

$$\dot{V} \leqslant -\eta_m \|\mathcal{X}\|^2 + \frac{\delta_2(t)\Delta_{P_2}{}^2 + \delta_3(t)\Delta_{P_3}{}^2}{2} \tag{22}$$

where $\mathcal{X} = \left[\tilde{\Xi}_1^{\mathrm{T}}, z_1^{\mathrm{T}}, z_2{}^{\mathrm{T}}, z_3{}^{\mathrm{T}}, \boldsymbol{y}_2{}^{\mathrm{T}}, \boldsymbol{y}_3{}^{\mathrm{T}}\right]^{\mathrm{T}}$. Integrating both sides over $[0, \infty]$ yields

$$\begin{aligned} rl \int_0^\infty \|\mathcal{X}\|^2 \mathrm{d}t &\leqslant \frac{\Delta_{P_2}{}^2}{2\eta_m} \int_0^\infty \delta_2(t)\mathrm{d}t + \frac{\Delta_{P_3}{}^2}{2\eta_m} \int_0^\infty \delta_3(t)\mathrm{d}t - \frac{(V(\infty) - V(0))}{\eta_m} \\ &\leqslant \frac{\rho_2 \Delta_{P_2}{}^2 + \rho_3 \Delta_{P_3}{}^2 + 2V(0)}{2\eta_m} < \infty \end{aligned} \tag{23}$$

which means $\mathcal{X} \in \mathcal{L}_2$; thus, $\mathcal{X}$ will eventually converges to zero, which equally implies the asymptotic convergence of the tracking error.

(ii) : Integrating both sides of (22) over $[0, t]$ gives

$$V(t) - V(0) \leqslant -\eta_m \int_0^t \|\mathcal{X}\|^2 \mathrm{d}\tau + \frac{\rho_2 \Delta_{P_2}{}^2 + \rho_3 \Delta_{P_3}{}^2}{2} < \infty$$

Thus, $V(t)$ is bounded, which implies that $z_i, \hat{\boldsymbol{W}}_i, \hat{\boldsymbol{\varepsilon}}_i, \boldsymbol{y}_j \in \mathcal{L}_\infty \ \ (i = 1, 2, 3; \ j = 2, 3)$. From (9), one can further conclude that the control signal is also bounded.

(iii) : From (21), it gives

$$\int_0^\infty z_1(\tau)^{\mathrm{T}} z_1(\tau)\mathrm{d}\tau \leqslant \frac{1}{\eta_1}(V(0) - V(\infty)) + \frac{\rho_2 \Delta_{P_2}{}^2 + \rho_3 \Delta_{P_3}{}^2}{2} \tag{24}$$

Because V is positive, one has

$$\int_0^\infty z_1(\tau)^{\mathrm{T}} z_1(\tau)\mathrm{d}\tau \leqslant \frac{1}{\eta_1} V(0) + \frac{\rho_2 \Delta_{P_2}{}^2 + \rho_3 \Delta_{P_3}{}^2}{2}$$

In view of the definition of z_1 and the definition of $\mathcal{L}_2$-norm, (17) can be obtained. By simple calculation, it can be found that $\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d = \boldsymbol{J}\dot{\boldsymbol{q}} - \dot{\boldsymbol{x}}_d = \boldsymbol{J}_0 z_2 + \boldsymbol{J}_0 \boldsymbol{y}_2 - k_1 z_1 + \tilde{\Xi}_1$, which means that

$$\begin{aligned} \int_0^\infty (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d)^{\mathrm{T}} (\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d)\mathrm{d}t &= \int_0^\infty (\boldsymbol{J}_0 z_2 + \boldsymbol{J}_0 \boldsymbol{y}_2 - k_1 z_1 + \tilde{\Xi}_1)^{\mathrm{T}} (\boldsymbol{J}_0 z_2 + \boldsymbol{J}_0 \boldsymbol{y}_2 - k_1 z_1 + \tilde{\Xi}_1)\mathrm{d}t \\ &\leqslant 4 \int_0^\infty \left((\boldsymbol{J}_0 z_2)^{\mathrm{T}} (\boldsymbol{J}_0 z_2) + (\boldsymbol{J}_0 \boldsymbol{y}_2)^{\mathrm{T}} (\boldsymbol{J}_0 \boldsymbol{y}_2) + k_1{}^2 z_1{}^{\mathrm{T}} z_1 + (\tilde{\Xi}_1)^{\mathrm{T}} (\tilde{\Xi}_1) \right) \mathrm{d}t \end{aligned} \tag{25}$$

DOI: 10.1002/rnc

Similar to (24), the following results can be obtained from (21)

$$\begin{aligned}
\int_0^\infty {z_2}^{\mathrm{T}} z_2 \mathrm{d}t &\leqslant \frac{1}{\eta_2} V(0) + \frac{\rho_2 {\Delta_{P_2}}^2 + \rho_3 {\Delta_{P_3}}^2}{2} \\
\int_0^\infty {\boldsymbol{y}_2}^{\mathrm{T}} \boldsymbol{y}_2 \mathrm{d}t &\leqslant \frac{1}{\eta_4} V(0) + \frac{\rho_2 {\Delta_{P_2}}^2 + \rho_3 {\Delta_{P_3}}^2}{2} \\
\int_0^\infty {\tilde{\Xi}_1}^{\mathrm{T}} \tilde{\Xi}_1 \mathrm{d}t &\leqslant \frac{1}{\varrho_1} V(0) + \frac{\rho_2 {\Delta_{P_2}}^2 + \rho_3 {\Delta_{P_3}}^2}{2}
\end{aligned} \tag{26}$$

Because $\boldsymbol{J}_0$ is assumed to be upper-bounded by Δ_{J_0}, combining this fact with (25) and (26), (18) can be obtained.

□

Remark 2

The implementation of the proposed controller requires full state information of $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, $\boldsymbol{\tau}$, and $\dot{\boldsymbol{\tau}}$. In real applications, $\boldsymbol{x}, \dot{\boldsymbol{x}}$ can be obtained by a global camera or an inertial measurement unit mounted to the end-effector. $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{\tau}$ and $\dot{\boldsymbol{\tau}}$ can be obtained by angle and torque sensors placed at the robot manipulator joints.

Remark 3

The closed-loop system (11) is in the form of the so-called standard singular perturbation model.

$$\begin{aligned}
\epsilon_2(t)\dot{\boldsymbol{y}}_2 &= -\boldsymbol{y}_2 + \epsilon_2(t)\boldsymbol{P}_2 \\
\epsilon_3(t)\dot{\boldsymbol{y}}_3 &= -\boldsymbol{y}_3 + \epsilon_3(t)\boldsymbol{P}_3
\end{aligned} \tag{27}$$

the state y_i of the fast varying system (27), whose velocity $\dot{y}_i$ can be large when ϵ_i is small, may rapidly converge to a root of $0 = -y_i - \epsilon_i(t)P_i$, which tends to zero. In this situation, the closed-loop system dynamics of (10) is determined by the following quasi-steady-state model

$$\begin{cases}
\dot{z}_1 = -k_1 z_1 + \boldsymbol{J}_0 z_2 + \tilde{\Xi}_1 \\
\dot{z}_2 = -{\boldsymbol{J}_0}^{\mathrm{T}} z_1 - k_2 z_2 + \boldsymbol{B}_0^{-1} z_3 - \varrho_1 {\boldsymbol{J}_0}^{\mathrm{T}} \hat{\boldsymbol{s}} + \tilde{\Xi}_2 \\
\dot{z}_3 = -({\boldsymbol{B}_0}^{-1})^{\mathrm{T}} z_2 - k_3 z_3 + \tilde{\Xi}_3
\end{cases} \tag{28}$$

As can be seen, system (28) can be rendered stable by appropriately choosing the parameters k_1, k_2, k_3. This singular perturbation model also implies that reducing ϵ_i diminishes the effect of y_2, y_3 in (28), which tells us that the introduction of the low-pass filters does not significantly affect the stability property of the system (10).

Remark 4

Compared with conventional adaptive dynamic surface control methods cited in the introduction part, the design of the time-varying first-order low-pass filter renders the tracking error asymptotically converges to zero as time evolves to infinity. When referred to conventional dynamic surface control, the designed function $\epsilon_i(t)$ could be regarded as a specific real constant number. However, as to the proposed controller in this paper, $\epsilon_i(t)$ changes to a time-varying function decreasing quickly towards zero, which essentially means the expanding of the bandwidth of the first-order low-pass filter. Specially, when time evolves to infinity, the low-pass filter will change to a full-pass filter, which is actually a direct-pass block. Then, the designed dynamic surface control degrades to the backstepping control with the replacement of the symbolic derivative calculation to a numerical one. It can be observed from (11) that a small ϵ_i would result in a large $\dot{y}_i$ and subsequently a large y_i. The computation of large numbers would require a large memory space and pose computational burdens on the controller. However, from (22), it is known that when $\|\mathcal{X}\|^2 > \frac{\delta_2(t){\Delta_{P_2}}^2 + \delta_3(t){\Delta_{P_3}}^2}{2\eta_m}$,

DOI: 10.1002/rnc

$\dot{V} < 0$. Thus, V will decrease until $\|\mathcal{X}\|^2 \leqslant \frac{\delta_2(t)\Delta_{P_2}{}^2+\delta_3(t)\Delta_{P_3}{}^2}{2\eta_m}$. From this perspective, a small δ_i is preferred. Thus, one can always make a compromise between the computational burden and the control performance by deciding when to stop changing the gains of the designed first-order low-pass filters.

4. PASSIVITY INTERPRETATION

The robustifying adaptation laws designed in (12) and (13) for the neural network-based controller are somewhat like the e_1 modification [38] but not exactly. In e_1 modification, the second terms of the right side of (12), (13) do not contain $\|z_i\|^2$. In this section, the physical meaning of the term $\|z_i\|^2$ is interpreted through the passivity theory. It is shown that the existence of $\|z_i\|^2$ renders the adaptive dynamic subsystem input feedforward passive (IFP). Further, combining the output feedback passive (OFP) property of the transformed plant, the global system is output strict passive. With this modification, the system energy is successfully reshaped to its desired equilibrium. In the following, basic concepts of passivity and dissipativity are introduced first.

Definition 1
Dissipativity [39]: Assume that associated with the system $\boldsymbol{H}$ is a function $\boldsymbol{\omega} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, called the supply rate, which is locally integrable for every $\boldsymbol{u} \in \boldsymbol{U}$. That is, it satisfies $\int_{t_0}^{t_1} |\boldsymbol{\omega}(\boldsymbol{u}(t), \boldsymbol{y}(t))|\mathrm{d}t < \infty$ for all $t_0 \leqslant t_1$. Let $\boldsymbol{X}$ be a connected subset of $\mathbb{R}^n$ containing the origin. We say that the system $\boldsymbol{H}$ is dissipative in $\boldsymbol{X}$ with the supply rate $\boldsymbol{\omega}(\boldsymbol{u}, \boldsymbol{y})$ if there exists a function $S(\boldsymbol{x})$, $S(\boldsymbol{0}) = 0$, such that for all $\boldsymbol{x} \in \boldsymbol{X}$

$$S(\boldsymbol{x}) \geqslant 0 \ \text{ and } \ S(\boldsymbol{x}(T)) - S(\boldsymbol{x}(0)) \leqslant \int_0^T \omega(\boldsymbol{u}(t), \boldsymbol{y}(t))\mathrm{d}t$$

for all $\boldsymbol{u} \in \boldsymbol{U}$ and all $T \geqslant 0$ such that $\boldsymbol{x}(t) \in \boldsymbol{X}$ for all $t \in [0, T]$. The function $S(\boldsymbol{x})$ is then called a storage function.

Definition 2
Passivity: System $\boldsymbol{H}$ is said to be passive if it is dissipative with a supply rate $\boldsymbol{\omega}(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{y}$.

Definition 3
Excess/shortage of passivity: System $\boldsymbol{H}$ is said to be

- Output feedback passive: if it is dissipative with respect to $\boldsymbol{\omega}(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{y} - \pi\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}$ for some $\pi \in \mathbb{R}$. In the following, this property is quantified with the notation OFP(π).
- Output strict passive: if it is OFP with $\pi > 0$.
- Input feedforward passive: if it is dissipative with respect to $\boldsymbol{\omega}(\boldsymbol{u}, \boldsymbol{y}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{y} - \sigma\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}$ for some $\sigma \in \mathbb{R}$. In the following, this property is quantified with the notation IFP(σ).
- Input strict passive: if it is IFP with $\sigma > 0$.

For simplicity, only the dynamics z_1 in (10) and the adaptation law for matrix gain $\hat{\boldsymbol{W}}_1$ are analyzed. It is assumed that $\tilde{\varepsilon}_1, z_2, \boldsymbol{y}_2$ all equal to zero. Then the closed-loop system can be rewritten into the following connected system as shown in Figure 1.

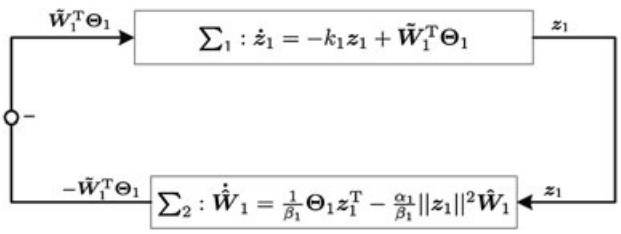

Figure 1. Interconnections of designed dynamic subsystems.

DOI: 10.1002/rnc

For $\sum_1$ subsystem denoted in Figure 1, define the input as $\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\boldsymbol{\Theta}_1$ and the output as z_1, then the energies injected into this subsystem can be regarded as $\int_0^t\left(\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\boldsymbol{\Theta}_1\right)^{\mathrm{T}}z_1\mathrm{d}t$, and by simple calculation, one can find that

$$\int_0^t\left(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Theta}_1\right)^{\mathrm{T}}z_1\mathrm{d}t=\int_0^t(\dot{z}_1+k_1z_1)^{\mathrm{T}}z_1\mathrm{d}t=\int_0^t{z_1}^{\mathrm{T}}\dot{z}_1\mathrm{d}t+\int_0^t k_1{z_1}^{\mathrm{T}}z_1\mathrm{d}t$$

which actually means that

$$\int_0^t\left(\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\boldsymbol{\Theta}_1-k_1z_1\right)^{\mathrm{T}}z_1\mathrm{d}t=\frac{1}{2}\,{z_1}^{\mathrm{T}}z_1\big|_0^t \tag{29}$$

If the storage function for $\sum_1$ subsystem is selected as $\mathrm{S}_1(t)=0.5z_1(t)^{\mathrm{T}}z_1(t)$, then (29) turns to be

$$S_1(t)-S_1(0)\leqslant\int_0^t\left(\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\boldsymbol{\Theta}_1-k_1z_1\right)^{\mathrm{T}}z_1\mathrm{d}t$$

which implies $\sum_1$ subsystem is OFP(k_1) and has an excess of passivity.

For $\sum_2$ subsystem, define the input as z_1 and define the output as $-\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\boldsymbol{\Theta}_1$, then by calculating the energies injected into this subsystem, one can obtain

$$\begin{aligned}\int_0^t{z_1}^{\mathrm{T}}\left(-\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\boldsymbol{\Theta}_1\right)\mathrm{d}t&=-\int_0^t\left\{\mathrm{tr}\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\left(\beta_1\dot{\hat{\boldsymbol{W}}}_1+\alpha_1\|z_1\|^2\hat{\boldsymbol{W}}_1\right)\right\}\,\mathrm{d}t\\&=\beta_1\int_0^t\mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\dot{\tilde{\boldsymbol{W}}}_1\right\}\mathrm{d}t-\alpha_1\int_0^t\|z_1\|^2\mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\hat{\boldsymbol{W}}_1\right\}\,\mathrm{d}t\end{aligned}$$

Using the property of the Frobenius norm (20), one can further obtain

$$\begin{aligned}\int_0^t{z_1}^{\mathrm{T}}\left(-\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\boldsymbol{\Theta}_1\right)\mathrm{d}t&\geqslant\beta_1\int_0^t\mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\dot{\tilde{\boldsymbol{W}}}_1\right\}\mathrm{d}t+\alpha_1\int_0^t\|z_1\|^2\left(\left(\left\|\tilde{\boldsymbol{W}}_1\right\|_F-\frac{\Delta_{W_1}}{2}\right)^2-\frac{{\Delta_{W_1}}^2}{4}\right)\mathrm{d}t\\&\geqslant\frac{\beta_1}{2}\mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\tilde{\boldsymbol{W}}_1\right\}\big|_0^t-\alpha_1\frac{{\Delta_{W_1}}^2}{4}\int_0^t{z_1}^{\mathrm{T}}z_1\mathrm{d}t\end{aligned}$$

From the definition, one can see that subsystem $\sum_2$ is IFP ($-0.25\alpha_1\Delta_{W_1}^2$) with storage function $S_2=\frac{1}{2}\beta_1\mathrm{tr}\left\{\tilde{\boldsymbol{W}}_1^{\mathrm{T}}\tilde{\boldsymbol{W}}_1\right\}$. Because $v=-0.25\alpha_1\Delta_{W_1}^2<0$, $\sum_2$ has a shortage of passivity. Thus, for the feedback connection of $\sum_1$ and $\sum_2$, if $\pi+\sigma=k_1-0.25\alpha_1\Delta_{W_1}^2>0$, the shortage of passivity of $\sum_2$ can be compensated by the excess of passivity of $\sum_1$ subsystem. Then the resulting system is OFP, from which one can conclude that $\lim_{t\to\infty}z_1=\mathbf{0}$, $\lim_{t\to\infty}\tilde{\boldsymbol{W}}_1=\mathbf{0}$, which coincides with Theorem 1 under the assumption that $\tilde{\varepsilon}_1=0$.

From the aforementioned analysis, the passive property of the entire closed-loop system is established. Based on the analysis, one can see that the shortage of passivity of the dynamic parameter estimation subsystem ($\sum_2$) is compensated by the excess of passivity of the plant dynamics ($\sum_1$). Besides, from the passivity theory, one knows that the excess or shortage of passivity reflects the energies extracted or injected into the system. Thus, by tuning the parameters k_1, α_1, the dissipative rate of the closed-loop system can be changed to obtain the desired transient performance. Furthermore, the explicit relations between tracking performance and controller parameters can be obtained to provide methods to systematically improve the transient performance of the closed-loop system.

DOI: 10.1002/rnc

5. SIMULATION EVALUATION

To study and demonstrate the effectiveness and performance of the proposed control strategies, numerical simulations have been carried out using a set of governing equations of motion described by (1), (2), and (3) in conjunction with the proposed control law (9) with neural network parameter updating laws (12), (13). The configuration of the model is illustrated in Figure 2.

In the simulations, all the physical parameters are expressed in the international system of units, and therefore, their units are omitted for simplicity. The physical parameters in Table I [16] are considered for the simulation model, with a_i, b_i denoting the length represented in Figure 2 and m_i, I_i being the mass and moment of inertia of the i-th body.

The parameters of the DC motor in (3) are set as $\boldsymbol{k}_\tau = \text{diag}([2,2])$, $\boldsymbol{L} = \text{diag}([0.2,0.2])$, $\boldsymbol{R} = \text{diag}([5,5])$, $\boldsymbol{k}_e = \text{diag}([2,2])$. In the proposed controller, assume that the measured physical parameters are $\hat{a}_1 = \hat{a}_2 = \hat{b}_1 = 0.45$, $\hat{b}_0 = \hat{b}_2 = 0.55, \hat{m}_0 = 30$, $\hat{m}_1 = \hat{m}_2 = 3$, $\hat{I}_0 = 5, \hat{I}_1 = \hat{I}_2 = 0.5$, $\hat{\boldsymbol{k}}_\tau = \text{diag}([3,3])$, $\hat{\boldsymbol{L}} = \text{diag}([0.3,0.3])$, $\hat{\boldsymbol{R}} = \text{diag}([7,7])$, $\hat{\boldsymbol{k}}_e = \text{diag}([3,3])$. In simulations, the desired end-effector trajectory of the planar manipulator is chosen to be a circle in the inertia space, which is $\boldsymbol{x}_d = [0.3\sin(t),\ 1.6 + 0.3\cos(t)]^{\mathrm{T}}$. The time-varying gains of the two first-order low-pass filters are chosen to be $\epsilon_i(t) = 0.05e^{-0.1t}$ $(i = 2,3)$, which decrease to zero exponentially. The controller parameters are chosen as

$$\begin{array}{lllll} k_1 = 1, & k_2 = 3, & k_3 = 2, & \alpha_1 = 3, & \alpha_2 = 3, \quad \alpha_3 = 3 \\ \beta_1 = 1, & \beta_2 = 1, & \beta_3 = 1, & \nu_1 = 3, & \nu_2 = 3, \quad \nu_3 = 3 \\ \mu_1 = 1, & \mu_2 = 1, & \mu_3 = 1, & \varrho_1 = 10, & \varrho_2 = 0.05, \quad \varrho_3 = 0.05 \end{array} \tag{30}$$

The weight matrices $\boldsymbol{W}_i$ $(i = 1,2,3)$ are 7×2 matrix with all elements initialized to be zero. The neural network basis functions $\boldsymbol{\Theta}_i(\boldsymbol{r}_i)$, where $\boldsymbol{r}_i$ is the input, are selected to be $\boldsymbol{\Theta}_i = [\phi_{i1}, \ldots, \phi_{i7}]^{\mathrm{T}}$ with $\phi_{ij} = \exp(-\|\boldsymbol{r}_i - c_{ij}\|/\varpi^2)$, $\varpi = 4$, $[c_{i1}, \ldots, c_{i7}]^{\mathrm{T}} = [-3 \times \mathbf{1}^{\text{length}(\boldsymbol{r}_i)}, -2 \times \mathbf{1}^{\text{length}(\boldsymbol{r}_i)}, -1 \times \mathbf{1}^{\text{length}(\boldsymbol{r}_i)}, 0 \times \mathbf{1}^{\text{length}(\boldsymbol{r}_i)}, 1 \times \mathbf{1}^{\text{length}(\boldsymbol{r}_i)}, 2 \times \mathbf{1}^{\text{length}(\boldsymbol{r}_i)}, 3 \times \mathbf{1}^{\text{length}(\boldsymbol{r}_i)}]^{\mathrm{T}}$, for $i = 1,2,3$,

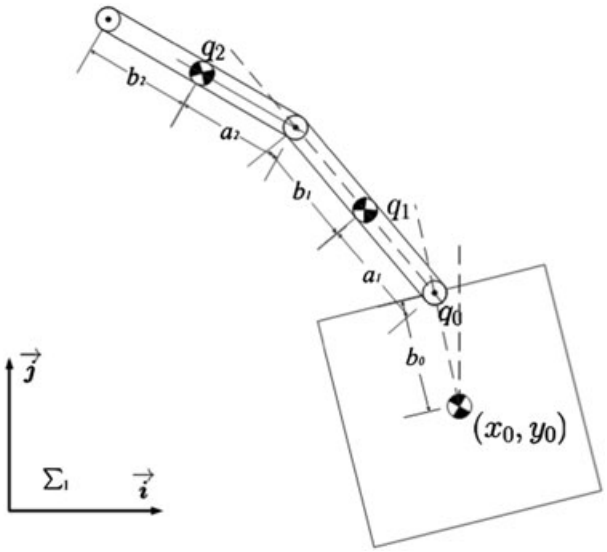

Figure 2. Configuration of the simulated free-floating space robotic system.

Table I. Physical parameters of the space robotic system.

Link	a_i	b_i	m_i	I_i
0 (base)		0.5	40	6.667
1	0.5	0.5	4	0.333
2	0.5	0.5	3	0.250

DOI: 10.1002/rnc

$j = 1, \ldots, 7$. In addition, for all numerical examples presented, the initial position of the center of mass of the spacecraft is set as $\boldsymbol{r}_{c0} = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathrm{T}}$; the initial configurations are set as $q_0(0) = 0$, $q_1(0) = -2$, and $q_2(0) = 2$; and the initial velocity is $\dot{q}_0(0) = 0$, $\dot{q}_1(0) = 0$, and $\dot{q}_2(0) = 0$, respectively. In this section, simulations are conducted separately to illustrate both the effectiveness and performance of the proposed control algorithms.

5.1. *Effectiveness verification*

Three different conditions are considered to emphasize the effectiveness and robustness of the proposed neural network- based adaptive dynamic surface controller. The first simulation is conducted without the compensation of neural networks. As can be seen in Figure 3, the actual trajectory does not converge to the desired one, which is mainly caused by the kinematic uncertainties. Because the Jacobian matrix based on the estimated physical parameters is used in the controller, the end-effector velocity computed by the controller from this nominal forward kinematics (the estimated Jacobian) does not equal to the actual one. When the wrong velocity information is used in the controller, the tracking error occurs. However, as shown in Figure 4, by utilizing the learning capability of the

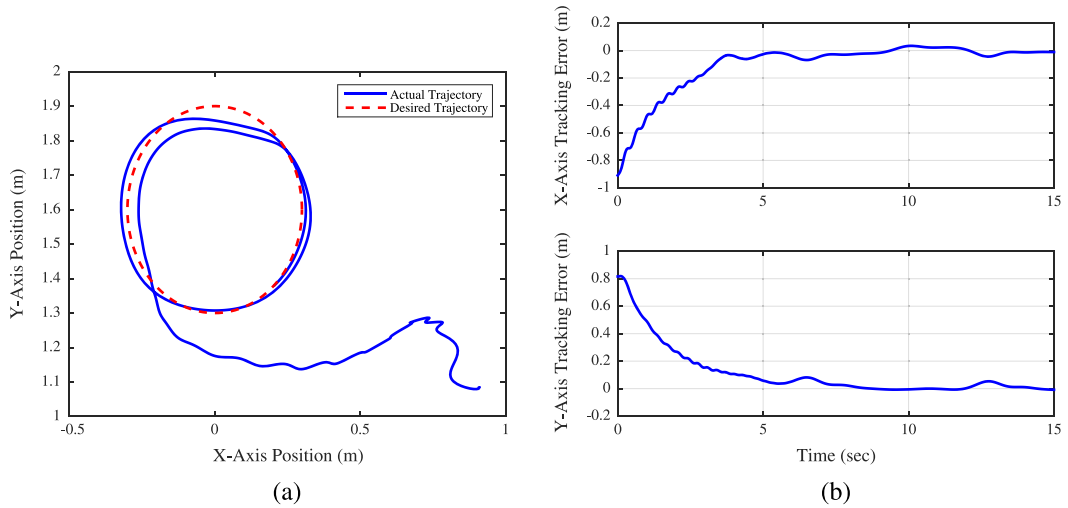

Figure 3. Simulated tracking results without utilization of neural network compensator: (a) trajectory in x-y plane and (b) tracking errors in x-axis and y-axis, respectively. [Colour figure can be viewed at wileyonlinelibrary.com]

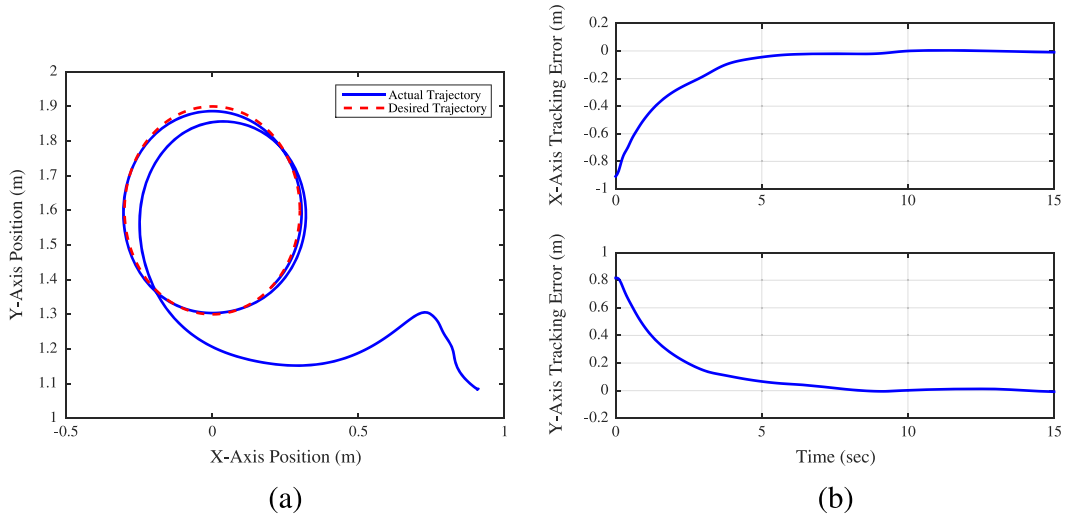

Figure 4. Simulated tracking results with neural network compensator: (a) trajectory in x-y plane and (b) tracking errors in x-axis and y-axis, respectively. [Colour figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/rnc

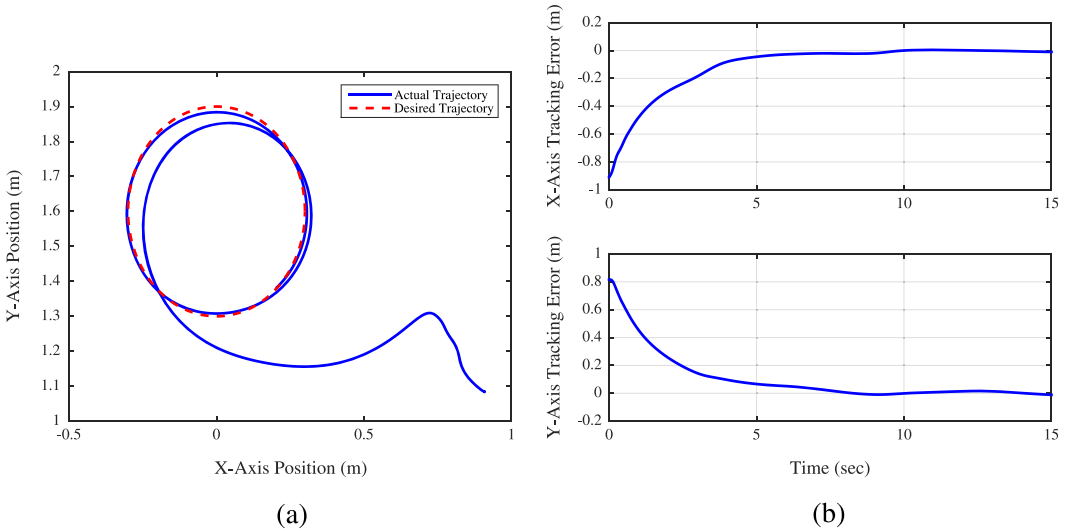

Figure 5. Simulated tracking results with neural network compensator and considering voltage disturbances: (a) trajectory in x-y plane and (b) tracking errors in x-axis and y-axis, respectively. [Colour figure can be viewed at wileyonlinelibrary.com]

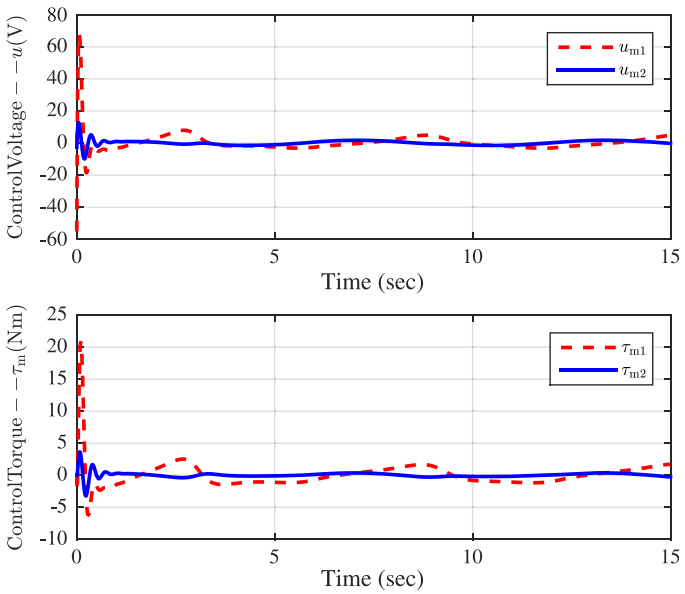

Figure 6. Profiles of actuator input voltage and torque. [Colour figure can be viewed at wileyonlinelibrary.com]

neural network, the difference between nominal kinematics and the unknown actual kinematics can be compensated and the results are rather compelling. This demonstrates the effectiveness of the initial motivation of this paper for combining neural network into this application. Furthermore, when additive bounded voltage disturbances are taken into consideration, which are assumed to be the form of $(1.5e^{-2t}, 0.3\cos(t))$ [22], significant degradation of the tracking performance is not observed, and this can be shown by the comparison in Figure 5. Furthermore, the signals of the closed-loop system, containing the armature voltage and torques generated by the actuators (Figure 6), norms of the neural network weighting matrix estimation (Figure 7(a)), norms of the neural network approximation error estimation (Figure 7(b)) are given respectively. From all these figures, it can be concluded that all the signals are rendered to be bounded by the proposed controller.

DOI: 10.1002/rnc

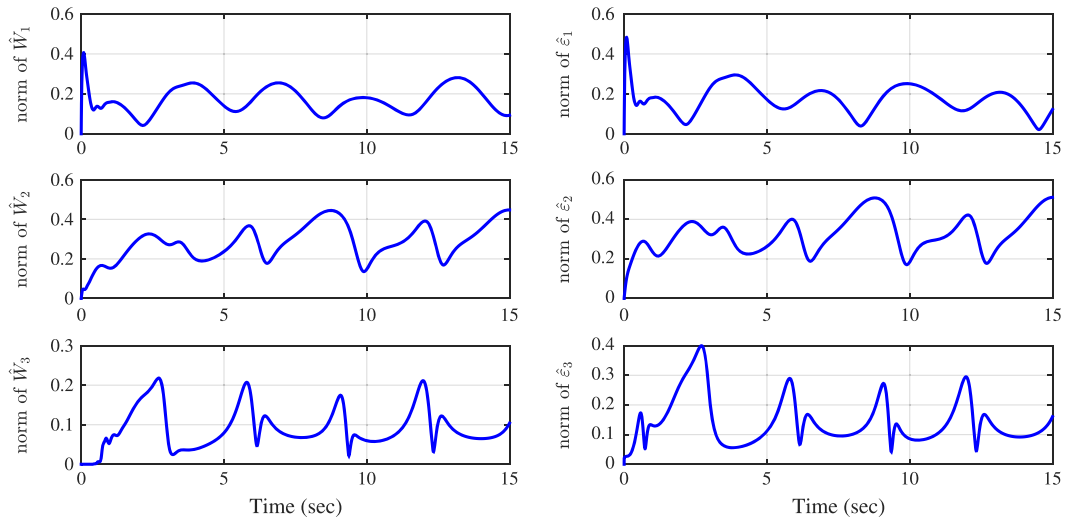

Figure 7. Evolution of the norms of (a) $\hat{W}_i$ $(i = 1, 2, 3)$ and (b) $\hat{\varepsilon}_i$ $(i = 1, 2, 3)$. [Colour figure can be viewed at wileyonlinelibrary.com]

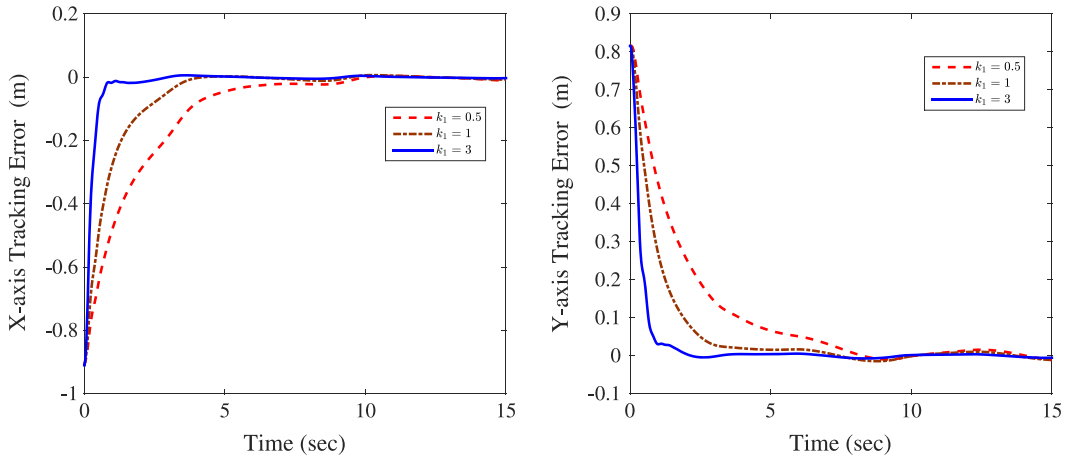

Figure 8. Tracking errors under different controller gains. [Colour figure can be viewed at wileyonlinelibrary.com]

5.2. *Control performance illustration*

Theorem 2 states the relations between the transient tracking performance and the controller gains. From (17), it can be observed that if a faster transient response of the tracking performance is required, one just needs to increase the value of controller gain k_1. Figure 8 shows three different simulations under the control gains of (a) $k_1 = 0.5, k_2 = 3$; (b) $k_1 = 1, k_2 = 3$; and (c) $k_1 = 3, k_2 = 3$. From simulation results with these controller gains, one can clearly observe that the greater the value of k_1, the faster the convergence of the tracking error. Thus, one just needs to appropriately tune the control parameters k_1, k_2, k_3 to obtain the desired transient tracking performance.

In conventional dynamic surface control methods, only the ultimately uniformly boundedness of the tracking error can be guaranteed. In this paper, the constant gain of the first-order low-pass filter in conventional control methods is replaced by a time-varying one, which can help to provide asymptotic stabilization of the tracking error. In the following, the asymptotic convergence of the tracking error is achieved and illustrated by a comparison with the conventional methods. In the simulation

DOI: 10.1002/rnc

using conventional methods, the constant filter gains are set to be 0.05, which are the same as the initial values of the time-varying filter gains. First, let us define the following performance index during time period $[t_0, t_f]$

$$p_{\text{pos}}(t_0, t_f) = \int_{t_0}^{t_f} \|\boldsymbol{x} - \boldsymbol{x}_d\| \, \mathrm{d}t$$

$$p_{\text{velo}}(t_0, t_f) = \int_{t_0}^{t_f} \|\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_d\| \, \mathrm{d}t$$

The previously defined tracking performance can describe both the position tracking errors and the velocity tracking errors during a certain time period. If the tracking error is only bounded stable, then after a transient period, the two performance indexes should not increase or decrease, but rather, they should fluctuate around certain values. However, if the tracking error is asymptotic stable, then they should keep decreasing until reaching to a value of zero.

Based on the earlier definition, both the position and velocity tracking performance indexes are computed with improved dynamic surface control when the controller parameters are selected as (30) and with conventional control in which the filter gain remains to be a constant. The simulation result is demonstrated in Figures 9 and 10, respectively. From the two figures, it can be concluded

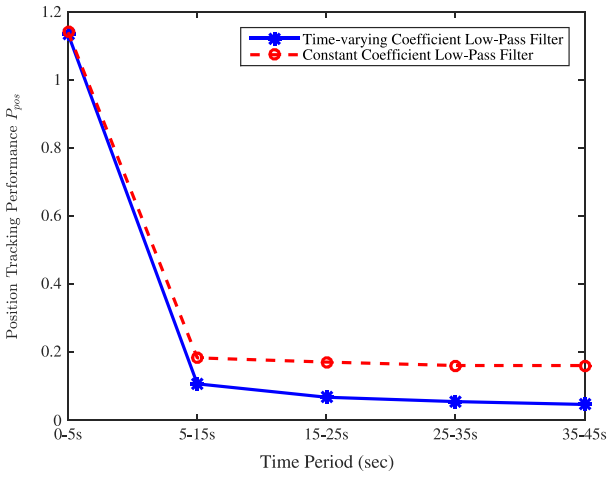

Figure 9. Graphic visualization of the position tracking performance comparison. [Colour figure can be viewed at wileyonlinelibrary.com]

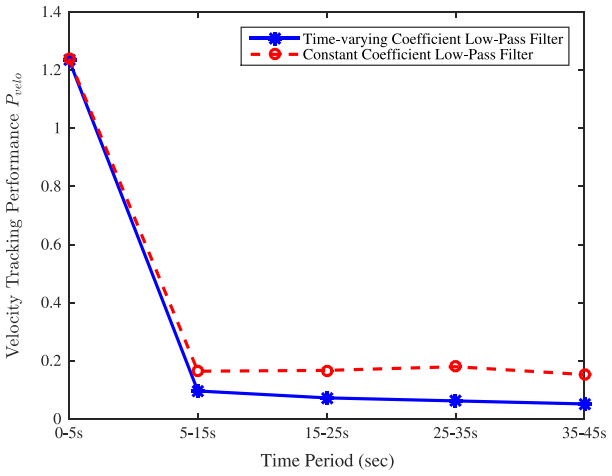

Figure 10. Graphic visualization of the velocity tracking performance comparison. [Colour figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/rnc

that the conventional constant gain-based dynamic surface control can only guarantee bounded stability of the tracking errors, while the proposed time-varying/adaptive dynamic surface control achieves asymptotic stability with smaller tracking errors.

6. CONCLUSIONS

By utilizing and combining its learning capability and unique function approximation property of the RBF-based neural network with dynamic surface control method, this paper developed an adaptive/intelligent dynamic surface control strategy for achieving improved task-space trajectory tracking control of robot manipulator systems in the presence of uncertainties due to the kinematics, dynamics, and actuator modeling. For faster convergence and better tracking performance, the constant gain first-order low-pass filter is replaced with a time-varying gain in the first-order low-pass filter such that the asymptotic stabilization of the tracking error has been achieved. In addition, the passivity structure of the designed robustifying adaptation law for the RBF neural network is analyzed. More importantly, the $\mathcal{L}_2$ transient tracking performance is analyzed, which offers explicit guidelines in tuning the control parameters to meet the specific transient tracking performance requirement. Simulation results on a free-floating space robot demonstrate that the proposed controller outperforms the conventional dynamic surface control methods. It should be pointed out that in this study, the controller was derived by assuming that full knowledge of states is available. However, the availability of velocity measurements is not always satisfied owing to either cost limitations or implementation constraints. Therefore, as one of future works, task-space trajectory tracking control design for free-floating space robots without both task-space and angular velocity measurements will be investigated.

REFERENCES

1. Lewis FL, Abdallah CT, Dawson DM, Lewis FL. *Robot Manipulator Control: Theory and Practice*. Marcel Dekker: New York, 2004.
2. Spong MW, Vidyasagar M. *Robot Dynamics and Control*. John Wiley: New York, 2008.
3. Huang CQ, Wang XG, Wang ZG. A class of transpose Jacobian-based NPID regulators for robot manipulators with an uncertain kinematics. *Journal of Robotic Systems* 2002; **19**(11):527–539.
4. Huang CQ, Xie LF, Liu YL. PD plus error-dependent integral nonlinear controllers for robot manipulators with an uncertain Jacobian matrix. *ISA Transactions* 2012; **51**(6):792–800.
5. Liu C, Cheah CC, Slotine JJE. Adaptive task-space regulation of rigid-link flexible-joint robots with uncertain kinematics. *Automatica* 2008; **44**(7):1806–1814.
6. Cheah CC, Liu CA, Slotine JJE. Adaptive Jacobian vision based control for robots with uncertain depth information. *Automatica* 2010; **46**(7):1228–1233.
7. Yazarel H, Cheah CC. Task-space adaptive control of robotic manipulators with uncertainties in gravity regressor matrix and kinematics. *IEEE Transactions on Automatic Control* 2002; **47**(9):1580–1585.
8. Cheah CC, Hirano M, Kawamura S, Arimoto S. Approximate Jacobian control with task-space damping for robot manipulators. *IEEE Transactions on Automatic Control* 2004; **49**(5):752–757.
9. Cheah CC. Task-space PD control of robot manipulators: unified analysis and duality property. *The International Journal of Robotics Research* 2008; **27**(10):1152–1170.
10. Dixon WE. Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control* 2007; **52**(3):488–493.
11. Liu C, Cheah CC. Task-space adaptive setpoint control for robots with uncertain kinematics and actuator model. *IEEE Transactions on Automatic Control* 2005; **50**(11):1854–1860.
12. Cheah CC, Liu C, Slotine JJE. Adaptive tracking control for robots with unknown kinematic and dynamic properties. *International Journal of Robotics Research* 2006; **25**(3):283–296.
13. Liang XW, Huang XH, Wang M, Zeng XJ. Adaptive task-space tracking control of robots without task-space- and joint-space-velocity measurements. *IEEE Transactions on Robotics* 2010; **26**(4):733–742.
14. Liu C, Cheah CC, Slotine JJE. Adaptive Jacobian tracking control of rigid-link electrically driven robots based on visual task-space information. *Automatica* 2006; **42**(9):1491–1501.
15. Wang HL, Xie YC. Adaptive Jacobian position/force tracking control of free-flying manipulators. *Robotics and Autonomous Systems* 2009; **57**(2):173–181.
16. Wang HL, Xie YC. Passivity based adaptive Jacobian tracking for free-floating space manipulators without using spacecraft acceleration. *Automatica* 2009; **45**(6):1510–1517.
17. Ahmadipour M, Khayatian A, Dehghani M. Adaptive control of rigid-link electrically driven robots with parametric uncertainties in kinematics and dynamics and without acceleration measurements. *Robotica* 2014; **32**(7):1153–1169.

DOI: 10.1002/rnc

18. Wang HL, Xie YC. Adaptive inverse dynamics control of robots with uncertain kinematics and dynamics. *Automatica* 2009; **45**(9):2114–2119.
19. Garcia-Rodriguez R, Parra-Vega V. Cartesian sliding PID control schemes for tracking robots with uncertain Jacobian. *Transactions of the Institute of Measurement and Control* 2012; **34**(4):448–462.
20. Wang HL, Xie YC. Prediction error based adaptive Jacobian tracking of robots with uncertain kinematics and dynamics. *IEEE Transactions on Automatic Control* 2009; **54**(12):2889–2894.
21. Wang HL, Xie YC. Prediction error based adaptive Jacobian tracking for free-floating space manipulators. *IEEE Transactions on Aerospace and Electronic Systems* 2012; **48**(4):3207–3221.
22. Cheng L, Hou ZG, Tan M. Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model. *Automatica* 2009; **45**(10):2312–2318.
23. Cheah CC, Liu C, Slotine JJE. Adaptive Jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models. *IEEE Transactions on Automatic Control* 2006; **51**(6):1024–1029.
24. Umetani Y, Yoshida K. Resolved motion rate control of space manipulators with generalized Jacobian matrix. *IEEE Transactions on Robotics and Automation* 1989; **5**(3):303–314.
25. Dubowsky S, Papadopoulos E. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on Robotics and Automation* 1993; **9**(5):531–543.
26. Zou AM, Kumar KD, de Ruiter AHJ. Robust attitude tracking control of spacecraft under control input magnitude and rate saturations. *International Journal of Robust and Nonlinear Control* 2016; **26**(4):799–815.
27. Rong-Jong Wai, Muthusamy R. Design of fuzzy-neural-network-inherited backstepping control for robot manipulator including actuator dynamics. *IEEE Transactions on Fuzzy Systems* 2014; **22**(4):709–722.
28. Zhao B, Xian B, Zhang Y, Zhang X. Nonlinear robust sliding mode control of a quadrotor unmanned aerial vehicle based on immersion and invariance method. *International Journal of Robust and Nonlinear Control* 2015; **25**(18):3714–3731.
29. Shen Q, Shi P. Distributed command filtered backstepping consensus tracking control of nonlinear multiple-agent systems in strict-feedback form. *Automatica* 2015; **53**(0):120–124.
30. Yip PP, Hedrick JK. Adaptive dynamic surface control: a simplified algorithm for adaptive backstepping control of nonlinear systems. *International Journal of Control* 1998; **71**(5):959–979.
31. Swaroop D, Hedrick JK, Yip PP, Gerdes JC. Dynamic surface control for a class of nonlinear systems. *IEEE Transactions on Automatic Control* 2000; **45**(10):1893–1899.
32. Liu Z, Lai GY, Zhang Y, Chen X, Chen CLP. Adaptive neural control for a class of nonlinear time-varying delay systems with unknown hysteresis. *IEEE Transactions on Neural Networks and Learning Systems* 2014; **25**(12): 2129–2140.
33. Xu B, Shi ZK, Yang CG, Sun FC. Composite neural dynamic surface control of a class of uncertain nonlinear systems in strict-feedback form. *IEEE Transactions on Cybernetics* 2014; **44**(12):2626–2634.
34. Park BS. Adaptive formation control of underactuated autonomous underwater vehicles. *Ocean Engineering* 2015; **96**:1–7.
35. Park BS. Neural network-based tracking control of underactuated autonomous underwater vehicles with model uncertainties. *Journal of Dynamic Systems, Measurement and Control* 2015; **137**(2):021004 1–7.
36. Ge SSam, Hang CC, Lee TH, Zhang T. *Stable Adaptive Neural Network Control*. Kluwer Academic: Boston, 2002.
37. Shores TS. *Applied Linear Algebra and Matrix Analysis*. Springer: New York, 2007.
38. Narendra KS, Annaswamy AM. A new adaptive law for robust adaptation without persistent excitation. *IEEE Transactions on Automatic Control* 1987; **32**(2):134–145.
39. Khalil HK. *Nonlinear Systems*. PTR Prentice-Hall: Upper Saddle River, NJ, 2002.

DOI: 10.1002/rnc