Hindawi
Complexity
Volume 2019, Article ID 1406534, 16 pages
https://doi.org/10.1155/2019/1406534

WILEY | Hindawi

Research Article

Position/Force Tracking Impedance Control for Robotic Systems with Uncertainties Based on Adaptive Jacobian and Neural Network

Jinzhu Peng, Zeqi Yang, and Tianlei Ma

School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China

Correspondence should be addressed to Tianlei Ma; tlma@zzu.edu.cn

Received 7 September 2018; Revised 27 November 2018; Accepted 20 December 2018; Published 3 January 2019

Academic Editor: Marcin Mrugalski

In this paper, an adaptive Jacobian and neural network based position/force tracking impedance control scheme is proposed for controlling robotic systems with uncertainties and external disturbances. To achieve precise force control performance indirectly by using the position tracking, the control scheme is divided into two parts: the outer-loop force impedance control and the inner-loop position tracking control. In the outer-loop, an improved impedance controller, which combines the traditional impedance relationship with the PID-like scheme, is designed to eliminate the force tracking error quickly and to reduce the force overshoot effectively. In this way, the satisfied force tracking performance can be achieved when the manipulator contacts with environment. In the inner-loop, an adaptive Jacobian method is proposed to estimate the velocities and interaction torques of the end-effector due to the system kinematical uncertainties, and the system dynamical uncertainties and the uncertain term of adaptive Jacobian are compensated by an adaptive radial basis function neural network (RBFNN). Then, a robust term is designed to compensate the external disturbances and the approximation errors of RBFNN. In this way, the command position trajectories generated from the outer-loop force impedance controller can be then tracked so that the contact force tracking performance can be achieved indirectly in the forced direction. Based on the Lyapunov stability theorem, it is proved that all the signals in closed-loop system are bounded and the position and velocity errors are asymptotic convergence to zero. Finally, the validity of the control scheme is shown by computer simulation on a two-link robotic manipulator.

1. Introduction

The development of technology has aroused people's ever-growing interest on the safer and reliable, higher accuracy of robots. Many position tracking control methods have made some achievements in various application fields in recent years, such as cutting, grinding, welding, and so on [1–3]; the insufficient compliance for manipulator is always a key problem and has been given a lot of attention in robotics field, especially in complex and accurate tasks and even in collaboration with humans [4, 5]. Therefore, the applications of human-robot interaction (HRI) have become a new development domain and direction [6, 7]. It is necessary to develop an interaction control method that achieves position tracking and reliably adapts the force exerted on the environment in order to avoid damage both in the environment and in the manipulator itself [8].

The traditional force control methods proposed since 1980s can be mainly classified into two categories, namely, hybrid position/force control [9] and impedance control [10]. Impedance control method is to apply the motion trajectory and contact force into one dynamic framework, which can avoid the separate control processes of positions and force. In general, compared with the hybrid position/force control, the impedance control method can reduce the complexity of controller design and has better adaptability and robustness in the complex dynamic tasks [11]. Hence, in order to achieve the stable force tracking control, Chan and Yao [12] integrated sliding mode control into the impedance control method, which includes ideal impedance relationship in the sliding mode. Seul et al. [13, 14] proposed a series of force tracking impedance control methods. In [13] the impedance control schemes were classified into torque-based and position-based impedance methods based on different implementations of

impedance function. Then, the manipulator was controlled in the free space and in the contact space [14], where the impedance function was improved to achieve position and force tracking, and the force error could be converged to zero for any environmental stiffness by using an adaptive technique. The force tracking performance of these methods, however, greatly depends on the accurate environmental information and the precise robotic system model.

Recently, many researchers tried to introduce the intelligent control methods into the force control to improve the tracking performance and robustness of the robotic system [15–20]. In [21], the neural network control technique was applied in impedance controller to compensate the uncertainties in an online manner. Li and Liu [22] designed an adaptive impedance hybrid controller, which could implement the desired contact force and track the command position in orthogonal subspace without precise environmental information. Jhan and Lee [23] proposed an adaptive fuzzy NN-based impedance control, where the adaptive fuzzy NN was used to approximate the robot dynamical model, so that the actual parameters of the manipulator need not be precisely known. In [24], a fuzzy logic system was applied as an approximator to estimate the unknown system dynamics, and then the proposed adaptive fuzzy back-stepping position/force control method could ensure all the signals of the close-loop system ultimately uniformly bounded. Duan et al. [25] presented an adaptive variable impedance control, which can adapt the environmental stiffness uncertainties. However, the contact force overshoot, which generated from the contact between robot and environment, was usually unconcerned in the above methods design. An improved impedance relationship was proposed in [26, 27] to reduce the contact force overshoot and to achieve the direct accurate time-varying force tracking. Roveda et al. [28, 29] calculated the parameters online in external controller to reduce the contact force overshoot. The methods can reduce the contact force overshoot provided that the model parameters of robotic systems are precisely known.

Since the model parameters of robotic systems can not be precisely obtained generally, lots of methods such as the adaptive Jacobian scheme, neural networks, back-stepping technology were employed to facilitate the tracking control of the manipulator with kinematic and dynamic uncertainties [30–32]. Liang et al. [33] designed a task-space observer to estimate the task-space positions and velocities simultaneously while NNs were employed to further improve the control performance through approximating the modified robot dynamics. These methods designed in Cartesian space can avoid inverse kinematical solution; however, there were only the position control problems included. To obtain the force control in Cartesian space, Jung et al. [13] proposed the position-based impedance control method, which can introduce the position-based control methods into the impedance control schemes. Literatures [27, 34, 35] used two-loop controllers, which included the inner-loop position control and the outer-loop force control, to achieve the position and force tracking control. Based on these methods, Bonilla et al. [36] proposed an inverse dynamical control law based on impedance control to achieve the position path tracking in both free and constrained spaces, which mainly focused on developing the compliant control scheme for constrained path tracking; however, the force tracking was not involved.

Consider the above drawbacks, in this paper, an adaptive neural network position/force tracking impedance controller is proposed for controlling robotic system with uncertainties in both free and contact spaces, where the force tracking can be achieved by position-based control method and the force overshoot can also be reduced efficiently. Consequently, the main contributions of this paper are presented as follows:

(1) Two-loop control architecture is presented to facilitate the position/force tracking impedance control for robotic systems, which are the outer-loop force impedance control and the inner-loop position tracking control.

(2) In the outer-loop, the improved impedance relationship based on PID-like scheme is proposed to reduce force overshoot when the end-effector contacts with environment.

(3) In the inner-loop, the adaptive Jacobian and RBFNN methods are employed to compensate the system model parameter and dynamical uncertainties. Then, the approximation errors of RBFNN and the external disturbances are restrained by a robust term. In this way, the desired contact force and the desired position trajectories can be tracked efficiently by the proposed intelligent position-based impedance control method.

(4) Based on the Lyapunov stability theorem, it is proved that all the signals in the closed-loop system are bounded, the position and velocity errors can be asymptotically converged to zero, and the contact force can be tracked to the desired force.

The reminder of this paper is organized as follows. In Section 2, some theoretical preliminaries are addressed, which consist of the mathematical notations, the RBFNN, and the details on dynamics of robotic system. In Section 3, the outer-loop position-based force tracking impedance control is designed to regulate interaction force when the robot contact with environment. Based on the reference trajectory generated from the outer-loop force impedance control, the adaptive neural network position tracking control scheme and its stability analysis are derived in Section 4. In Section 5, simulation results are presented to verify the effectiveness of the proposed method, and conclusions are drawn in Section 6.

2. Problem Formulation and Preliminaries

In this paper, standard notations are used. We denote $\mathfrak{R}$ as the real number set, $\mathfrak{R}^n$ as the n-dimensional vector space, and $\mathfrak{R}^{n\times n}$ represents the $n \times n$ real matrix space. The norm of vector $x \in \mathfrak{R}^n$ and that of matrix $A \in \mathfrak{R}^{n\times n}$ are defined as $\|x\| = \sqrt{x^{\mathrm{T}}x}$ and $\|A\| = \sqrt{\mathrm{tr}(A^{\mathrm{T}}A)}$, respectively. If y is a scalar, let $\|y\|$ denote the absolute value. $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ are the minimum and the maximum eigenvalue of matrix A, respectively. $I_{n\times n}$ is the $n \times n$ identity matrix.

2.1. Description of RBFNN. In general, RBFNN has fast learning convergence speed and strong capability and has been proved in theory that RBFNN can approximate any

nonlinear continuous function over a compact set to arbitrary accuracy [37]. The RBFNN structure is described as follows:

$$y(W,x) = W^{\mathrm{T}}\phi(x), \tag{1}$$

where $x = [x_1,\ldots,x_{N_n}]^{\mathrm{T}} \in \Re^{N_n}$ is the input vector, $y(W,x)$ is the output vector, N_n is the control input dimension, N_s is the neuron node number, N_o is the output dimension, $W = [w_1,\ldots,w_{N_o}]^{\mathrm{T}} \in \Re^{N_s\times N_o}$ is the weight matrix with $w_k \in \Re^{N_s}$, $k = 1,\ldots,N_o$, $\phi(x) = [\phi_1(x),\ldots,\phi_{N_s}(x)]^{\mathrm{T}} \in \Re^{N_s}$ is the RBFNN active function with hidden layer output function $\phi_j(x)$, and the Gaussian function is chosen as follows:

$$\phi_j(x) = \exp\left[-\frac{\left(x-c_j\right)^{\mathrm{T}}\left(x-c_j\right)}{\delta_j^{\,2}}\right], \tag{2}$$

where $j = 1,\ldots,N_s$, $c = [c_1,\ldots,c_{N_n}]^{\mathrm{T}} \in \Re^{N_n}$ is the center of the jth neuron node, and δ_j is the width of the jth neuron.

Numerous results indicate that for any continuous smooth function $f(x) : \Omega_x \longrightarrow R$ over a compact set $\Omega_x \in \Re^{N_n}$, applying RBFNN (1) to approximate $f(x)$, if N_s is sufficiently large, a set of ideal bounded weights W^* exist, and we have

$$f(x) = W^{*\mathrm{T}}\phi(x) + \varepsilon(x), \tag{3}$$

where $\varepsilon(x)$ is the RBFNN reconstruction error. Since the ideal weight W^* is unknown, the estimated weight $\widehat{W}$ is generally used to replace W^* to approximate the unknown, continuous, nonlinear function; that is,

$$\widehat{f}(x) = \widehat{W}^{\mathrm{T}}\phi(x). \tag{4}$$

where $\widehat{W}$ is the estimated weight matrix and can be trained by a weight learning law. We assume that Ω_W is existed, and ideal parameter is in the compact set Ω_W^*, which is defined as $\Omega_W^* = \{W^* \in \Re^{N_s\times N_o} : \|W^*\| \le E_W\}$. The ideal parameter can be given as

$$W^* = \arg\min_{W^*\in\Omega_W^*}\left\{\sup\left|f(x) - \widehat{W}^{\mathrm{T}}\phi(x)\right|\right\}. \tag{5}$$

Assumption 1. The RBFNN reconstruction error is bounded and satisfied

$$\left\|\varepsilon(\chi)\right\| \le \delta_\varepsilon, \tag{6}$$

where δ_ε is a positive constant.

In this paper, to ensure the control performance of manipulator, the RBFNN is used as a compensator to eliminate the dynamical uncertainties and the uncertain term of adaptive Jacobian in robotic system.

2.2. Robotic Manipulator Dynamics and Properties. Consider a general n-degree of freedom (DOF) rigid robotic manipulators with uncertain dynamics, the dynamical model can be described as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) + \tau_d = \tau - \tau_e \tag{7}$$

$$\tau_e = J^{\mathrm{T}}(q)F_e, \tag{8}$$

where q, $\dot{q}$, and $\ddot{q} \in \Re^n$ represent the joint angular position vectors, velocity vectors, and acceleration vectors of the manipulator, respectively; $M(q) \in \Re^{n\times n}$ is the positive definite and symmetric inertia matrix; $C(q,\dot{q}) \in \Re^{n\times n}$ represents the effect of centrifugal and Coriolis forces; $G(q) \in \Re^n$ is the gravity vector; $\tau_f(\dot{q}) \in \Re^n$ is the friction effects; $\tau_d \in \Re^n$ denotes the bounded unknown disturbances including unknown payload dynamics and unstructured dynamics; $\tau \in \Re^n$ is the torque input vector; $\tau_e \in \Re^n$ is the interaction torque vector when the manipulator contacts with environment; $J(q) \in \Re^{m\times n}$ represents the Jacobian matrix from join space to task space; and $F_e \in \Re^m$ is the contact force at the end-effector.

The following properties and assumption are required for the subsequent development.

Property 2. The inertia matrix $M(q)$ is positive definite and symmetric, which is uniformly bounded and satisfies

$$0 \le m_m I_{n\times n} \le M(q) \le m_M I_{n\times n}, \quad \forall q \in \Re^n, \tag{9}$$

where m_m and m_M are some positive constants.

Property 3. The matrix $\dot{M}(q) - 2C(q,\dot{q})$ is skew-symmetric; i.e.,

$$\zeta^{\mathrm{T}}\left(\dot{M}(q) - 2C(q,\dot{q})\right)\zeta = 0, \quad \forall\zeta \in \Re^n. \tag{10}$$

Property 4. The norm of the Coriolis and centrifugal forces matrix $C(q,\dot{q})$ is bounded and satisfies

$$\left\|C(q,\dot{q})\right\| \le C_b\left\|\dot{q}\right\|, \tag{11}$$

where C_b a positive constant.

Assumption 5. The unknown disturbance term is bounded by $\|\tau_d\| \le \tau_D$ where τ_D is a positive constant.

Let $X \in \Re^m$ be the position vector of the end-effector in task space. The relation between task space and joint space can be described by forward kinematics as

$$X = P(q), \tag{12}$$

where $P(\cdot)$ is the forward kinematics map, generally a nonlinear transformation between task space and joint space. The task-space velocities of end-effector $\dot{X}$ is related to joint velocities $\dot{q}$ as

$$\dot{X} = J(q)\dot{q}. \tag{13}$$

Assumption 6. In general, a manipulator should work in a finite task space. The matrix $J^{-1}(q) \in \Re^{m\times m}$ is the inverse matrix of the Jacobian matrix $J(q)$ when $m = n$. When $m \ne n$, the inverse matrix of the $J(q)$ can be represented as

$$J^+(q) = J^{\mathrm{T}}(q)\left(J(q)J^{\mathrm{T}}(q)\right)^{-1}, \tag{14}$$

where $J^+(q)$ denotes the generalized inverse matrix of $J(q)$. Similar notations hold for the estimate Jacobian matrix $\widehat{J}(q,\theta)$ as detailed later in Section 4.

In general, the uncertainties of the model parameters and the robot dynamics decreased the control performances of the robotic system directly. In this paper, an improved impedance relationship is designed to derive the reference trajectory planning scheme so that the reference position trajectories can be then generated. Then, an intelligent robust position-based impedance control scheme is proposed to achieve position trajectory tracking performance and the contact force tracking performance, where an adaptive Jacobian and RBFNN methods are designed to compensate the system uncertainties of robotic manipulator.

3. Design of Outer-Loop Force Tracking Control Based on Improved Impedance Relationship

Impedance control method regulates the relationship between the position and force by selecting suitable impedance parameters. According to literature [10], the traditional impedance relationship of the robotic system satisfies

$$M\left(\ddot{X}_r - \ddot{X}\right) + B\left(\dot{X}_r - \dot{X}\right) + K\left(X_r - X\right) = F_e, \tag{15}$$

where $X_r \in \Re^m$ is the planned reference trajectory of end-effector for position control, which determined from environmental position and parameters, impedance parameters, and desired contact force; M, B, and $K \in \Re^{m\times m}$ are the desired inertia, damping, and stiffness matrices, respectively; $F_e \in \Re^m$ is the force exerted on the environment at the end-effector.

In general, the contact force F_e is determined by environmental stiffness and environmental damping; therefore, a second-order nonlinear function is used to approximate the environmental model, which can be expressed as

$$F_e = K_e\left(X - X_e\right) + B_e\left(\dot{X} - \dot{X}_e\right), \tag{16}$$

where $K_e \in \Re^{m\times m}$ and $B_e \in \Re^{m\times m}$ denote the diagonal symmetric positive definite environmental stiffness and damping matrices, respectively, and $X_e \in \Re^m$ is the environment position vector. Assume that the environment position is a constant, we have $\dot{X}_e = 0$, then (16) can be rewritten as

$$F_e = K_e\left(X - X_e\right) + B_e\dot{X}, \tag{17}$$

3.1. Reference Trajectory Planning. The force tracking response cannot generally be achieved quickly by using the traditional impedance control method shown in (15); when the end-effector of manipulator contacts with environment, the force overshoot may result in task failures. Therefore, in this paper, a PID-like impedance relationship is designed to improve the force tracking performance, which can be expressed as

$$\begin{aligned} &M\left(\ddot{X}_r - \ddot{X}\right) + B\left(\dot{X}_r - \dot{X}\right) + K\left(X_r - X\right) \\ &\quad = K_pF_e + K_i\int_0^t F_e d\tau + K_d\dot{F}_e, \end{aligned} \tag{18}$$

where K_p, K_i, and $K_d \in \Re^{m\times m}$ are the diagonal symmetric positive definite parameter matrices. The introduced PID-like force compensation can achieve a better expectation than the pure force in (15) so that the contact force generated at the end-effector can quickly converge to the desired value and reduce the force overshoot.

For convenience, we consider the force is exerted on one direction only. Replacing x_r, x, x_e, f_e, f_d, m, b, k, k_p, k_i, k_d, k_e, b_e by X_r, X, X_e, F_e, F_d, M, B, K, K_p, K_i, K_d, K_e, B_e, respectively, then the improved impedance function (18) can be rewritten as

$$\begin{aligned} &m\left(\ddot{x}_r - \ddot{x}\right) + b\left(\dot{x}_r - \dot{x}\right) + k\left(x_r - x\right) \\ &\quad = k_pf_e + k_i\int_0^t f_e d\tau + k_d\dot{f}_e, \end{aligned} \tag{19}$$

and the environment model (17) becomes

$$f_e = k_e\left(x - x_e\right) + b_e\dot{x}, \quad x \geq x_e. \tag{20}$$

Define $e_f = f_d - f_e$; (19) and (20) can be rewritten as

$$\begin{aligned} &m\left(\ddot{x}_r - \ddot{x}\right) + b\left(\dot{x}_r - \dot{x}\right) + k\left(x_r - x\right) \\ &\quad = k_p\left(f_d - e_f\right) + k_i\int_0^t \left(f_d - e_f d\right)\tau + k_d\left(\dot{f}_d - \dot{e_f}\right) \end{aligned} \tag{21}$$

and

$$f_d - e_f = k_e\left(x - x_e\right) + b_e\dot{x}, \quad x \geq x_e. \tag{22}$$

Taking Laplace transform to (21) and (22) yields

$$\begin{aligned} &\left(ms^2 + bs + k\right)\left(X_r^s(s) - X^s(s)\right) \\ &\quad = \left(k_p + \frac{k_i}{s} + k_ds\right)\left(F_d^s(s) - E_f^s(s)\right) \end{aligned} \tag{23}$$

and

$$F_d^s(s) - E_f^s(s) = \left(b_es + k_e\right)X^s(s) - \frac{k_ex_e}{s}. \tag{24}$$

Then, (24) can be rewritten as

$$X^s(s) = \frac{s\left(F_d^s(s) - E_f^s(s)\right) + k_ex_e}{b_es^2 + k_es}. \tag{25}$$

Substituting (25) into (23) yields

$$\begin{aligned} &\left(ms^2 + bs + k\right)\left[X_r^s(s) - \frac{s\left(F_d^s(s) - E_f^s(s)\right) + k_ex_e}{b_es^2 + k_es}\right] \\ &\quad = \left(k_p + \frac{k_i}{s} + k_ds\right)\left(F_d^s(s) - E_f^s(s)\right). \end{aligned} \tag{26}$$

Then, the force tracking ${E_f}^s(s)$ in frequency domain can be obtained as

$$\begin{aligned} &E_f^s(s) \\ &\quad = F_d^s(s) \\ &\qquad + \frac{\left(ms^2 + bs + k\right)\left[k_ex_e - \left(b_es^2 + k_es\right)X_r^s(s)\right]}{T^s(s)}, \end{aligned} \tag{27}$$

where

$$T^s(s) = \left(k_d s^2 + k_p s + k_i\right)\left(b_e s + k_e\right) + \left(ms^3 + bs^2 + ks\right). \tag{28}$$

Then, steady state force tracking error can be obtained as

$$e_{ss} = \lim_{s\longrightarrow 0} sE_f^s(s) = \lim_{s\longrightarrow 0}\left\{sF_d^s(s) + \frac{\left(ms^3 + bs^2 + ks\right)\left[k_e x_e - \left(b_e s^2 + k_e s\right)X_r^s(s)\right]}{T^s(s)}\right\}. \tag{29}$$

To ensure the steady state force error e_{ss} to be zero as the system approaches the stable equilibrium state, the reference position trajectory can be designed as

$$X_r^s(s) = \frac{k_e x_e}{s\left(b_e s + k_e\right)} + \frac{T^s(s)}{s\left(b_e s + k_e\right)\left(ms^2 + bs + k\right)}F_d^s(s). \tag{30}$$

According to (30), it is obvious that the planned reference position trajectory $x_r(t)$ is a dynamic function including the desired force $f_d(t)$, the environmental position x_e, the environmental stiffness k_e and damping b_e, the impedance parameters m, b, k, and the PID-like parameters k_p, k_i, k_d. Assumed that the parameters in the improved impedance relationship (18) have been selected properly and the environmental information is accurately obtained, the reference position trajectory $x_r(t)$ can be generated by the desired input force $f_d(t)$.

3.2. Position-Based Impedance Control Scheme. In general, the traditional impedance control was classified into two methods: the position-based and the torque-based impedance control [13]. In the position-based impedance control, the outer-loop force impedance, and the inner-loop position tracking can be designed separately, where the force tracking performance mainly depends on the accuracy of the position tracking control in the inner-loop. Therefore, the position-based impedance control method has been widely applied in complex industrial systems such as servo control.

Denote the position command X_c as the control input of the inner-loop position tracking control,

$$X_c = \begin{cases} X_d & (\text{free space}) \\ X_r - Z & (\text{contact space}) \end{cases} \tag{31}$$

where $Z = X_r - X$. Based on the improved impedance equation (18), the desired relationship between Z and F_e can be represented as

$$M\ddot{Z} + B\dot{Z} + KZ = K_p F_e + K_i \int_0^t F_e d\tau + K_d \dot{F}_e. \tag{32}$$

Taking Laplace transform to (32) yields

$$Z^s(s) = \frac{K_d s^2 + K_p s + K_i}{Ms^3 + Bs^2 + Ks}F_e^s(s). \tag{33}$$

To achieve the force tracking performance, the contact force F_e is regulated to track the desired force F_d; an intelligent-based robust position tracking controller will be designed in Section 4.

Remark 7. The improved impedance relationship combines the PID-like method with the traditional impedance relationship in outer-loop force control to improve the response speed and the performance of force tracking. By choosing the appropriate PID parameters, the force tracking errors and the force overshoot can be reduced effectively with fast convergence when the manipulator contacts with environment. In addition, the improved impedance method can be applied to track a constant force or a twice-differentiable time-varying force.

Remark 8. Assumed that the proposed inner-loop position tracking controller is "perfect", the manipulator can work well in the free and contact spaces based on the position-based impedance control scheme [27, 34, 35]. It means that the position command X_c satisfies (31) if the forced direction is considered only. In this way, the position tracking and the force tracking can be achieved based on the two-loop separation design method in the position-based impedance control scheme.

4. Design of Inner-Loop Position Tracking Control and Stability Analysis

In this section, an adaptive position tracking controller is proposed as the inner-loop in control system to track the command position trajectory X_c generated from the outer-loop force impedance, where an adaptive Jacobian method is employed to approximate the task-space end-effector velocities and the interaction torques, and an adaptive RBFNN is designed to compensate the dynamic uncertainties of robotic systems and the uncertain Jacobian term. Based on Lyapunov theorem, the stability of the closed-loop robotic control system is then guaranteed.

4.1. Adaptive Jacobian and RBFNN Position Tracking Controller Design. Define $\theta = [\theta_1, \ldots, \theta_l]^{\mathrm{T}}$ as the parameter vector in Jacobian matrix $J(q)$; the task-space velocity of the end-effector and the robot interaction torque can be expressed as

$$\dot{X} = J(q)\dot{q} = Y(q, \dot{q})\theta \tag{34}$$

$$\tau_e = J^{\mathrm{T}}(q) F_e = Y_f(q, F_e)\theta, \tag{35}$$

where $Y(q, \dot{q}) \in \mathfrak{R}^{m\times l}$ and $Y_f(q, F_e) \in \mathfrak{R}^{n\times l}$ denote the velocity regressor matrix and the interaction torque regressor matrix, respectively. Note that the robot kinematic parameters uncertainties are always existed such as link

length and mass so that the Jacobian matrix $J(q)$ cannot be known precisely. Therefore, define the estimated Jacobian matrix $\widehat{J}(q,\widehat{\theta}) \in \mathfrak{R}^{m\times n}$, the estimation $\widehat{\dot{X}}$ of $\dot{X}$ and the estimation $\widehat{\tau}_e$ of τ_e can be represented as

$$\widehat{\dot{X}} = \widehat{J}\left(q,\widehat{\theta}\right)\dot{q} = Y\left(q,\dot{q}\right)\widehat{\theta} \tag{36}$$

$$\widehat{\tau}_e = \widehat{J}^{\mathrm{T}}\left(q,\widehat{\theta}\right)F_e = Y_f\left(q,F_e\right)\widehat{\theta}, \tag{37}$$

where $\widehat{\theta} \in \mathfrak{R}^l$ denotes the estimated parameters vector. Then, the estimated task-space velocity error $\widetilde{\dot{X}}$ and estimated interaction torque error $\widetilde{\tau}_e$ can be expressed as

$$\widetilde{\dot{X}} = \dot{X} - \widehat{\dot{X}} = Y\left(q,\dot{q}\right)\widetilde{\theta}, \tag{38}$$

$$\widetilde{\tau}_e = \tau_e - \widehat{\tau}_e = Y_f\left(q,F_e\right)\widetilde{\theta}, \tag{39}$$

where $\widetilde{\theta} = \theta - \widehat{\theta}$.

Define a vector $\nu \in \mathfrak{R}^m$ as

$$\nu = \dot{X}_c + \Lambda E_x. \tag{40}$$

where $\Lambda = \Lambda^{\mathrm{T}} > 0$ is a positive matrix and $E_x = X_c - X$ is the position tracking error of the end-effector. Then, differentiating (40) with respect to time yields

$$\dot{\nu} = \ddot{X}_c + \Lambda\dot{E}_x. \tag{41}$$

where $\dot{E}_x$ is the velocity tracking error of the end-effector.

Define a filtered tracking error as

$$\widehat{r}_x = \widehat{\dot{E}}_x + \Lambda E_x, \tag{42}$$

where $\widehat{\dot{E}}_x = \dot{X}_c - \widehat{\dot{X}}$ is the estimated value of the velocity tracking error $\dot{E}_x$.

Then, according to (36) and (40), we have

$$\widehat{r}_x = \nu - \widehat{\dot{X}} = \nu - \widehat{J}\left(q,\widehat{\theta}\right)\dot{q}. \tag{43}$$

Differentiating (43), we obtain

$$\dot{\widehat{r}}_x = \dot{\nu} - \widehat{\ddot{X}} = \dot{\nu} - \dot{\widehat{J}}\left(q,\widehat{\theta}\right)\dot{q} - \widehat{J}\left(q,\widehat{\theta}\right)\ddot{q}. \tag{44}$$

where $\widehat{\ddot{E}}_x$ and $\widehat{\ddot{X}}$ denote the derivative of $\widehat{\dot{E}}_x$ and $\widehat{\dot{X}}$, respectively.

Next, define a filtered tracking error in joint space as

$$r_q = \widehat{J}^{+}\left(q,\widehat{\theta}\right)\widehat{r}_x. \tag{45}$$

Assumed that the manipulator works in a finite task space, substituting (43) into (45) yields

$$r_q = \widehat{J}^{+}\left(q,\widehat{\theta}\right)\left(\nu - \widehat{J}\left(q,\widehat{\theta}\right)\dot{q}\right) = \widehat{J}^{+}\left(q,\widehat{\theta}\right)\nu - \dot{q}, \tag{46}$$

where $\widehat{J}^{+}(q,\widehat{\theta})$ is the inverse of the estimated Jacobian matrix $\widehat{J}(q,\widehat{\theta})$, which is similarly defined as $J^{+}(q)$ in Assumption 5. Let a virtual joint velocity $\dot{q}_\nu$ be

$$\dot{q}_\nu = \widehat{J}^{+}\left(q,\widehat{\theta}\right)\nu, \tag{47}$$

Substituting (47) into (46), we obtain

$$r_q = \dot{q}_\nu - \dot{q}, \tag{48}$$

and

$$\dot{r}_q = \ddot{q}_\nu - \ddot{q}. \tag{49}$$

Then, taking the derivative of (47) with respect to time, we obtain

$$\ddot{q}_\nu = \widehat{J}^{+}\left(q,\widehat{\theta}\right)\dot{\nu} - \widehat{J}^{+}\left(q,\widehat{\theta}\right)\dot{\widehat{J}}\left(q,\widehat{\theta}\right)\widehat{J}^{+}\left(q,\widehat{\theta}\right)\nu. \tag{50}$$

Considering the uncertain $\dot{E}_x$, (41) can be redefined as

$$\widehat{\dot{\nu}} = \ddot{X}_c + \Lambda\widehat{\dot{E}}_x. \tag{51}$$

Then, (50) can also be redefined as

$$\widehat{\ddot{q}}_\nu = \widehat{J}^{+}\left(q,\widehat{\theta}\right)\widehat{\dot{\nu}} - \widehat{J}^{+}\left(q,\widehat{\theta}\right)\dot{\widehat{J}}\left(q,\widehat{\theta}\right)\widehat{J}^{+}\left(q,\widehat{\theta}\right)\nu. \tag{52}$$

Substituting (50) into (52), according to (34), (36), (41), and (51), yields

$$\begin{aligned}\widehat{\ddot{q}}_\nu &= \ddot{q}_\nu - \widehat{J}^{+}\left(q,\widehat{\theta}\right)\left(\dot{\nu} - \widehat{\dot{\nu}}\right)\\ &= \ddot{q}_\nu - \Lambda\widehat{J}^{+}\left(q,\widehat{\theta}\right)\left(\dot{E}_x - \widehat{\dot{E}}_x\right)\\ &= \ddot{q}_\nu - \Lambda\widehat{J}^{+}\left(q,\widehat{\theta}\right)\left(\dot{X} - \widehat{\dot{X}}\right)\\ &= \ddot{q}_\nu - \Lambda\dot{q} + \Lambda\widehat{J}^{+}\left(q,\widehat{\theta}\right)J\left(q\right)\dot{q}.\end{aligned} \tag{53}$$

Substituting (53) into (49) yields

$$\dot{r}_q = \widehat{\ddot{q}}_\nu + \Lambda\dot{q} - \Lambda\widehat{J}^{+}\left(q,\widehat{\theta}\right)J\left(q\right)\dot{q} - \ddot{q}. \tag{54}$$

Define that the states are $x_1 = q$ and $x_2 = \dot{q}$ and then multiply $M(x_1)$ with (54) on both sides; we have

$$\begin{aligned}&M\left(x_1\right)\dot{r}_q\\ &= M\left(x_1\right)\left(\widehat{\ddot{q}}_\nu + \Lambda x_2 - \Lambda\widehat{J}^{+}\left(x_1,\widehat{\theta}\right)J\left(x_1\right)x_2 - \dot{x}_2\right)\\ &= M\left(x_1\right)\left(\widehat{\ddot{q}}_\nu + \Lambda x_2 - \Lambda\widehat{J}^{+}\left(x_1,\widehat{\theta}\right)J\left(x_1\right)x_2\right)\\ &\quad - M\left(x_1\right)\dot{x}_2\\ &= M\left(x_1\right)\left(\widehat{\ddot{q}}_\nu + \Lambda x_2 - \Lambda\widehat{J}^{+}\left(x_1,\widehat{\theta}\right)J\left(x_1\right)x_2\right)\\ &\quad - \left[\tau - \tau_e - C\left(x_1,x_2\right)x_2 - G\left(x_1\right) - \tau_f\left(x_2\right) - \tau_d\right]\\ &= -\tau + \tau_e - C\left(x_1,x_2\right)r_q + \tau_d + H\left(\chi\right),\end{aligned} \tag{55}$$

where $\chi = (x_1^{\mathrm{T}}, x_2^{\mathrm{T}}, \dot{q}_\nu^{\mathrm{T}}, \widehat{\ddot{q}}_\nu^{\mathrm{T}}, \widehat{\theta}^{\mathrm{T}})^{\mathrm{T}}$, and

$$\begin{aligned}H\left(\chi\right) = {}& M\left(x_1\right)\left(\widehat{\ddot{q}}_\nu + \Lambda x_2 - \Lambda\widehat{J}^{+}\left(x_1,\widehat{\theta}\right)J\left(x_1\right)x_2\right)\\ &+ C\left(x_1,x_2\right)\dot{q}_\nu + G\left(x_1\right) + \tau_f\left(x_2\right),\end{aligned} \tag{56}$$

Remark 9. In general, the precise values of the robotic matrices $M(x_1), C(x_1, x2), G(x_1)$, and $\tau_f(x_2)$ are difficult to acquire but bounded [38]. Moreover, the Jacobian matrix $J(x_1)$ is also bounded, it can be concluded that $J^+(x_1)$ is bounded according to Assumption 6. Therefore, the unknown nonlinear function $H(\chi)$ in (56) is bounded and can be approximated by using RBFNN.

In this paper, the unknown function $H(\chi)$ is approximated by using RBFNN,

$$H(\chi) = W^{*\mathrm{T}}\phi(\chi) + \varepsilon(\chi), \tag{57}$$

where $\phi(\chi)$ denotes the activation function of RBFNN, W^* represents the ideal weight matrix, and $\varepsilon(\chi)$ denotes the minimum reconstructed error vector. Then, the adaptive RBFNN position trajectory tracking control law can be given as

$$\tau = -\widehat{J}^{\mathrm{T}}\left(x_1, \widehat{\theta}\right)\left(L_p E_x + L_v \widehat{\dot{E}}_x + F_e\right) + \widehat{W}^{\mathrm{T}}\phi(\chi) + u_{\mathrm{rss}}, \tag{58}$$

where $L_p > 0$ and $L_v > 0$ are the controller position and velocity gain matrices, respectively, and u_{rss} denotes the robust compensation term which is used to compensate the approximation error of RBFNN and external disturbances. Substituting the control law (58) into (55), the closed-loop error equation of the robotic system is as follows:

$$M(x_1)\dot{r}_q = \left[-\widehat{J}^{\mathrm{T}}\left(x_1, \widehat{\theta}\right)\left(L_p E_x + L_v \widehat{\dot{E}}_x + F_e\right) - \widehat{W}^{\mathrm{T}}\phi(\chi) - u_{\mathrm{rss}}\right] + \tau_e - C(x1, x2)\, r_q + \tau_d + H(\chi). \tag{59}$$

So

$$H(\chi) - \widehat{W}^{\mathrm{T}}\phi(\chi) = W^{*\mathrm{T}}\phi(\chi) + \varepsilon(\chi) - \widehat{W}^{\mathrm{T}}\phi(\chi) = -\widetilde{W}^{\mathrm{T}}\phi(\chi) + \varepsilon(\chi), \tag{60}$$

where $\widetilde{W} = W^* - \widehat{W}$ denotes the weight estimated error.

Then, substituting (39) and (60) into (59) results in

$$M(x_1)\dot{r}_q = -\widehat{J}^{\mathrm{T}}\left(x_1, \widehat{\theta}\right)\left(L_p E_x + L_v \widehat{\dot{E}}_x\right) - C_0(x_1, x_2)\, r_q + \widetilde{\tau}_e - \widetilde{W}^{\mathrm{T}}\phi(\widehat{\chi}) - \left[u_{\mathrm{rss}} - \varepsilon(\chi) - \tau_d\right]. \tag{61}$$

According to Assumptions 1 and 5, the RBFNN modeling errors are bounded as

$$\left\|\varepsilon(\chi) + \tau_d\right\| \le \delta_\varepsilon + \tau_D \triangleq \rho. \tag{62}$$

4.2. Stability Analysis

Theorem 10. *According to the robotic dynamic model (7), assume that Assumptions 1–6 are all satisfied, the adaptive RBFNN position tracking control law is designed as (58), where the robust compensation term can be given by*

$$u_{\mathrm{rss}} = \begin{cases} \rho\dfrac{r_q}{\left\|r_q\right\|}, & \text{if } \left\|r_q\right\| \neq 0 \\ 0, & \text{if } \left\|r_q\right\| = 0. \end{cases} \tag{63}$$

By the projection algorithm, the adaptive updating law for the Jacobian matrix parameter $\widehat{\theta}$ and the weight matrix $\widehat{W}$ of RBFNN are designed as follows:

$$\dot{\widehat{\theta}} = -\Gamma_\theta\left[Y^{\mathrm{T}}(q, \dot{q})\left(L_p + \Lambda L_v\right)E_x - {Y_f}^{\mathrm{T}}(q, F_e)\, r_q\right], \tag{64}$$

$$\dot{\widehat{W}} = -\Gamma_W\phi(\widehat{\chi})\, r_q, \tag{65}$$

where Γ_θ and Γ_W are both the positive matrices.

Then, the filter error r_q, the task-space position error E_x, the adaptive Jacobian matrix parameter error $\widetilde{\theta}$, and the adaptive RBFNN weight matrix error $\widetilde{W}$ are all bounded and the contact force F_e can converge to desired force F_d as $t \longrightarrow \infty$.

Proof. Choose a Lyapunov function candidate as

$$V = \frac{1}{2}{r_q}^{\mathrm{T}}M(x_1)\, r_q + \frac{1}{2}E_x^{\mathrm{T}}\left(L_p + \Lambda L_v\right)E_x + \frac{1}{2}\mathrm{tr}\left\{\widetilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\widetilde{\theta}\right\} + \frac{1}{2}\mathrm{tr}\left\{\widetilde{W}^{\mathrm{T}}\Gamma_W^{-1}\widetilde{W}\right\}. \tag{66}$$

Differentiating (66) with respect to time and substituting (61) yield

$$\begin{aligned}\dot{V} &= {r_q}^{\mathrm{T}}M(x_1)\dot{r}_q + \frac{1}{2}{r_q}^{\mathrm{T}}\dot{M}(x_1)\, r_q + {E_x}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)\dot{E}_x + \mathrm{tr}\left\{\widetilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\widetilde{\theta}}\right\} + \mathrm{tr}\left\{\widetilde{W}^{\mathrm{T}}\Gamma_W^{-1}\dot{\widetilde{W}}\right\} \\ &= {r_q}^{\mathrm{T}}\left\{-\widehat{J}^{\mathrm{T}}\left(x_1, \widehat{\theta}\right)\left(L_p E_x + L_v \widehat{\dot{E}}_x\right) - C(x_1, x_2)\, r_q + \widetilde{\tau}_e - \widetilde{W}^{\mathrm{T}}\phi(\chi) - \left[u_{\mathrm{rss}} - \varepsilon(\chi) - \tau_d\right]\right\} + \frac{1}{2} \\ &\quad \cdot {r_q}^{\mathrm{T}}\dot{M}(x_1)\, r_q + {E_x}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)\dot{E}_x\Big\} + \mathrm{tr}\left\{\widetilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\widetilde{\theta}}\right\} + \mathrm{tr}\left\{\widetilde{W}^{\mathrm{T}}\Gamma_W^{-1}\dot{\widetilde{W}}\right\} \\ &= {r_q}^{\mathrm{T}}\left\{-\widehat{J}^{\mathrm{T}}\left(x_1, \widehat{\theta}\right)\left(L_p E_x + L_v \widehat{\dot{E}}_x\right) + \widetilde{\tau}_e - \widetilde{W}^{\mathrm{T}}\phi(\chi) + \left[\varepsilon(\chi) + \tau_d - u_{\mathrm{rss}}\right]\right\} + \frac{1}{2}{r_q}^{\mathrm{T}}\left(\dot{M}(x_1) - 2C(x_1, x_2)\right)r_q + {E_x}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)\dot{E}_x \\ &\quad + \mathrm{tr}\left\{\widetilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\widetilde{\theta}}\right\} + \mathrm{tr}\left\{\widetilde{W}^{\mathrm{T}}\Gamma_W^{-1}\dot{\widetilde{W}}\right\}.\end{aligned} \tag{67}$$

According to the bounded modeling errors (62) and robust term (63) and the RBFNN adaptive law (65), considering the fact $\dot{\widetilde{W}} = -\dot{\widehat{W}}$, we have

$${r_q}^{\mathrm{T}}\left[\varepsilon(\chi) + \tau_d - u_{\mathrm{rss}}\right] \le 0, \tag{68}$$

and

$$-\hat{r}_q^{\mathrm{T}}\widehat{W}^{\mathrm{T}}\phi(\chi) + \mathrm{tr}\left\{\widehat{W}^{\mathrm{T}}\Gamma_W^{-1}\dot{\widehat{W}}\right\} = 0. \tag{69}$$

According to Property 3, substituting (68) and (69) into (67) yields

$$\begin{aligned}\dot{V} \le {} & r_q{}^{\mathrm{T}}\left[-\widehat{J}^{\mathrm{T}}\left(x_1,\widehat{\theta}\right)\left(L_pE_x + L_v\dot{\widehat{E}}_x\right)\right] + r_q{}^{\mathrm{T}}\tilde{\tau}_e \\ & + E_x{}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)\dot{E}_x + \mathrm{tr}\left\{\tilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\tilde{\theta}}\right\}.\end{aligned} \tag{70}$$

Substituting (38), (39), (42), and (45) into (70) yields

$$\begin{aligned}\dot{V} \le {} & -\hat{r}_x^{\mathrm{T}}L_pE_x - \hat{r}_x^{\mathrm{T}}L_v\dot{\widehat{E}}_x + r_q{}^{\mathrm{T}}\tilde{\tau}_e \\ & + E_x{}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)\dot{E}_x + \mathrm{tr}\left\{\tilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\tilde{\theta}}\right\} \\ = {} & -\dot{E}_x^{\mathrm{T}}L_pE_x - (\Lambda E_x)^{\mathrm{T}}L_pE_x - \dot{\widehat{E}}_x{}^{\mathrm{T}}L_v\dot{\widehat{E}}_x \\ & - (\Lambda E_x)^{\mathrm{T}}L_v\dot{E}_x - E_x{}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)Y(q,\dot{q})\tilde{\theta} \\ & + r_q{}^{\mathrm{T}}Y_f(q,F_e)\tilde{\theta} + E_x{}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)\dot{E}_x \\ & + \mathrm{tr}\left\{\tilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\tilde{\theta}}\right\} \\ = {} & -E_x{}^{\mathrm{T}}\left(\Lambda L_p\right)E_x - \dot{\widehat{E}}_x{}^{\mathrm{T}}L_v\dot{\widehat{E}}_x \\ & - \left[E_x{}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)Y(q,\dot{q}) - r_q{}^{\mathrm{T}}Y_f(q,F_e)\right]\tilde{\theta} \\ & + \mathrm{tr}\left\{\tilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\tilde{\theta}}\right\}.\end{aligned} \tag{71}$$

According to the adaptive updating law (64) and considering the fact $\dot{\tilde{\theta}} = -\dot{\hat{\theta}}$, we have

$$\begin{aligned}& -\left[E_x{}^{\mathrm{T}}\left(L_p + \Lambda L_v\right)Y(q,\dot{q}) - r_q{}^{\mathrm{T}}Y_f(q,F_e)\right]\tilde{\theta} \\ & \quad + \mathrm{tr}\left\{\tilde{\theta}^{\mathrm{T}}\Gamma_\theta^{-1}\dot{\tilde{\theta}}\right\} = 0.\end{aligned} \tag{72}$$

Substituting (72) into (71) yields

$$\dot{V} \le -E_x{}^{\mathrm{T}}\left(\Lambda L_p\right)E_x - \dot{\widehat{E}}_x{}^{\mathrm{T}}L_v\dot{\widehat{E}}_x \le 0. \tag{73}$$

According to (66) and (73), it can be concluded that the error signals r_q, E_x, $\tilde{\theta}$, and $\widetilde{W}$ are all bounded. That means $\hat{\theta}$ and $\widehat{W}$ are both bounded, and $\hat{r}_x = \widehat{J}(q,\hat{\theta})r_q$ is also bounded using (45). From (39), τ_e is bounded since the contact force F_e is bounded. Then, according to (40) and (42), it can be concluded that $\dot{\widehat{E}}_x$ and ν are bounded as the position controller input X_c and $\dot{X}_c$ is bounded, and $\dot{q}_\nu$ is also bounded according to (47) as the estimate Jacobian matrix $\widehat{J}(q,\theta)$ is nonsingular. Therefore, $\dot{q}$ is bounded according to (46) which implies that $\dot{X}$ is bounded. Next, it can be concluded that $\dot{E}_x = \dot{X}_c - \dot{X}$ is bounded and $\dot{\nu}$ is also bounded from (41) as $\ddot{X}_c$ is bounded. And $\widehat{\dot{\nu}}, \ddot{q}_\nu$ are both bounded according to (51) and (52). In addition, according to Property 4, it can be concluded that $\dot{r}_q$ is bounded from (61) so that $\ddot{q}$ is also bounded from (49), and then $\ddot{X}$ is bounded, which means $\ddot{E}_x = \ddot{X}_c - \ddot{X}$ is bounded as $\ddot{X}_c$ is bounded. Considering (44), $\dot{\hat{r}}_x$ is bounded, which means $\ddot{\widehat{E}}_x$ is also bounded.

Differentiating (73) with respect to time, we have

$$\ddot{V} = -2E_x{}^{\mathrm{T}}\left(\Lambda L_p\right)\dot{E}_x - \dot{\widehat{E}}_x{}^{\mathrm{T}}L_v\ddot{\widehat{E}}_x \tag{74}$$

where $\ddot{\widehat{E}}_x$ denotes the derivative of $\dot{\widehat{E}}_x$. Since the error signals E_x, $\dot{E}_x$, $\dot{\widehat{E}}_x$, $\ddot{\widehat{E}}_x$ are all bounded, $\ddot{V}$ is uniformly continuous. According to **Barbalat's lemma** [39], we can conclude $E_x = X_c - X \longrightarrow 0$ and $\dot{\widehat{E}}_x = \dot{X}_c - \dot{\widehat{X}} \longrightarrow 0$ as $t \longrightarrow \infty$. Since $\ddot{E}_x$ is bounded, we can obtain $\dot{E} \longrightarrow 0$ as $t \longrightarrow \infty$. That is,

$$\begin{aligned} & X \longrightarrow X_c \\ \text{and } & \dot{X} \longrightarrow \dot{X}_c \\ & \text{as } t \longrightarrow \infty. \end{aligned} \tag{75}$$

Based on the position-based impedance control scheme proposed in Section 3, we can conclude

$$F_e \longrightarrow F_d \quad \text{as } t \longrightarrow \infty \tag{76}$$

when the manipulator contacts with environment. □

According to the above analysis, the block diagram of the whole closed-loop impedance control system is shown as Figure 1.

Remark 11. The position-based impedance control has a superior design feature that the pure position tracking control is designed separately as the inner-loop of the robotic control system. Based on the position trajectory command X_c generated from the outer-loop force impedance control (designed in Section 3), it allows the manipulator only uses a position trajectory tracking to achieve the desired contact force tracking and track the command position in orthogonal. If there is no robot–environment interaction, i.e., $f_e \equiv 0$, the objective of impedance control is equivalent to the objective of unconstrained motion control in the task space. This design structure is more concise and convenient and easy to implement when the robotic system is complex and uncertainty exists.

Remark 12. In this paper, since the position trajectory command X_c is as the input of the inner-loop position trajectory tracking in Cartesian space rather than in joint space, the proposed task-space control scheme can effectively avoid the robot inverse kinematic solution so that the robustness of robotic system can be then improved.

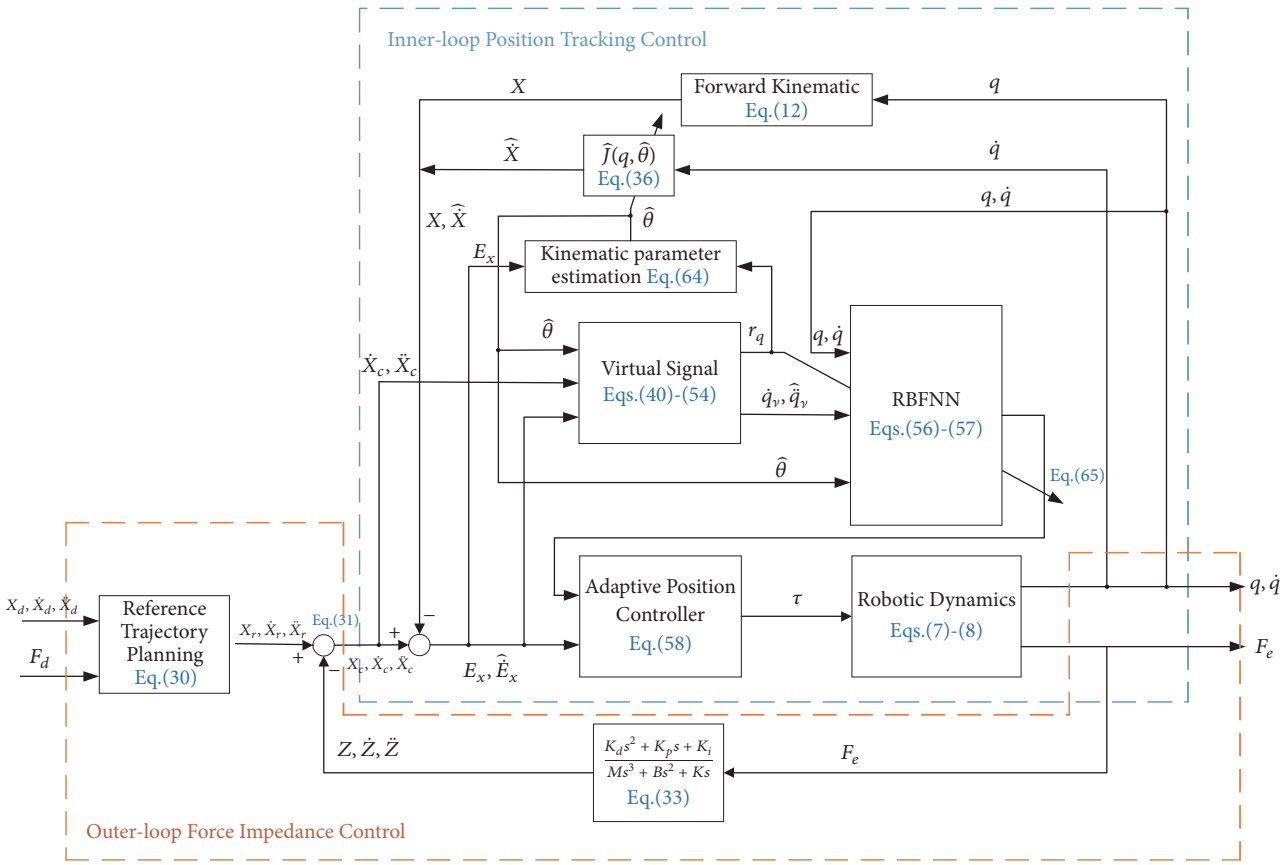


FIGURE 1: The proposed closed-loop impedance control system.

5. Simulation Examples

To verify the theoretical results, simulations were conducted on a two-DOF robotic manipulator, which is described as,

$$M(q) = \begin{bmatrix} (m_1+m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2c_2 & m_2\left(l_2^2 + l_1l_2c_2\right) \\ m_2\left(l_2^2 + l_1l_2c_2\right) & m_2(l_2^2 \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} -2m_2l_1l_2s_2\dot{q}_2 & m_2l_1l_2s_2\dot{q}_2 \\ m_2l_1l_2s_2\dot{q}_2 & 0 \end{bmatrix} \quad (77)$$

$$G(q) = \begin{bmatrix} (m_1+m_2)l_1gc_1 + m_2l_2gc_{12} \\ m_2l_2gc_{12} \end{bmatrix},$$

the Jacobian matrix is be given

$$J(q) = \begin{bmatrix} -l_1s_1 - l_2s_{12} & -l_2s_{12} \\ l_1c_1 + l_2c_{12} & l_2c_{12} \end{bmatrix} \quad (78)$$

where l_1 and l_2 are the length of link 1 and link 2, respectively; m_1 and m_2 are the mass of link 1 and link 2, respectively; s_i denotes $\sin(q_i)$; c_i represents $\cos(q_i)$; s_{ij} represents $\sin(q_i+q_j)$, for $i, j = 1, 2$; c_{ij} denotes $\cos(q_i + q_j)$, for $i, j = 1, 2$; g is acceleration of gravity.

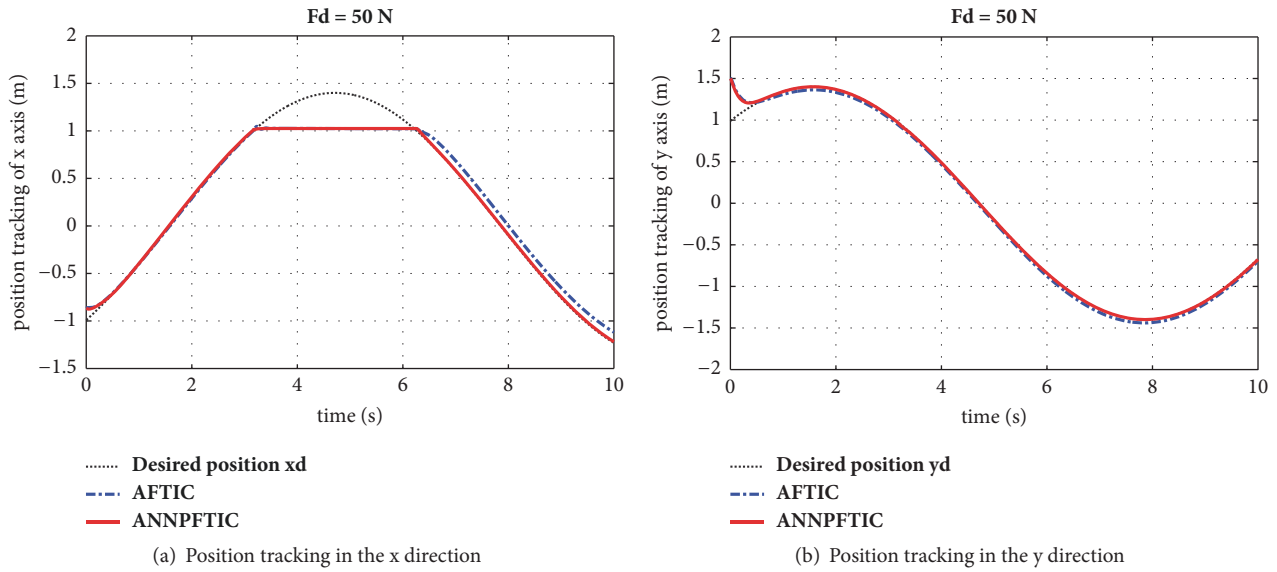
5.1. Design Procedure. To summarize the analysis in Sections 3 and 4, the step-by-step procedures of the adaptive RBFNN impedance force/position tracking control based on impedance control for robotic system are outlined as follows:

Step 1. Select the environmental stiffness and the damping parameters $K_e = 2000I_{2\times2}$ and $B_e = 5I_{2\times2}$ and the environmental position $x_e = 1$m.

Step 2. Select parameters of the improved impedance relationship $M = I_{2\times2}$, $B = 70I_{2\times2}$, $K = 330I_{2\times2}$, $K_p = 8I_{2\times2}$, $K_i = 0.025I_{2\times2}$, and $K_d = 0.5I_{2\times2}$, then the position-based impedance control outer-loop can be obtained based on Section 3.

Step 3. The initial values of the estimated parameters vector are selected as $\theta(0) = [1, 1]^T$ and adaptive learning parameter is selected as $\Gamma_\theta = 0.01I_{2\times2}$ in (64).

Step 4. Construct the RBFNN; it has 10 input nodes, 20 hidden layer nodes, and 2 output nodes, and its inputs are $\chi = (x_1, x_2, \dot{q}_v, \hat{\dot{q}}_v, \hat{\theta})^T$. The center and width of Gaussian function (2) are selected to be fixed which the centers are chosen as 20 points of uniform distribution in $[-5, 5]$ and the width $\delta_j = 5$, and the initial values of the adjustable weights are selected as $W(0) = 0$ and the learning parameter is selected as $\Gamma_W = 0.2I_{20\times20}$ in (65).



(a) Position tracking in the x direction

(b) Position tracking in the y direction

FIGURE 2: Position tracking for Case 1.

Step 5. Choose controller gain $L_p = 1600I_{2\times2}$, $L_v = 400I_{2\times2}$ in (58) and $\Lambda = 5I_{2\times2}$ in (40)-(44), and set $\rho = 20$, then the adaptive RBF neural network position tracking controller can be obtained from Theorem 10.

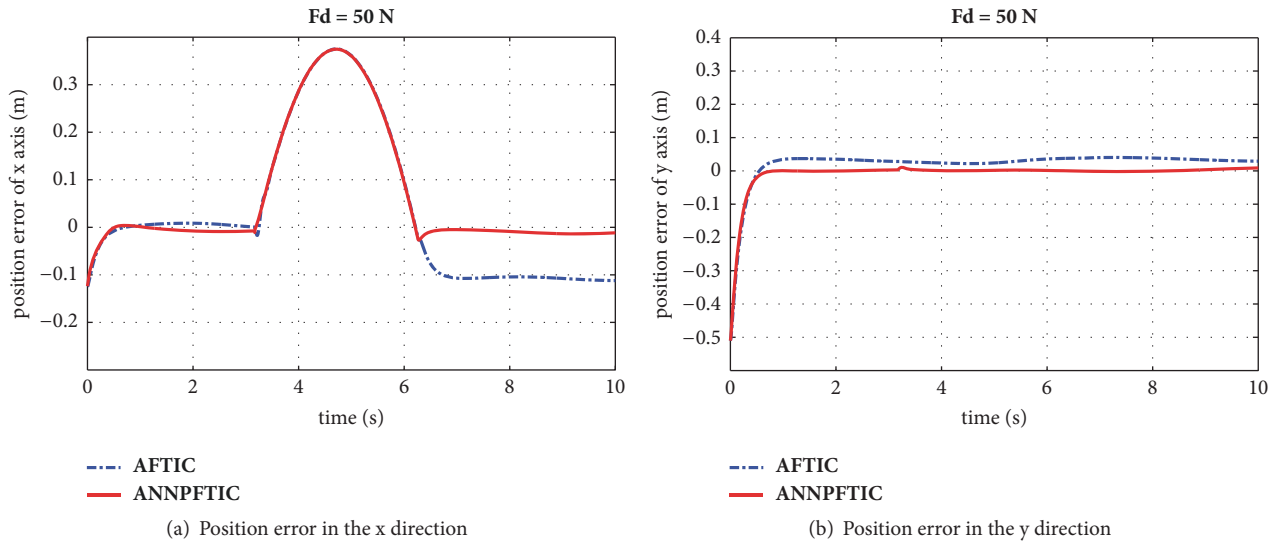
5.2. Simulation Results. In this section, the proposed approach is applied to control the two-DOF robotic manipulator. The parameters of robot for simulation are chosen as $m_1 = 1.6$ kg, $m_2 = 3.3$ kg and $l_1 = 1.1$ m, $l_2 = 0.9$ m, $g = 9.8\text{m/s}^2$, while actual parameters of robot are $m_1 = 2$ kg, $m_2 = 3$ kg and $l_1 = 1$ m, $l_2 = 1$ m to introduce the parameters uncertainties. The initial conditions are $q_1(0) = \pi/2$, $q_2(0) = \pi/3$ rad, and $\dot{q}_1(0) = \dot{q}_2(0) = 0$ rad/s. The desired position trajectory of end-effector is $X_d(t) = [1.4\sin(0.5t - \pi/4), 1.4\sin(0.5t + \pi/4)]^{\mathrm{T}}$m and the desired velocity is $\dot{X}_d(t) = [0.7\cos(0.5t - \pi/4), 0.7\cos(0.5t + \pi/4)]^{\mathrm{T}}$m/s. In order to compare the constant force and the time-vary force, the desired force input included two cases, the constant force $F_d = [50, 0]^{\mathrm{T}}$N and the time-varying force $F_d = [50 + 20\sin(2t), 0]^{\mathrm{T}}$N. The simulation time is 10s and the sample time is 0.001s.

For the purpose of comparison, simulation studies in two cases are conducted. To show the robustness of the adaptive RBFNN impedance controller, we choose the friction terms $F_n = [3\dot{q}_1 + 0.8\,\mathrm{sign}\,(3\dot{q}_1), \dot{q}_2 + 1.1\,\mathrm{sign}\,(2\dot{q}_2)]^{\mathrm{T}}$ and the external disturbances $\tau_d = [-3\cos(2t), 3\sin(t)]^{\mathrm{T}}$.

Case 1. Assume that the desired force is constant $F_d = [50, 0]^{\mathrm{T}}$N and the contact force exserted on the end-effector occurred as $x \geq x_e = 1$ in x direction. The adaptive force tracking impedance control (AFTIC) in [14] is compared with the adaptive neural network position/force tracking impedance control (ANNPFTIC) proposed in this paper. Figures 2–7 show the results, where Figures 2(a) and 2(b) are the desired and actual positions of end-effector in x and y directions of Cartesian space, respectively, Figures 3(a) and 3(b) are the position errors between the desired and actual positions in x and y directions of Cartesian space, respectively, Figures 4(a) and 4(b) are the desired and actual velocities of end-effector in x and y directions of Cartesian space, respectively, Figures 5(a) and 5(b) are the velocity errors in x and y directions of Cartesian space, respectively, Figure 5(a) is the force exerted on the environment at end-effector in x direction, Figure 6(b) is the force error in x direction, and Figures 7(a) and 7(b) are the end-effector position trajectory and joint angular position, respectively.

Case 2. Assume that the desired force is time-varying force $F_d = [50 + 20\sin(2t), 0]^{\mathrm{T}}$N. Similar to Case 1, Figures 8–13 show the results, where Figures 8(a) and 8(b) are the desired and actual positions of end-effector in x and y directions of Cartesian space, respectively, Figures 9(a) and 9(b) are the position errors of end-effector in x and y directions of Cartesian space, respectively, Figures 10(a) and 10(b) are the desired and actual velocities of end-effector in x and y directions of Cartesian space, respectively, Figures 11(a) and 11(b) are the velocity errors of end-effector in x and y directions of Cartesian space, respectively, Figure 10(a) is the force exerted on the environment of end-effector in the x direction, Figure 12(b) is the force error in the x direction, and Figures 13(a) and 13(b) are the end-effector position trajectory and joint angular position, respectively.

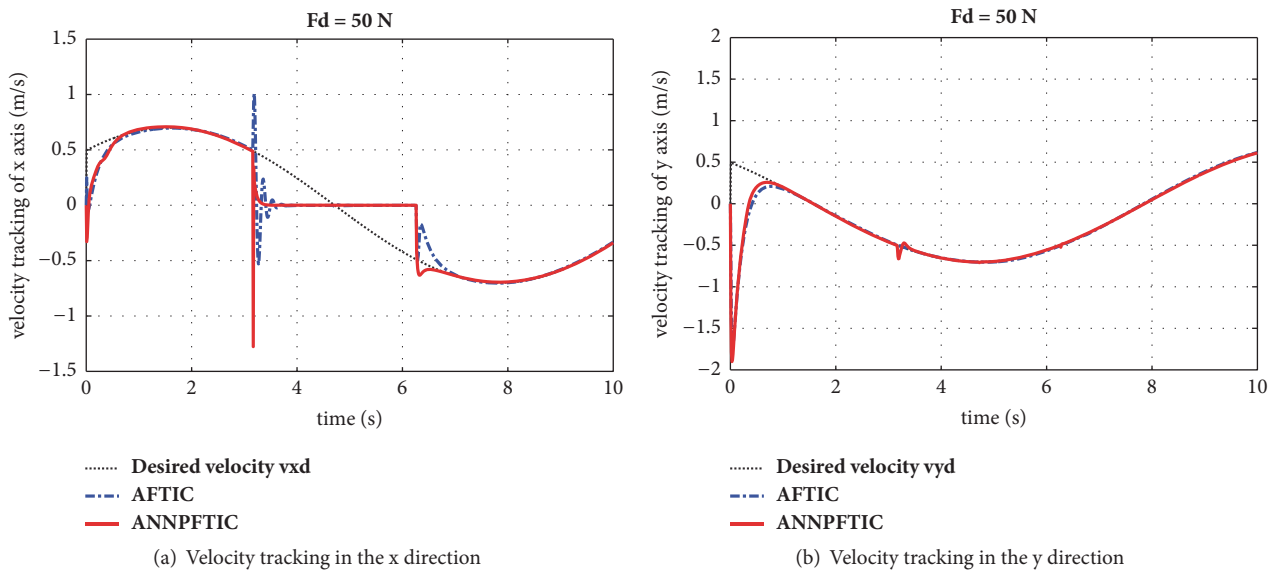
In the simulation results in both Cases 1 and 2, we can see that the manipulator can work well both in free space and in contact space. When the manipulator is in free space, the position tracking (Figures 1 and 7) and velocity tracking (Figures 3 and 9) can be achieved both in x and y direction. When the manipulator is in contact space, the force tracking (Figures 5 and 11) can be achieved in x direction and the position tracking (Figures 1 and 7) and velocity tracking (Figures 3



(a) Position error in the x direction

(b) Position error in the y direction

Figure 3: Position tracking errors for Case 1.



(a) Velocity tracking in the x direction

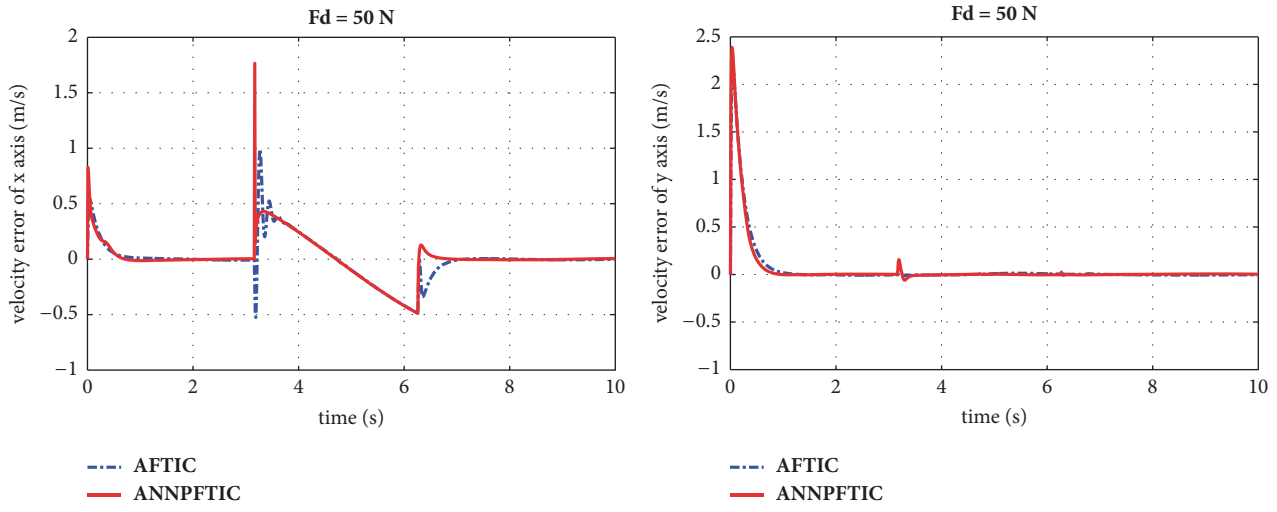
(b) Velocity tracking in the y direction

Figure 4: Velocity tracking for Case 1.

and 9) can be obtained in y direction (x direction is the forced direction). From the simulation results, one can see that the position/force tracking errors of ANNPFTIC are smaller than those of AFTIC, where the ANNPFTIC can effectively reduce the force overshoot with fast response by choosing the appropriate PID-like parameters in improved impedance relationship. It means that the proposed ANNPFTIC can achieve better performance than AFTIC and can be applied on position/force controlling of robotic systems with uncertainties and disturbances efficiently.

6. Conclusions

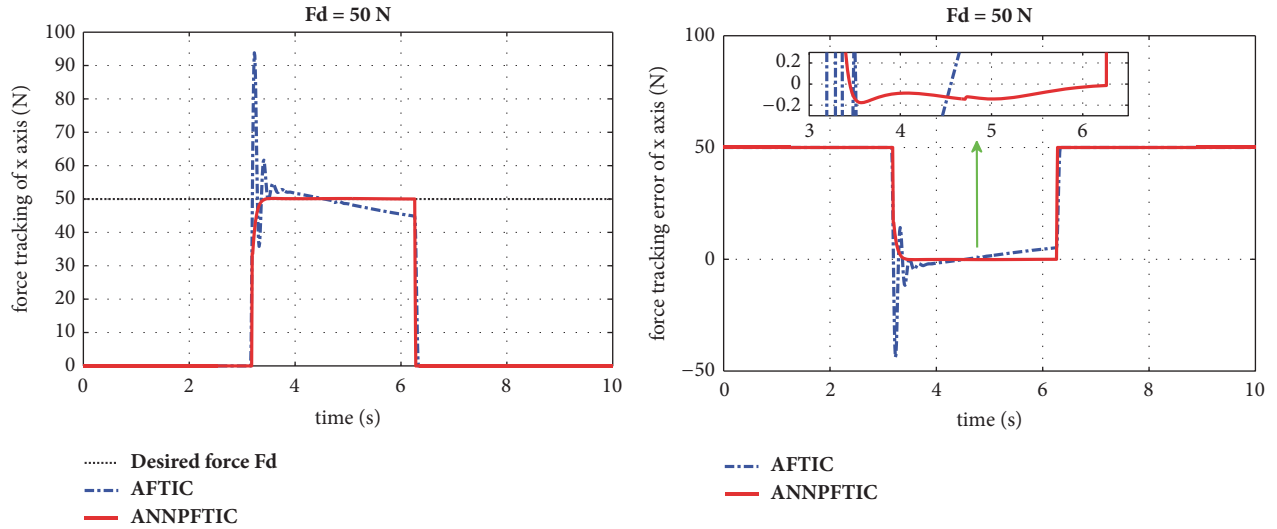
In this paper, an adaptive neural network position/force tracking impedance control scheme is proposed for controlling robotic systems with uncertainties and external disturbances, where the robot can work in both free space and contact space. The control strategy is divided into the outer-loop force tracking impedance control and the inner-loop position tracking control. In the outer-loop, a novel impedance relationship based on the PID-like scheme is proposed to improve the force tracking performance and to reduce the force overshoot, so that the good force tracking performance can be achieved when the manipulator contacts with environment. Next, according to the command position trajectory generated from the outer-loop force impedance control, an adaptive Jacobian and an adaptive RBFNN methods are designed in the inner-loop position tracking control to compensate the robotic system uncertainties so that the position and contact force tracking accuracy can be then improved, and a robust term is used to compensate the approximation error of RBFNN and the uncertain Jacobian



(a) Velocity error in the x direction

(b) Velocity error in the y direction

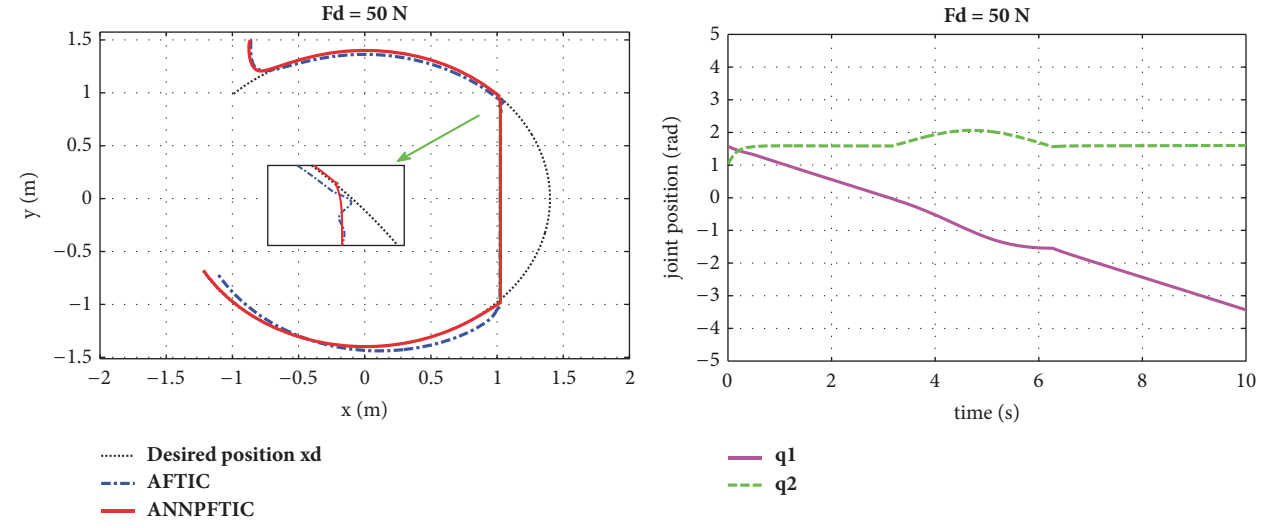
Figure 5: Velocity tracking errors for Case 1.



(a) Force tracking in the x direction

(b) Force error in the y direction

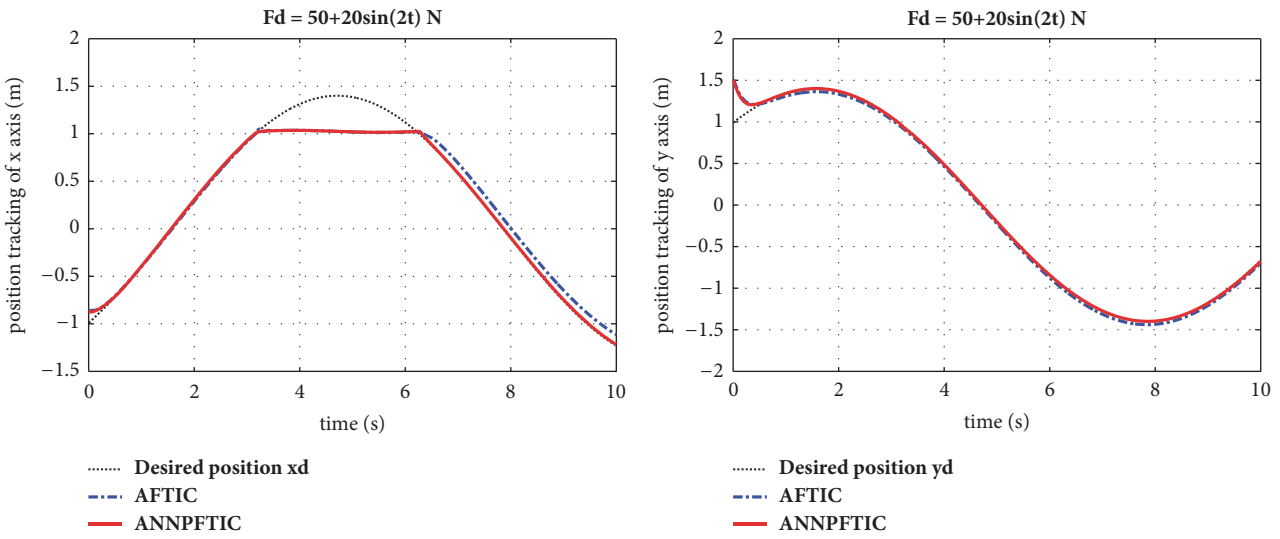
Figure 6: Force tracking and error for Case 1.



(a) Position trajectory of end-effector

(b) Joint positions of manipulator

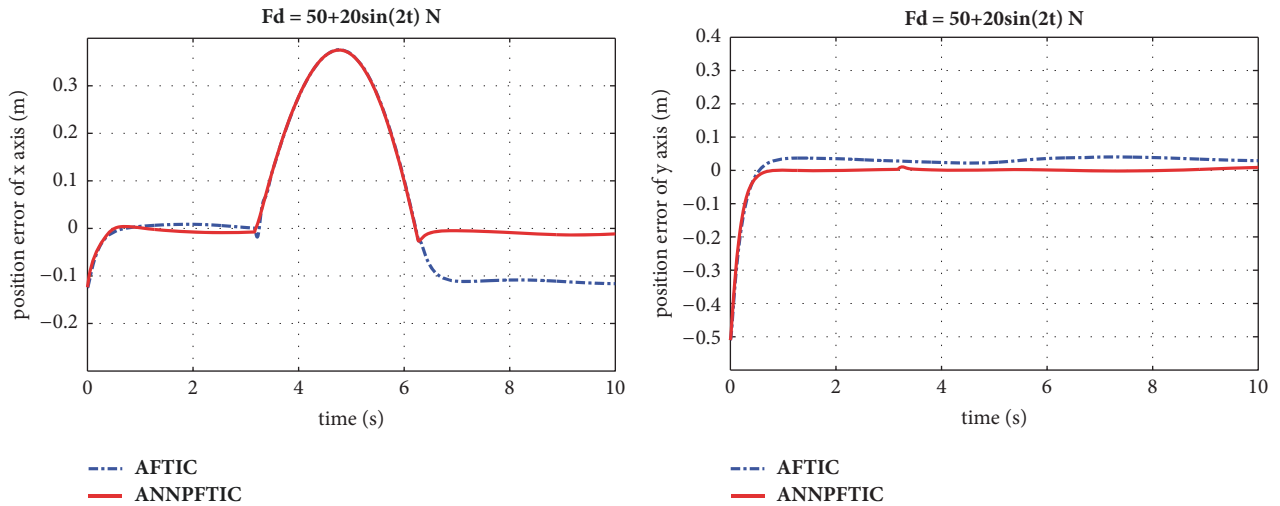
Figure 7: Position trajectory tracking and joint positions for Case 1.



(a) Position tracking in the x direction

(b) Position tracking in the y direction

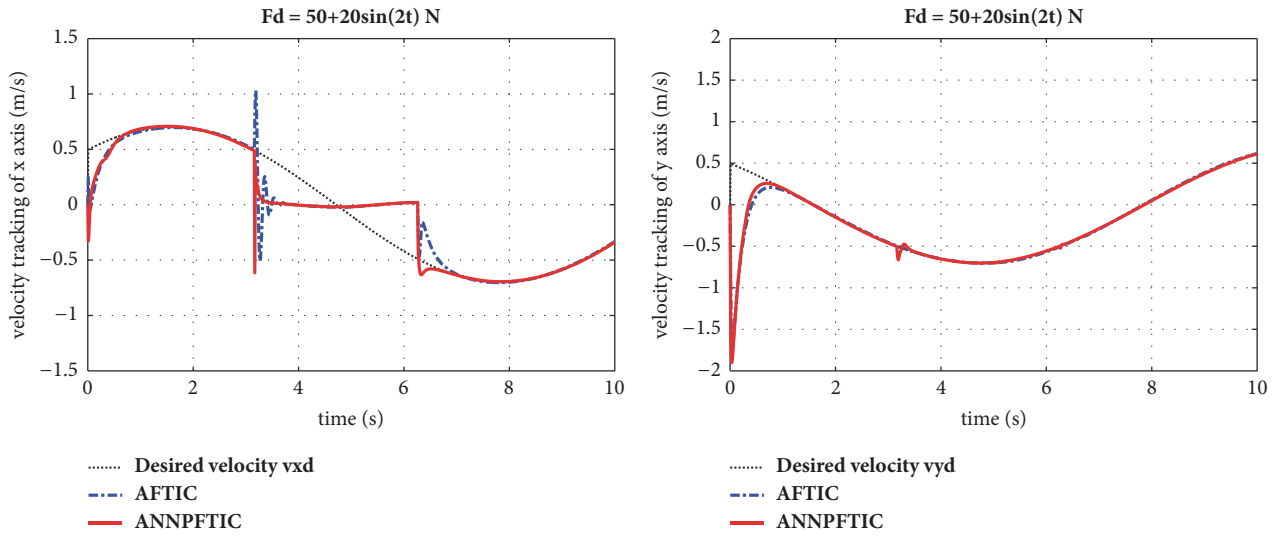
Figure 8: Position tracking for Case 2.



(a) Position error in the x direction

(b) Position error in the y direction

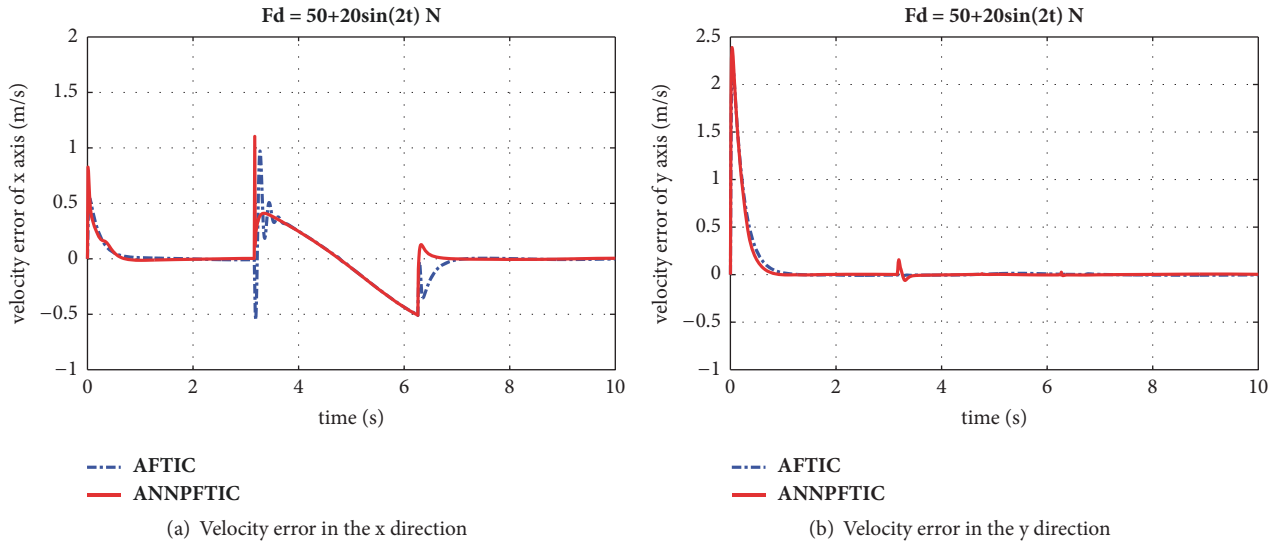
Figure 9: Position tracking errors for Case 2.



(a) Velocity in the x direction

(b) Velocity in the x direction

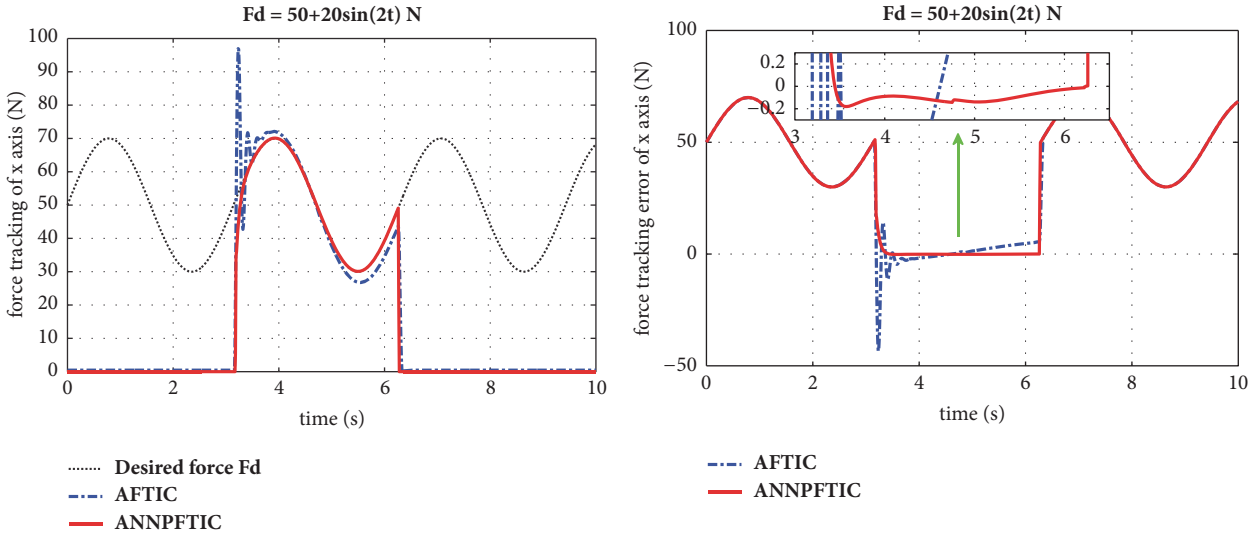
Figure 10: Velocity tracking for Case 2.



(a) Velocity error in the x direction

(b) Velocity error in the y direction

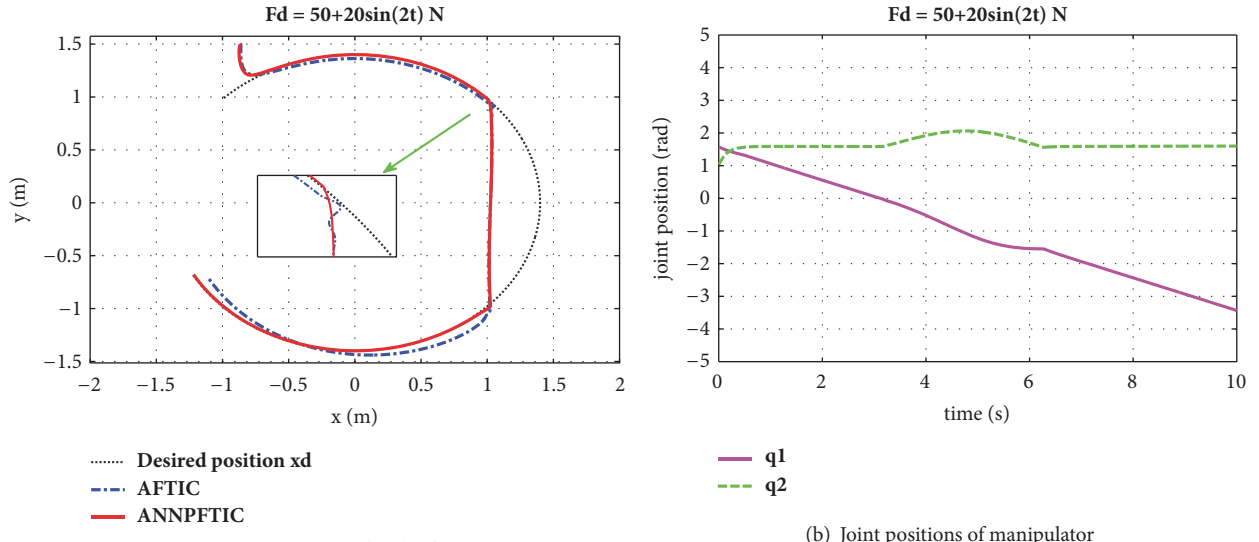
Figure 11: Velocity tracking errors for Case 2.



(a) Force tracking in the x axis

(b) Force error in the x axis

Figure 12: Force tracking and error for Case 2.



(a) Position trajectory of end-effector

(b) Joint positions of manipulator

Figure 13: Position trajectory tracking and joint positions for Case 2.

term. Finally, the validity of the control scheme is shown by computer simulation on a two-DOF robotic manipulator.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (61773351), the Natural Science Foundation of Henan Province (162300410260), the Outstanding Young Teacher Development Fund of Zhengzhou University (1521319025), the Training Plan for University's Young Backbone Teachers of Henan Province (2017GGJS004), the Key Scientific Research Projects of Universities in Henan Province (19A413002), and the Science and Technology Innovation Research Team Support Plan of Henan Province (17IRTSTHN013).

References

[1] A. G. Ak, G. Cansever, and A. Delibaşi, "Trajectory tracking control of an industrial robot manipulator using fuzzy SMC with RBFNN," *Gazi University Journal of Science*, vol. 28, no. 1, pp. 141–148, 2015.

[2] J. Peng, J. Yu, and J. Wang, "Robust adaptive tracking control for nonholonomic mobile manipulator with uncertainties," *ISA Transactions*, vol. 53, no. 4, pp. 1035–1043, 2014.

[3] J. Xin, R. R. Negenborn, and X. Lin, "Piecewise affine approximations for quality modeling and control of perishable foods," *Optimal Control Applications and Methods*, vol. 39, no. 2, pp. 860–872, 2018.

[4] A. O. Oke and A. Afolabi, "Development of a robotic arm for dangerous object disposal," in *Proceedings of the 2014 6th International Conference on Computer Science and Information Technology, CSIT 2014*, pp. 153–160, Ogbomoso, Nigeria, March 2014.

[5] F. Ficuciello, L. Villani, and B. Siciliano, "Variable Impedance Control of Redundant Manipulators for Intuitive Human-Robot Physical Interaction," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 850–863, 2015.

[6] Z. Li, C. L. Chen, and W. He, "Guest editorial for special issue on human-centered intelligent robots: issues and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 599–601, 2017.

[7] D. Herrera, F. Roberti, M. Toibero, and R. Carelli, "Human interaction dynamics for its use in mobile robotics: Impedance control for leader-follower formation," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 696–703, 2017.

[8] H. Jianbin, X. Zongwu, J. Minghe, J. Zainan, and L. Hong, "Adaptive Impedance-controlled Manipulator Based on Collision Detection," *Chinese Journal of Aeronautics*, vol. 22, no. 1, pp. 105–112, 2009.

[9] M. H. Raibert and J. J. Craig, "Hybrid posiition/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 1981.

[10] N. Hogan, "Impedance Control - An approach to manipulation. I - Theory. II - Implementation. III - Applications," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 304–313, 1985.

[11] C. D. Takahashi, R. A. Scheidt, and D. J. Reinkensmeyer, "Impedance control and internal model formation when reaching in a randomly varying dynamical environment," *Journal of Neurophysiology*, vol. 86, no. 2, pp. 1047–1051, 2001.

[12] S. Chan, B. Yao, W. Gao et al., "Robust impedance control of robot manipulators," *Robotics Automation*, vol. 6, no. 4, pp. 220–227, 1991.

[13] S. Jung and T. C. Hsia, "Neural network impedance force control of robot manipulator," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 3, pp. 451–461, 1998.

[14] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control of robot manipulators under unknown environment," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 474–483, 2004.

[15] J. Peng, J. Wang, and Y. Wang, "Neural network based robust hybrid control for robotic system: an H_∞ approach," *Nonlinear Dynamics*, vol. 65, no. 4, pp. 421–431, 2011.

[16] H. P. Singh and N. Sukavanam, "Stability analysis of robust adaptive hybrid position/force controller for robot manipulators using neural network with uncertainties," *Neural Computing and Applications*, vol. 22, no. 7-8, pp. 1745–1755, 2013.

[17] J. Peng, Y. Liu, and J. Wang, "Fuzzy adaptive output feedback control for robotic systems based on fuzzy adaptive observer," *Nonlinear Dynamics*, vol. 78, no. 2, pp. 789–801, 2014.

[18] J. Peng and R. Dubay, "Adaptive fuzzy backstepping control for a class of uncertain nonlinear strict-feedback systems based on dynamic surface control approach," *Expert Systems with Applications*, vol. 120, pp. 239–252, 2019.

[19] W. He and Y. Dong, "Adaptive fuzzy neural network control for a constrained robot using impedance learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1174–1186, 2018.

[20] H. N. Rahimi, I. Howard, and L. Cui, "Neural adaptive tracking control for an uncertain robot manipulator with time-varying joint space constraints," *Mechanical Systems and Signal Processing*, vol. 112, pp. 44–60, 2018.

[21] Z. Yang, J. Peng, and Y. Liu, "Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer," *Neurocomputing*, 2018.

[22] J. Li, L. Liu, Y. Wang, and W. Liang, "Adaptive hybrid impedance control of robot manipulators with robustness against environment's uncertainties," in *Proceedings of the 2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1846–1851, Beijing, China, August 2015.

[23] Z. Jhan, C. Lee, and C. Lin, "A new adaptive fuzzy neural force controller for robots manipulator interacting with environments," in *Proceedings of the 2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 572–577, Guangzhou, China, July 2015.

[24] B. Baigzadehnoe, Z. Rahmani, A. Khosravi, and B. Rezaie, "On position/force tracking control problem of cooperative robot manipulators using adaptive fuzzy backstepping approach," *ISA Transactions*, vol. 70, pp. 432–446, 2017.

[25] J. Duan, Y. Gan, M. Chen, and X. Dai, "Adaptive variable impedance control for dynamic contact force tracking in uncertain environment," *Robotics and Autonomous Systems*, vol. 102, pp. 54–65, 2018.

[26] W. Xu, C. Cai, M. Yin et al., "Time-varying force tracking in impedance control a case study for automatic cell manipulation," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA)*, pp. 344–349, Hangzhou, China, June 2013.

[27] W. Xu, "Robotic Time-Varying Force Tracking in Position-Based Impedance Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 9, pp. 1–12, 2016.

[28] L. Roveda, F. Vicentini, N. Pedrocchi, and L. M. Tosatti, "Impedance control based force-tracking algorithm for interaction robotics tasks: an analytically force overshoots-free approach," in *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO '15)*, pp. 386–391, Milan, Italy, July 2015.

[29] L. Roveda, N. Iannacci, F. Vicentini, N. Pedrocchi, F. Braghin, and L. M. Tosatti, "Optimal Impedance Force-Tracking Control Design With Impact Formulation for Interaction Tasks," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 130–136, 2016.

[30] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive Jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 1024–1029, 2006.

[31] L. Cheng, Z.-G. Hou, and M. Tan, "Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model," *Automatica*, vol. 45, no. 10, pp. 2312–2318, 2009.

[32] H. Wang, "Passivity based synchronization for networked robotic systems with uncertain kinematics and dynamics," *Automatica*, vol. 49, no. 3, pp. 755–761, 2013.

[33] X. Liang, H. Wang, Y.-H. Liu, W. Chen, G. Hu, and J. Zhao, "Adaptive Task-Space Cooperative Tracking Control of Networked Robotic Manipulators Without Task-Space Velocity Measurements," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2386–2398, 2016.

[34] K. Lee and M. Buss, "Force Tracking Impedance Control with Variable Target Stiffness," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6751–6756, 2008.

[35] T. Kim, H. S. Kim, and J. Kim, "Position-based impedance control for force tracking of a wall-cleaning unit," *International Journal of Precision Engineering and Manufacturing*, vol. 17, no. 3, pp. 323–329, 2016.

[36] I. Bonilla, M. Mendoza, D. U. Campos-Delgado et al., "Adaptive impedance control of robot manipulators with parametric uncertainty for constrained path-tracking," *International Journal of Applied Mathematics and Computer Science*, vol. 28, no. 2, pp. 363–374, 2018.

[37] H. C. Lu, C. H. Tsai, and M. H. Chang, "Radial Basis Function Neural Network with Sliding Mode Control for Robotic Manipulators," in *Proceedings of the 2010 IEEE International Conference on Systems Man and Cybernetics*, pp. 1209–1215, Istanbul, Turkey, 2010.

[38] R. Kelly, V. Santibez, and A. Lora, *Control of robot manipulators in joint space*, Springer, London, UK, 2005.

[39] H. K. Khalil, *Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, USA, 2002.