Neural Processing Letters (2020) 51:1125–1144
https://doi.org/10.1007/s11063-019-10138-1

Neural Network-Based Hybrid Position/Force Tracking Control for Robotic Systems Without Velocity Measurement

Jinzhu Peng[1] · Shuai Ding[1] · Zeqi Yang[1] · Fangfang Zhang[1]

Published online: 24 October 2019

Abstract
In this paper, a hybrid position/force tracking control scheme based on neural network observer is proposed for robotic systems with uncertain parameters and external disturbances. First, an observer based on neural network is designed to estimate joint velocities. Then, a neural network-based adaptive hybrid position/force controller is proposed based on the observed joint velocities. By using strict positive real method and Lyapunov stability theory, it is proved that all the signals of the closed-loop system are ultimately uniformly bounded. Finally, the simulation tests on a two-link manipulator are conducted. The simulation results show the feasibility and effectiveness of the control scheme.

Keywords Robotic system · Hybrid position/force control · State observer · Neural network · Lyapunov stability

1 Introduction

With the development of technology, robot has been widely used in industrial fields, and is gradually entering people's daily life. However, due to external disturbance, friction, external environment and nonlinear coupling, there are still many problems to be solved. For example, when the joint velocities of the manipulator are unknown, it is difficult to find an excellent state feedback control scheme to achieve the desired tracking goal. In many industrial applications, due to the requirements of the task, the robot needs to contact with the environment, and the position and contact force must be accurately controlled. Therefore, how to design an ideal and effective control scheme is complex and difficult.

In recent years, many researchers have investigated various control methods to make the robot motion more compliant. There are two main ways to achieve the compliant motion: the hybrid position/force control and the impedance control. Raibert and Craig [1] proposed a hybrid position/force control, which assigned the force and position in any direction of

✉ Jinzhu Peng
jzpeng@zzu.edu.cn

Fangfang Zhang
zhangfangfang@zzu.edu.cn

[1] School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China

the workspace to each joint controller by Jacobian matrix. Lozano and Brogliato [2] proposed an adaptive force/position control scheme for robot manipulators based on a particular decomposition of the robot Jacobian and environment stiffness matrices. The schemes did not require measurement of the joint acceleration or the first derivative of force. In order to improve the force tracking control performance, many advanced methods have been further integrated into the hybrid position/force control scheme, such as sliding mode control [3–6], robust control [7–10], fuzzy control [11–14], adaptive neural network [15–19] and so on. Karayiannidis [15] proposed a neuro-adaptive controller for the force and position trajectory tracking of a robot in compliant contact with a flat surface under non-parametric uncertainties existing in the dynamic and contact model. Peng et al. [16] proposed an adaptive Jacobian and radial basis function (RBF) neural network-based position/force tracking impedance control scheme to control robotic system with uncertainties and external disturbances. The schemes mainly realized force feedback control of the manipulator through the impedance relationship. Kumar [17] proposed an adaptive control method based on neural network for the hybrid control of force and position of the rigid-link manipulator. However, the convergence speed of the system was relatively slow since the feedforward neural network (FFNN) was used to approximate the nonlinear model of the manipulator. Ghajar [18] designed an intelligent position/force hybrid controller to solve the contact friction between the end-effector and the environment in the present of the system uncertainties, then the experimental tests of constant force were also conducted. Rani [19] proposed a hybrid intelligent force/position control scheme in the presence of external disturbances and model uncertainties. However, the joint velocities of the robot were assumed to be measured directly and accurately.

To solve the trajectory control and the environmental contact constraints, Mills and Goldenberg [20] proposed a method to simultaneously control the contact force exerted by the manipulator and the position of the end-effector on the contact surface. Yoshikawa and Sudou [21] proposed a dynamic hybrid control method, and developed an on-line estimation algorithm to estimate the local shape constrained surface measurement data, so that the dynamic hybrid control method was more practical and effective. However, the above control methods were designed under the known environment. For the uncertain environment, Ravandi et al. [22] combined fuzzy logic with traditional sliding mode control, a full approximation force/position hybrid control was then designed, and the adaptive PI controller was used to estimate the robust part.

In the above investigations, the joint position and velocity of the robot were usually assumed to be measured directly and accurately. However, in practical, due to the cost and noise disturbances, the effective observers were usually designed to replace the actual joint velocities measurement to reduce the cost and noise disturbances [23–30]. By employing neural network to compensate gravity and friction terms, Yu et al. [23] proposed a high-gain observer and presented the direct relationship between observer gain and observer error. Su et al. [24] proposed a simple nonlinear observer for a class of uncertain nonlinear multiple input multiple output (MIMO) mechanical systems with first order differentiable dynamics. Golea [25] proposed a fuzzy adaptive control method for multi-link robots with structural uncertainties and unstructured uncertainties. The scheme did not need to derive the linear equations of the robot dynamics. Chaudhary [27] proposed a hybrid fuzzy PD+I force-plus-position controller based on velocity observer. In the above observers, neural network control techniques were also widely used to approximate unknown nonlinear dynamics and compensate for uncertainties in dynamic models. Yang et al. [29] proposed an adaptive neural network force tracking impedance control scheme based on a nonlinear observer, the accuracy of joint position and force tracking were then improved by using adaptive RBF neural network to compensate the uncertainty of the system. Sun [31] proposed an observer-based adaptive controller based on neural network for trajectory tracking of robot with unknown dynamic nonlinearity. The scheme could guarantee the final values of tracking error and observer error as well as the bound of neural network weights. Abdollahi [32] proposed an observer based on nonlinear parameter neural network for general multi-variable nonlinear systems. The observer could be applied to systems with higher nonlinearity without any prior knowledge of system dynamics.

In this paper, assume that only the positions and force of robotic manipulator can be measured, while the velocities are assumed to be unknown, a neural network-based hybrid position/force tracking control scheme is proposed to control the robotic system based on a neural network-based velocity observer. The contributions of this manuscript can be stated as follows,

(1) A neural network-based velocity observer is combined into position/force tracking control framework.
(2) A state observer based on RBF neural network is proposed to estimate the velocities of the robotic system.
(3) Based on the observed joint velocities, a hybrid position/force control scheme based on RBF neural network is proposed for controlling robotic system without velocity measurement.
(4) By using the strict positive real method and Lyapunov stability theory, it is proved that all signals of the closed-loop system composed of robotic system, velocity observer and hybrid position/force controller are eventually uniformly bounded.

Finally, the simulation tests on the two-link manipulator are conducted, which show the feasibility and effectiveness of the design scheme.

The rest of this paper is organized as follows. In Sect. 2, some necessary mathematical knowledge is given, and the robot dynamic model and the RBF neural network are also presented. In Sect. 3, the RBF neural network-based adaptive observer is designed and the stability of the observer is also analyzed. In Sect. 4, based on the estimated joint velocities, a neural network adaptive hybrid position/force control scheme is designed. Contrastive simulation tests on a two-link manipulator are conducted in Sect. 5. Finally, the conclusion is drawn in Sect. 6.

2 Preliminaries

In this paper, standard notations are used. Let $\Re$ be the real number set, $\Re^n$ be the n-dimensional vector space, $\Re^{n\times n}$ be the $n \times n$ real matrix space. The norm of vector $x \in \Re^n$ and that of matrix $A \in \Re^{n\times n}$ are defined as $\parallel x \parallel = \sqrt{x^{\mathrm{T}}x}$ and $\parallel A \parallel = tr(A^{\mathrm{T}}A)$, respectively. If y is a scalar, then $\parallel y \parallel$ denotes 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 an $n \times n$ identity matrix. sgn$(\cdot)$ is a standard sign function. Denote $\parallel x \parallel_A^2 = x^{\mathrm{T}}Ax \geq \lambda_{\min}(A) \parallel x \parallel^2$, A is a positive symmetry matrix. From [33], we can obtain the definitions of strictly positive real (SPR) and Kalman–Yakubovich–Popov (KYP) Lemma as follows.

Definition 1 A real rational transfer function matrix $G(s)$ is SPR if: for some $\varepsilon > 0$, (i) no element of $G(s-\varepsilon)$ has a pole in $Re[s-\varepsilon] > 0$; (ii) $G(s-\varepsilon) + G^{\mathrm{T}}(-(s-\varepsilon)) \geq 0$ in $Re[s-\varepsilon] > 0$.

Lemma 1 *Suppose that the matrix $G(s) + G^{\mathrm{T}}(-s)$ has full rank in the complex plane,where $G(s) = C(sI-A)^{-1}B$ is the real rational transfer function; A satisfies that $det(sI-A)$ has* *only zeros in the open left-half plane;* (A, B) *is completely controllable. Then the positive definite symmetric matrices* Q *and* P *satisfying,*

$$\begin{cases} A^{\mathrm{T}}P + PA = -Q \\ PB = C^{\mathrm{T}} \end{cases}$$

exist if and only if $G(s)$ *is SPR.*

2.1 Dynamical Model and Properties for Robot Manipulators

The general dynamic model of a rigid-link robot with n joints can be represented as follows,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F_n(\dot{q}) + \tau_d = \tau - \tau_e \tag{1}$$

where $q, \dot{q}, \ddot{q} \in \Re^n$ are the position, velocity and acceleration vectors, respectively. $M(q) \in \Re^{n\times n}$ is the inertia matrix. $C(q, \dot{q}) \in \Re^{n\times n}$ is the vector of centripetal and Coriolis forces term. $G(q) \in \Re^n$ is the gravity effects. $F_n(\dot{q}) \in \Re^n$ denotes the friction effects. $\tau_d \in \Re^n$ is the vector of unknown bounded external disturbances. $\tau \in \Re^n$ is the torque of robot input vector. $\tau_e \in \Re^n$ is the interaction torque between robot and environment.

The dynamic model of Eq. (1) has the following properties which are useful for subsequent work.

Property 1 *The inertia matrix* $M(q)$ *is symmetric, positive definite and satisfies,*

$$M_m\|\zeta\|^2 \leq \zeta^{\mathrm{T}}M(q)\zeta \leq M_M\|\zeta\|^2 \quad \forall\zeta \in \Re^n$$

where M_m *and* M_M *are positive constants.*

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

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

Assumption 1 The friction effects and external disturbances are bounded, i.e.,

$$\|F_n(\dot{q})\| \leq \alpha + \beta\|\dot{q}\|, \ \|\tau_d\| \leq d$$

where α, β and d are positive constants.

Considering the measurement error and environmental factors, it is difficult to obtain the actual values of the physical parameters such as the length of the link and the mass of the link. Therefore, in this paper, the matrices $M(q)$, $C(q, \dot{q})$, $G(q)$ are represented by the nominal parts $M_0(q)$, $C_0(q, \dot{q})$, $G_0(q)$ and the uncertain parts $\Delta M(q)$, $\Delta C(q, \dot{q})$, $\Delta G(q)$ respectively, that is,

$$\begin{aligned} M(q) &= M_0(q) + \Delta M(q) \\ C(q, \dot{q}) &= C_0(q, \dot{q}) + \Delta C(q, \dot{q}) \\ G(q) &= G_0(q) + \Delta G(q) \end{aligned}$$

Assumption 2 The uncertain parts of the robot parameters are bounded, that is

$$\|\Delta M(q)\| \leq \psi_M, \|\Delta C(q, \dot{q})\| \leq \psi_C, \|\Delta G(q)\| \leq \psi_G$$

where ψ_M, ψ_C and ψ_G are positive constants, and $\psi_M \leq \lambda_{\min}(M_0(q))$.

The relationship between the contact force $F_e \in \Re^m$ at the end-effector and the contact torque $\tau_e \in \Re^n$ in the joint space can be expressed as,

$$\tau_e = J^{\mathrm{T}}(q)F_e \tag{2}$$

where $J(q)$ is the $m \times n$ $(m \leq n)$ dimensional Jacobian matrix.

In the Cartesian space, the position of the end-effector associated with the joint angle of the robot joint can be obtained by the following kinematic equation,

$$X = L(q) \tag{3}$$

where $X \in \Re^m$ is the position of the end-effector, $L(q)$ is the kinematic function of robot. Then,

$$\dot{X} = J(q)\dot{q} \tag{4}$$

Taking the first derivative of Eq. (4), we can obtain,

$$\ddot{X} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \tag{5}$$

In the task space, when the robot is in contact with the environment, the contact force F_e of the robot can be obtained from the deformation of the environment, that is,

$$F_e = K_e(X - X_e) \tag{6}$$

where K_e is the environmental stiffness matrix, X_e is the coordinate of the contact point of the end-effector when the contact force F_e is zero. In general, the stiffness matrix K_e may be singular. Therefore, F_e can be decomposed as,

$$F_e = \begin{bmatrix} F \\ F' \end{bmatrix} = \begin{bmatrix} K \\ K' \end{bmatrix} (X - X_e) \tag{7}$$

where $F \in \Re^s$, $K \in \Re^{s \times m}$ $(s \leq m)$ is a full-rank matrix, and the rows of $K' \in \Re^{(m-s)\times m}$ depend linearly on the rows of K. Therefore, only F can be controlled, and the element of F' is a linear combination of F.

2.2 Radial Basis Function Neural Network

As a typical neural network, it has been proved that the radial basis function neural network (RBFNN) can approximate any nonlinear function. The RBFNN with multiple input and single output can be described as,

$$y(x) = \sum_{j=1}^{m} w_j \xi_j(x) = W^{\mathrm{T}}\xi(x) \tag{8}$$

where $W = [w_1, ..., w_m]^{\mathrm{T}}$ is the network weight vector and x is the network input vector. $\xi(x) = [\xi_1(x), ..., \xi_m(x)]^{\mathrm{T}}$ is the basis function of the hidden layer that is usually defined as follows,

$$\xi_j(x) = e^{-\frac{\|x - c_j\|}{\delta_j^2}} \quad j = 1, 2, ..., m \tag{9}$$

where c_j and δ_j represent the center of hidden layer neuron and the width of the Gauss basis function, respectively.

3 Neural Network-Based Velocity Observer Design and Stability Analysis

Let $x_1 = q$ and $x_2 = \dot{q}$, the robotic dynamic Eq. (1) can be rewritten as,

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M^{-1}(x_1)(\tau - \tau_e - \tau_d) - M^{-1}(x_1)[C(x_1, x_2)x_2 + F_n(x_2) + G(x_1)] \end{cases} \tag{10}$$

Defining

$$D_o(x_1, x_2) = K_1 x_2 - M^{-1}(x_1)[C(x_1, x_2)x_2 + F_n(x_2) + G(x_1)] \tag{11}$$

where $K_1 \in \Re^{n\times n}$ is a diagonal constant positive definite matrix.

Considering Eqs. (10) and (11), the robotic dynamic Eq. (1) can be further written as,

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -K_1 x_2 + M^{-1}(x_1)(\tau - \tau_e - \tau_d) + D_o(x_1, x_2) \end{cases} \tag{12}$$

Using the RBFNN for approximation of unknown function $D_o(x_1, x_2)$, which can be written as,

$$D_o(x_1, x_2) = W_o^{\mathrm{T}} \xi_o(x_1, x_2) + \varepsilon_o(x_1, x_2) \tag{13}$$

where W_o represents the ideal weight matrix of the RBFNN, $\xi_o(x_1, x_2)$ is the basis functions of the hidden layer, $\varepsilon_o(x_1, x_2)$ denotes reconstruction error of RBFNN.

3.1 Neural Network-Based Velocity Observer

First, the nonlinear velocity observer based on neural network is designed as follows,

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_2 \\ \dot{\hat{x}}_2 = -K_1 \hat{x}_2 + K_2 \tilde{x}_1 + M_0^{-1}(x_1)(\tau - \tau_e) + \hat{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) + v_o \end{cases} \tag{14}$$

where $\hat{x}_1$ and $\hat{x}_2$ are the estimated values of x_1 and x_2, respectively, $K_2 \in \Re^{n\times n}$ is a diagonal constant positive definite matrix, and satisfies $K_2 > K_1$, $\hat{W}_o$ represents the estimation of W_o, v_o is a robust compensator.

Let us define the observation error as follows,

$$\tilde{x}_1 = x_1 - \hat{x}_1 \quad \tilde{x}_2 = x_2 - \hat{x}_2 \tag{15}$$

Then, according to Eqs. (14) and (15), the observation error of observer can be obtained as,

$$\begin{cases} \dot{\tilde{x}}_1 = \tilde{x}_2 \\ \dot{\tilde{x}}_2 = \dot{x}_2 + K_1 \hat{x}_2 - K_2 \tilde{x}_1 - M_0^{-1}(x_1)(\tau - \tau_e) - \hat{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) - v_o \end{cases} \tag{16}$$

Furthermore, the observation error can be written as,

$$\begin{cases} \dot{\tilde{x}}_1 = \tilde{x}_2 \\ \dot{\tilde{x}}_2 = -K_1 \tilde{x}_2 - K_2 \tilde{x}_1 + D_o(x_1, x_2) - \hat{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) \\ \qquad - M^{-1}(x_1)\tau_d - M^{-1}(x_1)\Delta M(x_1) M_0^{-1}(x_1)(\tau - \tau_e) - v_o \end{cases} \tag{17}$$

Since x_2 can not be obtained directly from the measurement, according to Eq. (13), we have,

$$\begin{aligned}
&D_o(x_1, x_2) - \hat{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) \\
&\quad = W_o^{\mathrm{T}} \xi_o(x_1, x_2) + \varepsilon_o(x_1, x_2) - \hat{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) \\
&\quad = \tilde{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) - W_o^{\mathrm{T}} [\xi_o(x_1, x_2) - \xi_o(x_1, \hat{x}_2)] + \varepsilon_o(x_1, x_2) \\
&\quad = \tilde{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) - W_o^{\mathrm{T}} \tilde{\xi}_o(x_1, x_2, \hat{x}_2) + \varepsilon_o(x_1, x_2)
\end{aligned} \tag{18}$$

where $\tilde{W}_o = W_o - \hat{W}_o$ is the weight estimation error and $\tilde{\xi}_o(x_1, x_2, \hat{x}_2) = \xi_o(x_1, x_2) - \xi_o(x_1, \hat{x}_2)$ is the estimated error of the hidden layer.

Assumption 3 The estimation errors of weight and the basis functions are bounded and satisfied,

$$\|W_o^{\mathrm{T}} \tilde{\xi}_o(x_1, x_2, \hat{x}_2)\| \le \psi_{W_o} \psi_{\xi_o}, \quad \|\varepsilon_o(x_1, x_2)\| \le \psi_{\varepsilon_o}$$

where ψ_{W_o}, ψ_{ξ_o} and ψ_{ε_o} are positive constants.

According to Eqs. (17) and (18), the observer estimation errors Eq. (16) can be written as

$$\begin{cases} \dot{\tilde{x}}_1 = \tilde{x}_2 \\ \dot{\tilde{x}}_2 = -K_1 \tilde{x}_2 - K_2 \tilde{x}_1 + U_o \end{cases} \tag{19}$$

where

$$\begin{aligned}
U_o = {} & \tilde{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) - M^{-1}(x_1) \tau_d - M^{-1}(x_1) \varDelta M(x_1) M_0^{-1}(x_1)(\tau - \tau_e) \\
& - W_o^{\mathrm{T}} \tilde{\xi_o}(x_1, x_2, \hat{x}_2) + \varepsilon_o(x_1, x_2) - v_o
\end{aligned} \tag{20}$$

Therefore, Eq. (19) can be further written as,

$$\begin{cases} \dot{\tilde{x}} = A_o \tilde{x} - B_o U_o \\ \tilde{x}_1 = C_o \tilde{x} \end{cases} \tag{21}$$

where

$$\tilde{x} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix}, \quad A_o = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ -K_2 & -K_1 \end{bmatrix}, \quad B_o = \begin{bmatrix} 0_{n \times n} \\ I_{n \times n} \end{bmatrix}, \quad C_o = \begin{bmatrix} I_{n \times n} & 0_{n \times n} \end{bmatrix}$$

According to Definition 1, defining the output of observer estimation errors Eq. (21) as

$$\tilde{x}_1 = G_o(s) U_o \tag{22}$$

where $G_o(s) = C_o(sI - A_o)^{-1} B_o$ is stable. The matrices K_1 and K_2 can be selected to satisfy that $det(sI - A_o)$ has zeros only in the open left-half plane and $G_o(s)$ is SPR.

For the subsequent development, considering the following inequations as,

$$\|M^{-1}(x_1)\| \le \frac{1}{\lambda_{\min}(M_0^{-1}(x_1)) - \psi_M} \tag{23}$$

$$\|M^{-1}(x_1) \varDelta M(x_1)\| \le \frac{\psi_M}{\lambda_{\min}(M_0^{-1}(x_1)) - \psi_M} \tag{24}$$

$$\begin{aligned}
&\| - M^{-1}(x_1) \tau_d - W_o^{\mathrm{T}} \tilde{\xi_o}(x_1, x_2, \hat{x}_2) + \varepsilon_o(x_1, x_2)\| \\
&\quad \le \frac{d}{\lambda_{\min}(M_0^{-1}(x_1)) - \psi_M} + \psi_{W_o} \psi_{\xi_o} + \psi_{\varepsilon_o} \\
&\quad \triangleq \rho_o
\end{aligned} \tag{25}$$

where ρ_o is a positive constant.

3.2 Stability Analysis

Theorem 1 *Assume that the robotic dynamic is given as Eq.* (1)*, Assumptions 1–3 are satisfied, and the transfer function* $G_o(s)$ *is SPR. According to Lemma 1, there exists a matrix* $P = P^{\mathrm{T}} > 0$ *such that the following matrix equality holds,*

$$\begin{cases} A_o^{\mathrm{T}} P + P A_o = -Q \\ P B_o = C_o^{\mathrm{T}} \end{cases} \tag{26}$$

Then, the nonlinear velocity observer based on RBFNN designed as Eq. (14) *is asymptotically stable, provided that the updating laws for the weights of the RBFNN and the robust compensator term are given as,*

$$\dot{\hat{W}}_o = -\Gamma_{W_o}^{-1} \xi_o(x_1, \hat{x}_2) \tilde{x}_1^{\mathrm{T}} \tag{27}$$

$$v_o = -\left[\frac{\psi_M \| M_0^{-1}(x_1)(\tau - \tau_e) \|}{\lambda_{\min}(M_0^{-1}(x_1)) - \psi_M} + \rho_o \right] \mathrm{sgn}(\tilde{x}_1) \tag{28}$$

where Γ_{W_o} *is a positive definite matrix.*

Proof Please refer to “Appendix A”. □

4 Neural Network-Based Hybrid Position/Force Tracking Controller Based on Velocity Observer and Stability Analysis

In this section, a hybrid position/force controller based on RBFNN is designed by using the measured joint positions and the observed joint velocities designed in the previous Sect. 3. The following identities are used for subsequent work,

$$I = J^+ J + J^- \tag{29}$$

$$I = K^+ K + K^- \tag{30}$$

where J^+ and K^+ are the pseudo inverse of J and K, respectively. J^- and K^- are projections of J and K, respectively.

Using Eqs. (6) and (30), $\dot{X}$ can be written as,

$$\dot{X} = K^+ K \dot{X} + K^- \dot{X} = K^+ \dot{F} + K^- \dot{X} \tag{31}$$

Then, the differentiation of Eq. (31) can be obtained as,

$$\ddot{X} = K^+ \ddot{F} + K^- \ddot{X} \tag{32}$$

Using Eqs. (4), (29) and (31), one can obtain,

$$\dot{q} = J^+ J \dot{q} + J^- \dot{q} = J^+ (K^+ \dot{F} + K^- \dot{X}) + J^- \dot{q} \tag{33}$$

Using Eqs. (5), (29) and (32), we have,

$$\ddot{q} = J^+ J \ddot{q} + J^- \ddot{q} = J^+ (K^+ \ddot{F} + K^- \ddot{X} - \dot{J} \dot{q}) + J^- \ddot{q} \tag{34}$$

Let $F_1 = F$, $F_2 = \dot{F}$, $X_1 = X$, $X_2 = \dot{X}$, $x_1 = q$ and $x_2 = \dot{q}$, and substituting Eqs. (33) and (34) into Eq. (1), we can obtain,

$$\begin{aligned} &M_0(x_1)[J^+(K^+\dot{F}_2 + K^-\dot{X}_2 - \dot{J}x_2) + J^-\dot{x}_2] + C_0(x_1, x_2)[J^+(K^+F_2 + K^-X_2) \\ &\quad + J^-x_2] + G_0(x_1) + \tau_d = \tau - \tau_e - D_{c0}(x_1, x_2, \dot{x}_2) \end{aligned} \tag{35}$$

where $D_{c0}(x_1, x_2, \dot{x}_2) = \varDelta M(x_1)\dot{x}_2 + \varDelta C(x_1, x_2)x_2 + \varDelta G(x_1) + F_n(x_2)$.

4.1 Neural Network-Based Hybrid Position/Force Tracking Controller Based on Velocity Observer

Since X_2 can not be measured directly, the observed velocities obtained by the observer in Sect. 3 are used to the controller. Defining the tracking error, estimation of velocities error and observation of velocity error as follows,

$$e_{F_1} = F_1 - F_d, e_{X_1} = X_1 - X_d, e_{x_1} = x_1 - x_d \tag{36}$$

$$\hat{e}_{F_2} = \hat{F}_2 - \dot{F}_d, \hat{e}_{X_2} = \hat{X}_2 - \dot{X}_d, \hat{e}_{x_2} = \hat{x}_2 - \dot{x}_d \tag{37}$$

$$\tilde{e}_{F_2} = F_2 - \hat{F}_2, \tilde{e}_{X_2} = X_2 - \hat{X}_2, \tilde{x}_2 = x_2 - \hat{x}_2 \tag{38}$$

where F_d, X_d, x_d are the desired values of F_1, X_1, x_1, respectively. $\hat{F}_2$, $\hat{X}_2$, $\hat{x}_2$ are the estimated values of F_2, X_2, x_2, respectively.

Defining the following filtered errors,

$$s = J^+(K^+E_F + K^-E_X) + J^-e_{x_2} \tag{39}$$

$$\hat{s} = J^+(K^+\hat{E}_F + K^-\hat{E}_X) + J^-\hat{e}_{x_2} \tag{40}$$

$$\tilde{s} = J^+(K^+\tilde{e}_{F_2} + K^-\tilde{e}_{X_2}) + J^-\tilde{x}_2 \tag{41}$$

where s, $\hat{s}$ and $\tilde{s}$ are the filtered tracking error, filtered estimation error and filtered observation error, respectively.

$$E_F = \dot{e}_{F_1} + K_{fp}e_{F_1}, E_X = \dot{e}_{X_1} + K_{pp}e_{X_1} \tag{42}$$

$$\hat{E}_F = \hat{e}_{F_2} + K_{fp}e_{F_1}, \hat{E}_X = \hat{e}_{X_2} + K_{pp}e_{X_1} \tag{43}$$

where K_{fp} and K_{pp} are diagonal constant positive definite matrices.

Taking the first derivative of Eq. (40) and multiplying $M_0(x_1)$ on both sides, we can obtain,

$$\begin{aligned} M_0(x_1)\dot{\hat{s}} &= M_0(x_1)[J^+(K^+\dot{\hat{E}}_F + K^-\dot{\hat{E}}_X) + J^-\dot{\hat{e}}_{x_2} + Z_{\hat{s}}] \\ &= M_0(x_1)\{J^+[K^+(\dot{\hat{E}}_F - \dot{F}_2) +^- (\dot{\hat{E}}_X - \dot{X}_2) + \dot{J}x_2] + J^-(\dot{\hat{e}}_{x_2} - \dot{x}_2) \\ &\quad + Z_{\hat{s}}\} + M_0(x_1)[J^+(K^+\dot{F}_2 + K^-\dot{X}_2 - \dot{J}x_2) + J^-\dot{x}_2] \end{aligned} \tag{44}$$

where $Z_{\hat{s}} = \dot{J}^+(K^+\hat{E}_F + K^-\hat{E}_X) + \dot{J}^-\hat{e}_{x_2}$.

Substituting Eq. (35) into Eq. (44), we can obtain,

$$\begin{aligned} M_0(x_1)\dot{\hat{s}} &= M_0(x_1)\{J^+[K^+(\dot{\hat{E}}_F - \dot{F}_2) + K^-(\dot{\hat{E}}_X - \dot{X}_2) + \dot{J}x_2] + J^-(\dot{\hat{e}}_{x_2} - \dot{x}_2) \\ &\quad + Z_{\hat{s}}\} + C_0(x_1, x_2)\{J^+[K^+(\hat{E}_F - F_2) + K^-(\hat{E}_X - X_2)] + J^-(\hat{e}_{x_2} \\ &\quad - x_2)\} - G_0(x_1) - C_0(x_1, x_2)\hat{s} + \tau - \tau_e - \tau_d - D_{c0}(x_1, x_2, \dot{x}_2) \\ &= M_0(x_1)\{J^+[K^+(K_{fp}\hat{e}_{F_2} - \ddot{F}_d) + K^-(K_{pp}\hat{e}_{X_2} - \ddot{X}_d)] - J^-\ddot{x}_d\} \\ &\quad + C_0(x_1, \hat{x}_2)\{J^+[K^+(K_{fp}e_{F_1} - \dot{F}_d) + K^-(K_{pp}e_{X_1} - \dot{X}_d)] - J^-\dot{x}_d\} \\ &\quad - G_0(x_1) + M_0(x_1)\{J^+[K^+(\dot{\tilde{e}}_{F_2} - K_{fp}\tilde{e}_{F_2}) + K^-(\dot{\tilde{e}}_{X_2} - K_{pp}\tilde{e}_{X_2}) \\ &\quad + \dot{J}x_2] + J^-\dot{\tilde{x}}_2 + Z_{\hat{s}}\} - \tilde{C}_0(x_1, x_2, \hat{x}_2)\{J^+[K^+(K_{fp}e_{F_1} - \dot{F}_d) \end{aligned}$$

$$+ K^{-}(K_{pp}e_{X_1} - \dot{X}_d)] - J^{-}\dot{x}_d\} + C_0(x_1, x_2)\{J^{+}[K^{+}\tilde{e}_{F_2} + K^{-}\tilde{e}_{X_2}] \\ + J^{-}\tilde{x}_2\} + D_{c0}(x_1, x_2, \dot{x}_2) - C_0(x_1, x_2)\hat{s} + \tau - \tau_e - \tau_d \tag{45}$$

where $\tilde{C}_0(x_1, x_2, \hat{x}_2) = C_0(x_1, x_2) - C_0(x_1, \hat{x}_2)$.

Combining Eqs. (3), (4), (5) with (6), one can conclude that the force and position of the end-effector can be obtained from the joint position information. Since $\hat{s}$ is the function of $[x_1, \hat{x}_2]$, defining

$$D_{c1}(x_1, \hat{x}_2) = M_0(x_1)\{J^{+}[K^{+}(K_{fp}\hat{e}_{F_2} - \ddot{F}_d) + K^{-}(K_{pp}\hat{e}_{X_2} - \ddot{X}_d)] - J^{-}\ddot{x}_d\} \\ + C_0(x_1, \hat{x}_2)\{J^{+}[K^{+}(K_{fp}e_{F_1} - \dot{F}_d) + K^{-}(K_{pp}e_{X_1} - \dot{X}_d)] \\ - J^{-}\dot{x}_d\} - G_0(x_1) \tag{46}$$

$$D_{c2}(x_1, x_2) = M_0(x_1)\{J^{+}[K^{+}(\dot{\tilde{e}}_{F_2} - K_{fp}\tilde{e}_{F_2}) + K^{-}(\dot{\tilde{e}}_{X_2} - K_{pp}\tilde{e}_{X_2}) + \dot{J}x_2] \\ + J^{-}\dot{\tilde{x}}_2 + Z_{\hat{s}}\} - \tilde{C}_0(x_1, x_2, \hat{x}_2)\{J^{+}[K^{+}(K_{fp}e_{F_1} - \dot{F}_d) \\ + K^{-}(K_{pp}e_{X_1} - \dot{X}_d)] - J^{-}\dot{x}_d\} + C_0(x_1, x_2)\{J^{+}[K^{+}\tilde{e}_{F_2} \\ + K^{-}\tilde{e}_{X_2}] + J^{-}\tilde{x}_2\} + D_{c0}(x_1, x_2, \dot{x}_2) \tag{47}$$

Therefore, the closed-loop error dynamics Eq. (45) can be rewritten as,

$$M_0(x_1)\dot{\hat{s}} = D_{c1}(x_1, \hat{x}_2) - D_{c2}(x_1, x_2) - C_0(x_1, x_2)\hat{s} + \tau - \tau_e - \tau_d \tag{48}$$

Using the RBFNN for approximation of unknown function $D_{c_2}(x_1, x_2)$, which can be written as,

$$D_{c2}(x_1, x_2) = W_c^{\mathrm{T}}\xi_c(x_1, x_2) + \varepsilon_c(x_1, x_2) \tag{49}$$

where W_c represents the ideal weight matrix of the RBFNN, $\xi_c(x_1, x_2)$ is the basis functions of the hidden layer, $\varepsilon_c(x_1, x_2)$ is the approximation error.

Furthermore,

$$M_0(x_1)\dot{s} = D_{c1}(x_1, \hat{x}_2) - W_c^{\mathrm{T}}\xi_c(x_1, x_2) - C_0(x_1, x_2)\hat{s} + \tau - \tau_e - \varepsilon_c(x_1, x_2) - \tau_d \tag{50}$$

Then, the control law based on neural network observer can be designed as follows,

$$\tau = -K_d\hat{s} + \tau_e - D_{c1}(x_1, \hat{x}_2) + \hat{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2) + v_c \tag{51}$$

where K_d is a diagonal constant positive definite matrix, v_c is the robust compensator term.

Substituting Eq. (51) into Eq. (50) yields,

$$M_0(x_1)\dot{\hat{s}} = -[K_d + C_0(x_1, x_2)]\hat{s} - [W_c^{\mathrm{T}}\xi_c(x_1, x_2) - \hat{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2)] \\ - \varepsilon_c(x_1, x_2) + v_c - \tau_d \tag{52}$$

Since

$$W_c^{\mathrm{T}}\xi_c(x_1, x_2) - \hat{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2) \\ = \tilde{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2) + W_c^{\mathrm{T}}[\xi_c(x_1, x_2) - \xi_c(x_1, \hat{x}_2)] \\ = \tilde{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2) + W_c^{\mathrm{T}}\tilde{\xi}_c(x_1, x_2, \hat{x}_2) \tag{53}$$

where $\tilde{W}_c = W_c - \hat{W}_c$ is the weight estimation error and $\tilde{\xi}_c(x_1, x_2, \hat{x}_2) = \xi_c(x_1, x_2) - \xi_c(x_1, \hat{x}_2)$ is the estimated error of the basis function.

Assumption 4 The estimation errors of weight and the basis functions are bounded and satisfied,

$$\|W_c^{\mathrm{T}}\tilde{\xi}_c(x_1, x_2, \hat{x}_2)\| \leq \psi_{W_c}\psi_{\xi_c}, \quad \|\varepsilon_c(x_1, x_2)\| \leq \psi_{\varepsilon_c}$$

where ψ_{W_c}, ψ_{ξ_c} and ψ_{ε_c} are positive constants.

Then, substituting Eq. (53) into Eq. (52), we can obtain,

$$\begin{aligned} M_0(x_1)\dot{\hat{s}} = &-[K_d + C_0(x_1, x_2)]\hat{s} - \tilde{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2) + v_c - \varepsilon_c(x_1, x_2) \\ &- W_c^{\mathrm{T}}\tilde{\xi}_c(x_1, x_2, \hat{x}_2) - \tau_d \end{aligned} \tag{54}$$

Considering Assumptions 1 and 4, we have,

$$\|W_c^{\mathrm{T}}\tilde{\xi}_c(x_1, x_2, \hat{x}_2) + \varepsilon_c(x_1, x_2) + \tau_d\| \leq \psi_{W_c}\psi_{\xi_c} + \psi_{\varepsilon_c} + d \triangleq \rho_c \tag{55}$$

where ρ_c is a positive constant.

4.2 Stability Analysis

Theorem 2 *Considering the robotic dynamic Eq. (1), suppose that Assumptions 1–4 are satisfied, the control input torque τ is designed as Eq. (51), the adaptation laws for the weights of RBFNN and the robust compensator term are given as,*

$$\dot{\hat{W}}_c = -\Gamma_{W_c}^{-1}\xi_c(x_1, \hat{x}_2)\hat{s}^{\mathrm{T}} \tag{56}$$

$$v_c = \rho_c \mathrm{sgn}(\hat{s}) \tag{57}$$

where Γ_{W_c} is positive definite matrices.

Then, the RBFNN-based hybrid position/force controller designed as Eq. (51) is asymptotically stable, and the filtered estimation error $\hat{s}$ and the weight estimation error $\tilde{W}_c$ are bounded.

Proof Please refer to "Appendix B". □

Using Eqs. (39), (40) and (41), we can obtain

$$s = \hat{s} + \tilde{s} \tag{58}$$

Due to physical constraints, X, F, J^+, J^-, K^+, K^- are all bounded. According to the bounded J and K and Eqs. (4) and (6), one can conclude that $\tilde{s}$ defined by Eq. (38) is bounded since the observer error $\tilde{x}_2$ is bounded, which has been proven in Sect. 3. Therefore, s is also bounded since $\hat{s}$ is bounded that has been proved in "Appendix B".

Remark 1 In this paper, only the revolute joints of the robotic system are considered, where the joint positions and the lengths of the link are all bounded in the actual scene. Therefore, from Eqs. (3), (6) and $J(q) = \partial(X)/\partial(q)$, one can obtain that X, F, J^+, J^- are all bounded. Moreover, K^+ and K^- are both bounded since the environmental stiffness matrix K is also generally bounded.

Since K, J are full rank matrices, multiplying Eq. (39) by KJ, we can obtain,

$$KJs = E_F \tag{59}$$

Multiplying Eq. (39) by J, we can obtain,

$$Js = K^{-}E_X \tag{60}$$

Since s and K^{-} are bounded, one can obtain that E_F and E_X are bounded. It means that the tracking error e_{F_1} and e_{X_1}, velocities error $\dot{e}_{F_1}$ and $\dot{e}_{X_1}$ are all bounded. Using Eqs. (39), (59) and (60), since J^{-} is bounded, the joint velocity e_{x_2} is also bounded.

5 Simulation Examples

To verify the theoretical results, simulations are conducted on a two-link manipulator, and its dynamic model is given as follows,

$$\begin{aligned}&\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} + \begin{bmatrix} F_{n1} \\ F_{n2} \end{bmatrix} + \begin{bmatrix} \tau_{d1} \\ \tau_{d2} \end{bmatrix} \\ &= \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} - J^{\mathrm{T}}F\end{aligned} \tag{61}$$

where the inertia matrix,the vector of centripetal and coriolis forces term,the gravity effects and friction effects are given as follows,

$$\begin{aligned}&M_{11} = (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 c_2;\ M_{12} = m_2 l_2^2 + m_2 l_1 l_2 c_2;\\ &M_{21} = m_2 l_2^2 + m_2 l_1 l_2 c_2;\ M_{22} = m_2 l_2^2;\\ &C_{11} = -2m_2 l_1 l_2 \dot{q}_2 s_2;\ C_{12} = m_2 l_1 l_2 \dot{q}_2 s_2;\ C_{21} = m_2 l_1 l_2 \dot{q}_2 s_2;\ C_{22} = 0;\\ &G_1 = (m_1 + m_2) l_1 g c_1 + m_2 l_2 g c_{12};\ G_2 = m_2 l_2 g c_{12};\end{aligned}$$

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

5.1 Design Process

To summarize the analysis in Sects. 3 and 4, the step-by-step procedures of the neural network-based hybrid position/force control for robotic systems based on neural network-based observer are outlined as follows,

Step 1 *Select the observer gains as,*

$$K_1 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, K_2 = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}$$

Step 2 *The number of hidden-layer nodes of the RBFNN is chosen to be 20. The center* c_j *of the Gaussian function are 20 points in* $[-5, 5]$ *and width* δ_j *are set to be 0.5. The learning rate of weight is selected as* $\Gamma_{W_o} = 0.2I_{20}$*, set* $\psi_M = 10^{-8}$ *and* $\rho_o = 50$*, then the RBFNN-based observer can be obtained from Theorem 1.*

Step 3 *Choosing the controller gains* $K_{pp} = 25I_{2\times2}$, $K_{fp} = 100I_{2\times2}$, $K_d = 200I_{2\times2}$ *in Eqs.* (43) *and* (51)*, respectively, the structure of RBFNN is chosen as Step 3. The learning rate of weight is selected as* $\Gamma_{W_c} = 0.2I_{20}$*, and set* $\rho_c = 50$*, then the RBFNN-based hybrid position/force control for robotic systems based on neural network-based observer can be obtained from Theorem 2.*

Remark 2 To ensure that the errors of observer and controller are converged to zero quickly, the observer gains and controller gains would be usually selected to be large enough. However, it is not recommended to use very large gains, because this may lead to a stronger noise effect and may decrease the service life of the robotic system. Moreover, too large or too small value of the gains will lead to the overshoot. Therefore, the gains should be adjusted carefully for achieving suitable performances of the observer and controller.

5.2 Simulation Results

In this section, the proposed intelligent hybrid position/force control scheme based on observer is applied to control a two-link robotic manipulator. The nominal parameters of the robot used for simulation are $m_1 = 8\,\mathrm{kg}, m_2 = 4\,\mathrm{kg}$ and $l_1 = 1.1\,\mathrm{m}, l_2 = 0.9\,\mathrm{m}, g = 9.8\,\mathrm{m/s^2}, F_n = [5\dot{q}_1, 3\dot{q}_2]^{\mathrm{T}}$, while actual parameters of robot are chosen as $m_1 = 10\,\mathrm{kg}, m_2 = 8\,\mathrm{kg}$ and $l_1 = l_2 = 1\,\mathrm{m}$ to introduce the parameters uncertainties. Choosing the contact force $F = k(X - X_{xe})$ in the X_x direction where $k = 2000\,\mathrm{N/m}$ and $X_{xe} = 1\,\mathrm{m}$, the desired trajectories $X_{yd} = 1.5(1 - e^{-t})\mathrm{m}$ in X_y direction. For testing the robustness of the proposed method, choosing external disturbances $\tau_d = [-3\cos(5t), 3\sin(5t)]^{\mathrm{T}}$. The desired time-varying force F_d is chosen as a non-smooth function, which is described as,

$$F_d = \begin{cases} 4t + 20 - 8\mu, \; 2\mu \le t < 2\mu + 1 \\ -4t + 28 + 8\mu, \; 2\mu + 1 \le t < 2\mu + 2 \end{cases} \tag{62}$$

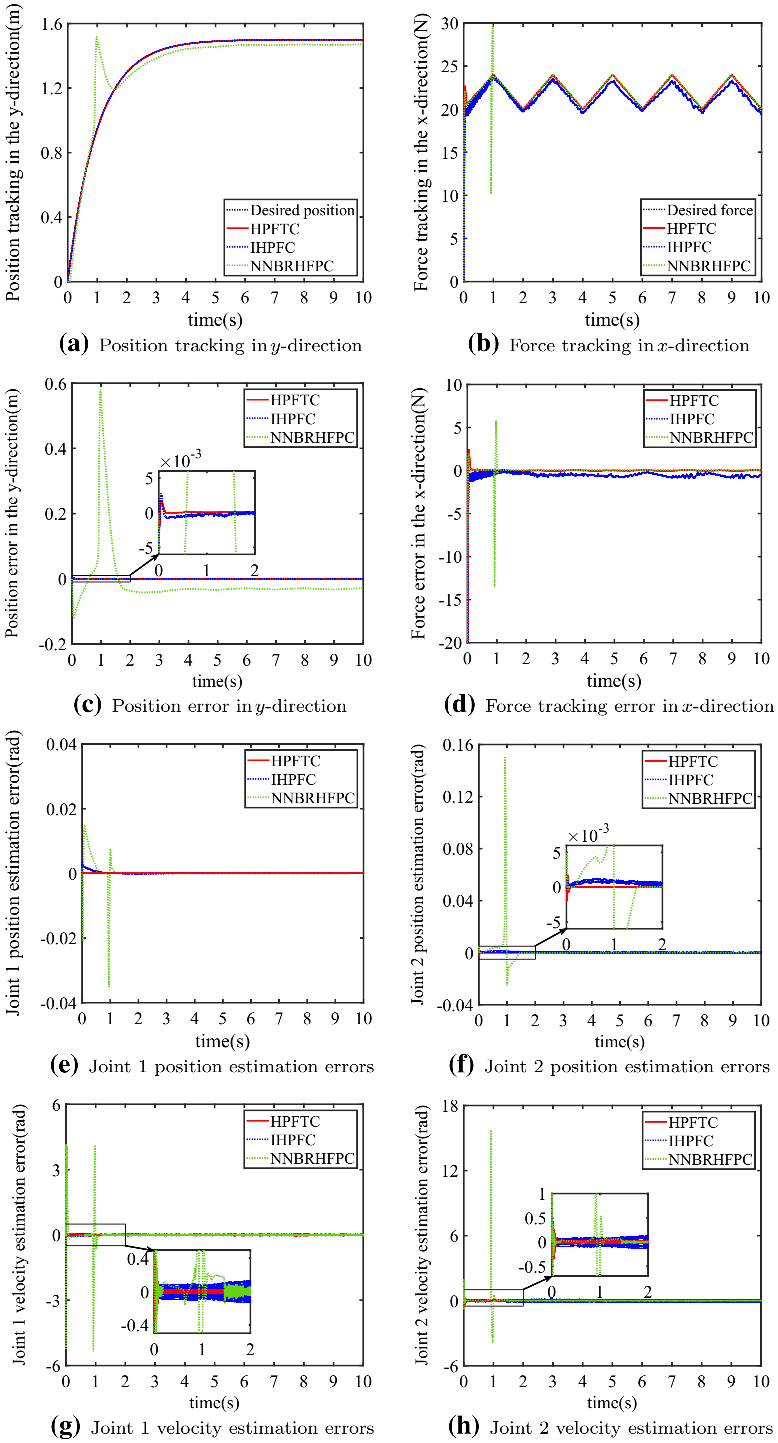
where $\mu = 0, 1, 2, \cdots$.

The initial conditions are $q(0) = [\pi/3, -2\pi/3]^{\mathrm{T}}\mathrm{rad}, \dot{q}(0) = [0, 0]^{\mathrm{T}}\mathrm{rad/s}$, and $K^+ = 1/k, K^- = [0\;\,0; 0\;\,1], J^+ = J^{-1}, J^- = 0$. The initial conditions are $\hat{q}(0) = [\pi/3, -2\pi/3]^{\mathrm{T}}\mathrm{rad}, \dot{\hat{q}}(0) = [0, 0]^{\mathrm{T}}\mathrm{rad/s}$. The total simulation time is 10s and the sampling time is 0.001s.

In order to facilitate the comparison and analysis, we conduct two cases to verify the performances of the proposed method, where Case 1 is used to test the performance of the designed observer and Case 2 is used to test the performance of the proposed observer-based controller.

Case 1 Under the condition of friction and external disturbance, based on high-gain observer [24],the intelligent hybrid position/force controller (IHPFC) [19] and the neural-network-based robust hybrid force/position controller (NNBRHFPC) [18] are used to compare with the proposed hybrid position/force tracking controller (HPFTC) based on neural network observer in this paper. Figure 1 shows the simulation results, where Fig. 1a is the position tracking the end-effector of the robot manipulator in y-direction. Figure 1b is the force tracking in x-direction. Fig. 1c is the position tracking error the end-effector in y-direction. Figure 1d is the force tracking error in x-direction. Figure 1e, f are the position estimation errors of joint 1 and joint 2, respectively. Figure 1g, h are the velocity estimation errors of joint 1 and joint 2, respectively.

From the simulation results of Fig. 1 the stability of the robot system can be achieved under the control of the above three observer-based hybrid position/force controllers. From Fig. 1c, d, it is obvious that the steady-state error of HPFTC tracking in force and position subspace is smaller than that of IHPFC and NNBRHFPC. At $t = 1$s, the tracking of NNBRHFPC presents severe chattering phenomenon in force and position subspace, which is caused by singularity. While there are no singularity in HPFTC and IHPFC, the system response of HPFTC is significantly better than that of IHPFC. Moreover, due to the large gain of the high-gain observer, the observed state has a certain range of chattering, which would affect



(a) Position tracking in y-direction

(b) Force tracking in x-direction

(c) Position error in y-direction

(d) Force tracking error in x-direction

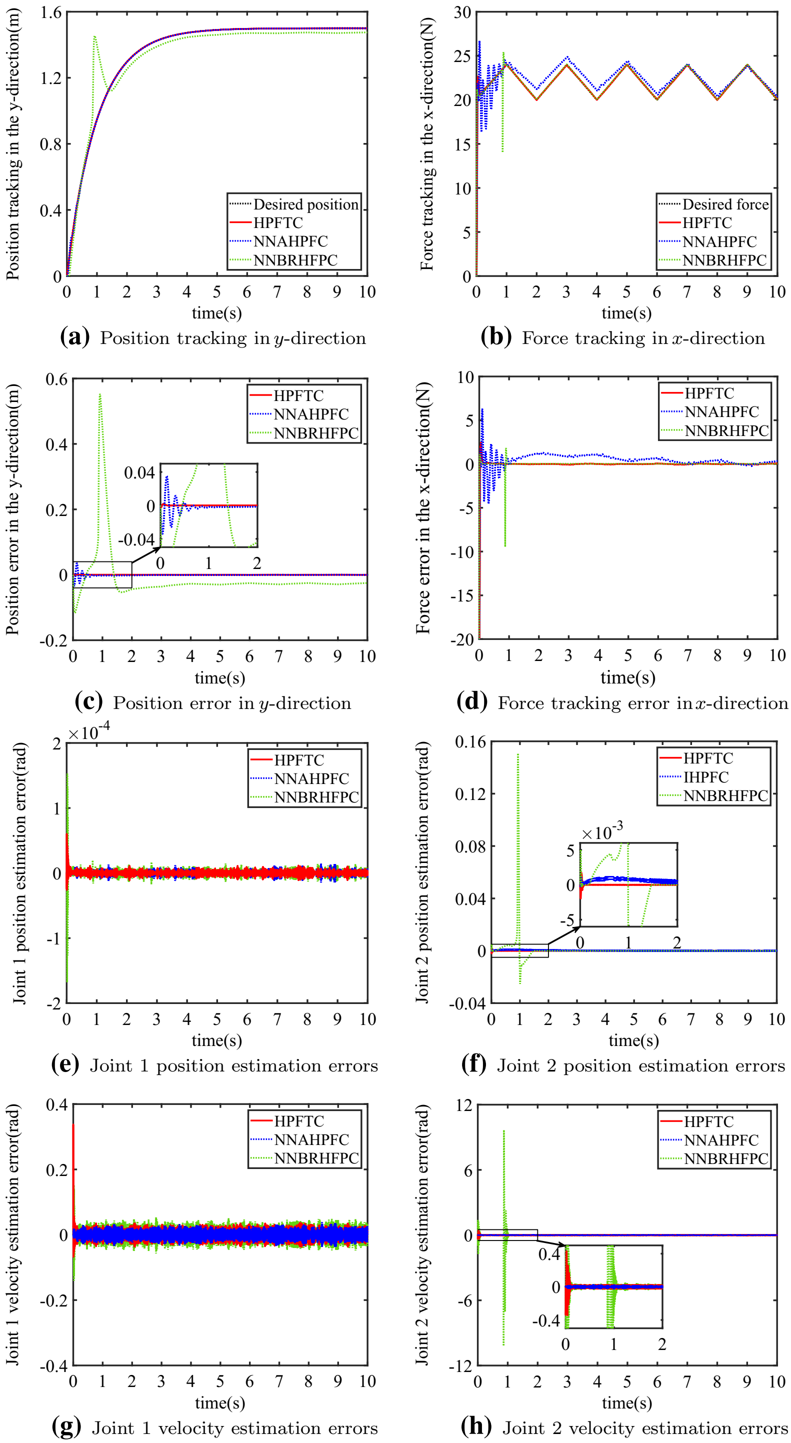
(e) Joint 1 position estimation errors

(f) Joint 2 position estimation errors

(g) Joint 1 velocity estimation errors

(h) Joint 2 velocity estimation errors

Fig. 1 Simulation results using NNO-based HPFTC, HGO-based IHPFC and HGO-based NNBRHFPC



(a) Position tracking in y-direction

(b) Force tracking in x-direction

(c) Position error in y-direction

(d) Force tracking error in x-direction

(e) Joint 1 position estimation errors

(f) Joint 2 position estimation errors

(g) Joint 1 velocity estimation errors

(h) Joint 2 velocity estimation errors

Fig. 2 Simulation results using NNO-based HPFTC, NNO-based NNAHPFC and NNO-based NNBRHFPC

the performance of the controller. From Fig. 1g, h, it can be seen that the designed observer based on neural network can achieve smaller steady-state error in comparison to the high-gain observer (HGO).

Case 2 Under the same conditions as **Case 1**, the neural network adaptive hybrid position/force controller(NNAHPFC) [17] and the neural-network-based robust hybrid force/position controller (NNBRHFPC) [18] are used to control the robotic system, but the joint velocity of the manipulator is obtained by the observer designed in this paper. Figure 2 show the simulation results, where Fig. 2a is the position tracking the end-effector of the robot manipulator in y-direction. Figure 2b is the force tracking in x-direction. Figure 2c is the position tracking error the end-effector in y-direction. Figure 2d is the force tracking error in x-direction. Figure 2e, f are the position estimation errors of joint 1 and joint 2, respectively. Figure 2g, h are the velocity estimation errors of joint 1 and joint 2, respectively.

According to the simulation results of Fig. 2, we can conclude that the proposed HPFTC, NNAHPFC and NNBRHFPC can achieve the stability of the system. From Fig. 2e–h, it can be seen that the observer designed in this paper demonstrates excellent performance. And the system regulation time of NNAHPFC is longer than that of the proposed HPFTC. This is because the convergence speed of FFNN used in NNAHPFC was relatively slow, which would affect the convergence rate and steady-state error of the system. The convergence speed of RBFNN used in the proposed HPFTC is better than that of FFNN. Moreover, from Fig. 2c,d, there is severe chattering of the force tracking and position tracking by using NNAHPFC in the initial stage, which may degrade the tracking performances and even cause the systems instability. Furthermore, the tracking control using NNBRHFPC also presents severe chattering phenomenon at $t = 1$s because of the existence of singularity, which reduces the control accuracy of the system. Instead, the response and steady state performances of the proposed HPFTC are better than those of NNAHPFC and NNBRHFPC. These results indicate that the proposed HPFTC can achieve the good convergence speed and small steady-state error.

6 Conclusions

This paper proposed a hybrid position/force tracking control scheme based on neural network observer for robotic systems in the presence of system uncertainties and external disturbances. For the unmeasurable joint velocities, a neural network-based observer was designed, and the observation errors are proved to be ultimately uniformly bounded by using strict positive real method and Lyapunov stability theory. An adaptive neural network-based hybrid position/force controller was then proposed based on the observed velocities to guarantee stability of closed-loop system and achieve a certain tracking performance. By using Lyapunov stability theory, it was proved that all the signals of the closed-loop system were ultimately uniformly bounded. Finally, the feasibility and effectiveness of the proposed neural network observer-based position/force control scheme were demonstrated by the simulation on a two-link manipulator.

Acknowledgements The authors would like to acknowledge the funding received from the National Natural Science Foundation of China (61773351, 61603345), the Program for Science & Technology Innovation Talents in Universities of Henan Province (20HASTIT031), the Training Plan for University's Young Backbone Teachers of Henan Province (2017GGJS004), the Natural Science Foundation of Henan Province (162300410260) and the China Scholarship Council (201907045008) to conduct this research investigation.

Compliance with Ethical Standards

Conflict of interest The authors declare that there are no conflicts of interest.

Appendix A: Proof of Theorem 1

Considering the following Lyapunov function,

$$V_o = \frac{1}{2}\tilde{x}^{\mathrm{T}} P \tilde{x} + \frac{1}{2}\tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o \tag{63}$$

Taking the differentiation of Eq. (63) and substituting Eqs. (21) and (26), yields

$$\begin{aligned}\dot{V}_o &= \frac{1}{2}\dot{\tilde{x}}^{\mathrm{T}} P \tilde{x} + \frac{1}{2}\tilde{x}^{\mathrm{T}} P \dot{\tilde{x}} + \tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o \\ &= \frac{1}{2}\tilde{x}^{\mathrm{T}}(A_o^{\mathrm{T}} P + P A_o)\tilde{x} + \frac{1}{2}U_o^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} + \frac{1}{2}\tilde{x}^{\mathrm{T}} P B_o U_o + \tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o \\ &= -\frac{1}{2}\tilde{x}^{\mathrm{T}} Q \tilde{x} + U_o^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} + \tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o\end{aligned} \tag{64}$$

According to Eq. (20), we can obtain,

$$\begin{aligned}&U_o^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} + \tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o \\ &\quad= [\tilde{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2) - M^{-1}(x_1)\tau_d - M^{-1}(x_1)\Delta M(x_1) M_0^{-1}(x_1)(\tau - \tau_e) \\ &\qquad + \varepsilon_o(x_1, x_2) - v_o - W_o^{\mathrm{T}} \tilde{\xi}_o(x_1, x_2, \hat{x}_2)]^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} + \tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o \\ &\quad= [\tilde{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2)]^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} + \tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o + [-M^{-1}(x_1)\Delta M(x_1) M_0^{-1}(x_1)(\tau - \tau_e) \\ &\qquad - M^{-1}(x_1)\tau_d - W_o^{\mathrm{T}} \tilde{\xi}_o(x_1, x_2, \hat{x}_2) + \varepsilon_o(x_1, x_2) - v_o]^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x}\end{aligned} \tag{65}$$

Now, using the updating laws Eq. (27), the facts $\dot{\tilde{W}} = -\dot{\hat{W}}$ and $B_o P^{\mathrm{T}} \tilde{x} = C_o \tilde{x} = \tilde{x}_1$, we can obtain,

$$[\tilde{W}_o^{\mathrm{T}} \xi_o(x_1, \hat{x}_2)]^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} + \tilde{W}_o^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o \leq 0 \tag{66}$$

According to inequality (25) and Eq. (28), we can obtain,

$$\begin{aligned}&[-M^{-1}(x_1)\Delta M(x_1) M_0^{-1}(x_1)(\tau - \tau_e) - M^{-1}(x_1)\tau_d - W_o^{\mathrm{T}} \tilde{\xi}_o(x_1, x_2, \hat{x}_2) \\ &\quad + \varepsilon_o(x_1, x_2) - v_o]^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} \leq 0\end{aligned} \tag{67}$$

Therefore,

$$U_o^{\mathrm{T}} B_o^{\mathrm{T}} P \tilde{x} + \widetilde{W_o}^{\mathrm{T}} \Gamma_{W_o} \dot{\tilde{W}}_o \leq 0 \tag{68}$$

Substituting Eq. (68) into Eq. (64) yields,

$$\dot{V}_o \leq -\frac{1}{2}\tilde{x}^{\mathrm{T}} Q \tilde{x} \tag{69}$$

This implies $V_o > 0$ and $\dot{V}_o \leq 0$, the stability of the observer can be then ensured so that $\tilde{x}$ and $\tilde{W}_o$ are bounded.

Appendix B: Proof of Theorem 2

Considering the following Lyapunov function,

$$V_c = \frac{1}{2}\hat{s}^{\mathrm{T}} M_0(x_1)\hat{s} + \frac{1}{2}\tilde{W}_c^{\mathrm{T}} \Gamma_{W_c} \dot{\tilde{W}}_c \tag{70}$$

Taking the differentiation of Eq.(70) yields

$$\dot{V}_c = \tfrac{1}{2}\hat{s}^{\mathrm{T}} \dot{M}_0(x_1)\hat{s} + \hat{s}^{\mathrm{T}} M_0(x_1)\dot{\hat{s}} + \tilde{W}_c^{\mathrm{T}} \Gamma_{W_c} \dot{\tilde{W}}_c \tag{71}$$

Substituting Eqs. (54) and (56) into Eq. (71), and considering the fact $\dot{\tilde{W}}_c = -\dot{\hat{W}}_c$, we can obtain,

$$\begin{aligned}\dot{V}_c = &\frac{1}{2}\hat{s}^{\mathrm{T}} \dot{M}_0(x_1)\hat{s} + \hat{s}^{\mathrm{T}}\{-[K_d + C_0(x_1, x_2)]\hat{s} - \tilde{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2) + \upsilon_c \\ &- W_c^{\mathrm{T}}\tilde{\xi}_c(x_1, x_2, \hat{x}_2) - \varepsilon_c(x_1, x_2) - \tau_d\} + \tilde{W}_c^{\mathrm{T}}\xi_c(x_1, \hat{x}_2)\hat{s}^{\mathrm{T}} \\ = &\frac{1}{2}\hat{s}^{\mathrm{T}}[\dot{M}_0(x_1) - 2C_0(x_1, x_2)]\hat{s} - \hat{s}^{\mathrm{T}} K_d \hat{s} + \hat{s}^{\mathrm{T}}\{\upsilon_c - \varepsilon_c(x_1, x_2) \\ &- W_c^{\mathrm{T}}\tilde{\xi}_c(x_1, x_2, \hat{x}_2) - \tau_d\}\end{aligned} \tag{72}$$

According to property 2, we have,

$$\dot{V}_c = -\hat{s}^{\mathrm{T}} K_d \hat{s} + \hat{s}^{\mathrm{T}}[\upsilon_c - \varepsilon_c(x_1, x_2) - W_c^{\mathrm{T}}\tilde{\xi}_c(x_1, x_2, \hat{x}_2) - \tau_d] \tag{73}$$

Substituting Eqs. (55) and (57) into Eq. (73), we can obtain,

$$\dot{V}_c \le -\hat{s}^{\mathrm{T}} K_d \hat{s} \tag{74}$$

It can be concluded that the closed-loop system is asymptotically stable. Integrating Eq. (74) from time $t = 0$ to $t = T$ yields,

$$\int_0^{\mathrm{T}} \parallel \hat{s} \parallel^2 dt \le \frac{V_c(0) - V_c(T)}{\lambda_{\min}(K_d)} \tag{75}$$

Using Eq. (70), we can obtain,

$$V_c(0) = \frac{1}{2}\hat{s}^{\mathrm{T}}(0) M_0(x_1)\hat{s}(0) + \frac{1}{2}\tilde{W}_c^{\mathrm{T}}(0)\Gamma_{W_c}\dot{\tilde{W}}_c(0) \tag{76}$$

According to inequality (75), we have,

$$\parallel \hat{s} \parallel \le \sqrt{\frac{\hat{s}^{\mathrm{T}}(0) M_0(x_1)\hat{s}(0) + \tilde{W}_c^{\mathrm{T}}(0)\Gamma_{W_c}\dot{\tilde{W}}_c(0)}{2\lambda_{\min}(K_d)}} \tag{77}$$

This implies that $\hat{s}$ and $\tilde{W}_c$ are bounded. Since W_c is bounded, $\hat{W}_c$ is hence bounded.

References

1. Raibert MH, Craig JJ (1981) Hybrid position/force control of manipulators. ASME J Dyn Syst Meas Control 103(2):126–133
2. Lozano R, Brogliato B (1992) Adaptive hybrid force-position control for redundant manipulators. IEEE Trans Autom Control 37(10):1501–1505
3. Kwan CM (1995) Hybrid force/position control for manipulators with motor dynamics using a sliding-adaptive approach. IEEE Trans Autom Control 40:963–968

4. Bassi E, Benzi F, Capisani LM, Cuppone D (2009) Hybrid position/force sliding mode control of a class of robotic manipulators. In: IEEE Conference on Decision & Control, pp 2966–2971
5. Chen ZH, Chen L (2011) Dynamics for dual-arm space robot with closed-chain and hybrid force and position control for grasped object based on sliding-mode compensation. Eng Mech 28(5):226–232
6. Peng J, Yang Z, Wang Y, Zhang F, Liu Y (2019) Robust adaptive motion/force control scheme for crawler-type mobile manipulator with nonholonomic constraint based on sliding mode control approach. ISA Trans 92:166–179
7. Peng J, Wang J, Wang Y (2011) Neural network based robust hybrid control for robotic system: an H_∞ approach. Nonlinear Dyn 65(4):421–431
8. Mohajerpoor R, Rezaei M, Talebi A et al (2012) A robust adaptive hybrid force/position control scheme of two planar manipulators handling an unknown object interacting with an environment. Proc Inst Mech Eng Part I J Syst Control Eng 226(4):509–522
9. Singh HP (2013) Stability analysis of robust adaptive hybrid position/force controller for robot manipulators using neural network with uncertainties. Neural Comput Appl 22(7–8):1745–1755
10. Zhou F, Li Y, Liu G (2017) Robust decentralized force/position fault-tolerant control for constrained reconfigurable manipulators without torque sensing. Nonlinear Dyn 89(3):955–969
11. Touati Y, Djouani K, Amirat Y (2004) Neuro-fuzzy based approach for hybrid force/position robot control. Integr Comput-Aided Eng 11(1):85–98
12. Canul RC, Garcia-Hernandez R, Rullan-Lara JL et al (2014) Decentralized adaptive fuzzy control applied to a robot manipulator. Adv Trends Soft Comput 314:123–135
13. Wen S, Yu H, Zhang B et al (2017) Fuzzy identification and delay compensation based on the force/position control scheme of the 5-DOF redundantly actuated parallel robot. Int J Fuzzy Syst 19(1):124–140
14. Peng J, Rickey D (2019) Adaptive fuzzy backstepping control for a class of uncertain nonlinear strict-feedback systems based on dynamic surface control approach. Expert Syst Appl 120:239–252
15. Karayiannidis Y, Rovithakis G, Doulgeri Z (2007) Force/position tracking for a robotic manipulator in compliant contact with a surface using neuro-adaptive control. Automatica 43(7):1281–1288
16. Peng J, Yang Z, Ma T (2019) Position/force tracking impedance control for robotic systems with uncertainties based on adaptive Jacobian and neural network. Complexity 2019:1–16
17. Kumar N, Panwar V, Sukavanam N, Sharma SP, Borm JH (2011) Neural network based hybrid force/position control for robot manipulators. Int J Precis Eng Manuf 12(3):419–426
18. Ghajar M-H, Keshmiri M, Bahrami J (2018) Neural-network-based robust hybrid force/position controller for a constrained robot manipulator with uncertainties. Trans Inst Meas Control 40(5):1625–1636
19. Rani K, Kumar N (2018) Design of intelligent hybrid force and position control of robot manipulator. Procedia Comput Sci 125:42–49
20. Mills JK, Goldenberg AA (1989) Force and position control of manipulators during constrained motion tasks. IEEE Trans Robot Autom 5(1):30–46
21. Yoshikawa T, Sudou A (1990) Dynamic hybrid position/force control of robot manipulators: on-line estimation of unknown constraint. IEEE Trans Robot Autom 2:1231–1236
22. Ravandi AK, Khanmirza E, Daneshjou K (2018) Hybrid force/position control of robotic arms manipulating in uncertain environment based on adaptive fuzzy sliding mode control. Appl Soft Comput 70:864–874
23. Yu W, Li XO (2006) PD control of robot with velocity estimation and uncertainties compensation. Int J Robot Autom 21(1):1–9
24. Su Y, Muller PC, Zheng C (2007) A simple nonlinear observer for a class of uncertain mechanical systems. IEEE Trans Autom Control 52(7):1340–1345
25. Goléa N, Goléa A, Barra K et al (2008) Observer-based adaptive control of robot manipulators: fuzzy systems approach. Appl Soft Comput 8(1):778–787
26. Peng J, Liu Y, Wang J (2014) Fuzzy adaptive output feedback control for robotic systems based on fuzzy adaptive observer. Nonlinear Dyn 78(2):789–801
27. Chaudhary H, Parashar A, Prasad R et al (2014) Velocity observer based fuzzy PD+I based hybrid force/position control of an industrial robot. In Engineering and computational sciences, pp 1–6
28. Bouakrif F (2017) Trajectory tracking control using velocity observer and disturbances observer for uncertain robot manipulators without tachometers. Meccanica 52(4–5):1–15
29. Yang Z, Peng J, Liu Y (2019) Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer. Neurocomputing 331:263–280
30. Kim YH, Lewis FL, Abdallah CT (1997) A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems. Automatica 33(8):1539–1543
31. Sun F, Sun Z, Woo PY (2001) Neural network-based adaptive controller design of robotic manipulators with an observer. IEEE Trans Neural Networks 12(1):54–67
32. Abdollahi F, Talebi HA, Patel RV (2006) A stable neural network-based observer with application to flexible-joint manipulators. IEEE Trans Neural Networks 17(1):118–129

33. Tao G, Ioannou PA (1988) Strictly positive real matrices and the Lefschetz–Kalman–Yakubovich lemma. IEEE Trans Autom Control 33(12):1183–1185

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.