


RESEARCH ARTICLE

Asymmetric constrained control scheme design with discrete output feedback in unknown robot–environment interaction system

Xinyi Yu, Huizhen Luo , Shuanwu Shi, Yan Wei and Linlin Ou*

Zhejiang University of Technology College of Information Engineering, Hangzhou 310023, China

*Corresponding author: E-mail: linlinou@zjut.edu.cn

Received: 30 October 2021; **Revised:** 14 July 2022; **Accepted:** 14 July 2022; **First published online:** 9 September 2022

Keywords: discrete output feedback, Adaptive dynamic programming, Neural network, Asymmetric barrier Lyapunov function, Human–robot interaction

Abstract

In this paper, an overall structure with the asymmetric constrained controller is constructed for human–robot interaction in uncertain environments. The control structure consists of two decoupling loops. In the outer loop, a discrete output feedback adaptive dynamics programming (OPFB ADP) algorithm is proposed to deal with the problems of unknown environment dynamic and unobservable environment position. Besides, a discount factor is added to the discrete OPFB ADP algorithm to improve the convergence speed. In the inner loop, a constrained controller is developed on the basis of asymmetric barrier Lyapunov function, and a neural network method is applied to approximate the dynamic characteristics of the uncertain system model. By utilizing this controller, the robot can track the prescribed trajectory precisely within a security boundary. Simulation and experimental results demonstrate the effectiveness of the proposed controller.

1. Introduction

The applications of human–robot interaction (HRI) range from material assembly [1], rehabilitation [2], and elderly care [3]. With the variability of the environment, the robot needs to adjust the uncertainty automatically. Improvement of the flexibility of the robot's interaction with the environment becomes a challenging problem.

In the fields of HRI control, impedance control [4] and admittance control [5] are two common methods by adjusting robot impedance parameters. In the 1980s, the concept of impedance control was proposed by Hogan [6]. In this method, a damping–spring–mass model was used for representing the dynamic relationship between the environment and the robot position. The input force of the robot is obtained by measuring the position of the robot with appropriate impedance parameters. Admittance control realizes the compliant control by controlling the interaction force and generating the desired trajectory. In the early research, model parameters of the two methods are usually prescribed [7–9]. In ref. [8], a guaranteed-stable interaction controller was derived for arbitrary linear, time-invariant systems. In ref. [9], a loop-shaping method was developed to design actuator controllers for physically interactive machines. However, inappropriate model parameters will cause outrageous errors. For human–environment interaction, both the varying environmental factors and the robot model parameters need to be considered, which makes it difficult to obtain appropriate parameters.

To solve the above problems, it is necessary to introduce variable impedance method into the field of robot control. The development of reinforcement learning provides the conditions for adaptive dynamic programming (ADP) [10–12], which is broadly applied in optimal control problems with uncertain dynamics. An adaptive optimization algorithm was proposed for continuous-time based on part of dynamic model [10]. An improved algorithm [11] was developed for the system with complete unknown dynamic

model. A model-based reinforcement learning variable impedance controller is proposed, in which model predictive control (MPC) was applied to optimize the impedance control parameters online [12]. However, the state constraints and the system stability analysis are not considered. For those systems under conditions of measurable state, the output feedback algorithm is an effective method to estimate the state information [13–15]. Rizvi established a state observer to estimate the state of the system [13]. Gao et al. reconstructed the state based on the discrete state space model [14]. The output feedback (OPFB) ADP algorithm was selected to obtain optimal solution which considers the arbitrary environment position/trajectory [15]. The OPFB ADP algorithm is an effective method to determine the optimal impedance parameters of the human–robot system in unknown environment dynamics.

In HRI tasks, humans make direct collaborations with robots. In this situation, safety is vital in HRI systems. However, the desired trajectory guided by interaction force may exceed regular working space in admittance control. In the past decades, a lot of research works has been done in the safety control field [16–20]. An integrated HRI strategy was proposed to ensure safety through a coordinated suite of safety strategies [16]. A cooperative fuzzy-impedance control with embedded velocity and force rules was proposed to enhance safety [17]. In order to improve the safety performance, an actuation mechanism was built up in ref. [18]. A task-space admittance controller was proposed in which the inertia matrix was conditional online [19]. Human–robot collaboration and control were addressed in ref. [20], including safety assessment, advanced force control, human-aware motion planning, gesture recognition, and task scheduling. The conception of barrier Lyapunov function (BLF) was put forwarded to guarantee states within certain bounds [21]. The BLF method was widely applied in the control field [22–24], which was employed to prevent the violation of the output constraints in ref. [23], and was used in the design of an adaptive impedance control [24]. In terms of the ability to fit arbitrary nonlinear curves [24–26], neural network (NN) was considered to be an effective method in control systems, which require less relative information about the system dynamics [25]. BLF-based controllers with NN approaches were studied by many researchers to ensure the safety and stability of robot [26–28].

In order to ensure the high accuracy of the impedance model, while guaranteeing the performance of the trajectory tracking, a double loops control system was proposed [29, 30]. The control method decouples the robot-specific inner loop controller from the task-specific outer loop control, which considers the safety of tracking and the factor of the environment model. Inspired by this double loop system, a new two loops control structure through the feedback of force is proposed in this paper. Different from ref. [30], the environment and robot impedance model are integrated together with linear quadratic regulation (LQR). Considering the partial unmeasurable states, an output feedback algorithm is presented to estimate the optimal control policy of LQR problem in the outer loop. By constructing the asymmetric barrier Lyapunov function (ABLF) controller in the inner loop, the stability of the system is ensured under the condition of flexible constrains in HRI system. The design of NN in adaptive controller compensates for the dynamic uncertainties. Simulations and experimental results validate the effectiveness of inner and outer loops. The primary contributions of this paper are summarized as follows:

- 1) Two-loop control structure is designed to assist the human operators to perform cooperative task with robot and optimize the tracking performance. The inner and outer loops are decoupled.
- 2) Considering the unknown dynamic and position parameters when interacting with the environment, optimal impedance adaptation method is used to obtain the optimal control law in the outer loop. The discrete output feedback algorithm realizes the optimal control of the complete robot–environment interaction system, and a discount factor is added to the algorithm to improve the convergence speed.
- 3) Robot is controlled in the inner loop to track the desired trajectory from the output of admittance control. ABLF is proposed in the design of inner loop controller to restrict the motion space and tracking errors. Compared to the symmetrical controllers [28], this method is able to adjust the upper and lower bounds to adapt to different task requirements. Meanwhile, radial basis function neural network (RBFNN) is designed to compensate for the unknown dynamics of the robot, guaranteeing the precision of admittance model.

The rest of this paper is organized as following arrangements. In Section 2, a system of HRI is described. In Section 3, an overall system structure is proposed with outer loop and inner loop, which are described, respectively. In Section 4, simulations of inner and outer loops are given to verify the proposed methods, respectively. In Section 5, experimental results of a practical task conducted on a Franka robot are shown. Conclusion is presented in Section 6.

2. System description

The HRI system includes a rigid robot manipulator and human, where the end-effector of the robot manipulator interacts with human physically. The dynamic of robot manipulator is described by

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = \tau + J^T(q)F, \tag{1}$$

where

$$q = \varphi(x), \tag{2}$$

$$\dot{q} = J^{-1}(q)\dot{x}, \tag{3}$$

$$\ddot{q} = \dot{J}^{-1}(q)\dot{x} + J^{-1}(q)\ddot{x}. \tag{4}$$

Here $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ are joint angles, velocities, and accelerations, respectively. $x, \dot{x}, \ddot{x} \in \mathbb{R}^m$ are the Cartesian positions, velocities, and accelerations of robot end effector. $M(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the Coriolis and centripetal coupling matrix, $G(q) \in \mathbb{R}^n$ is the gravity loading, $J(q) \in \mathbb{R}^{n \times m}$ is the Jacobian matrix. τ is the vector of control input, $F \in \mathbb{R}^n$ denotes the interaction force with the environment. $\varphi(\cdot)$ represents the inverse kinematics. n denotes the number of joints, and m denotes the degree of freedom (DOF).

1. The matrix $M(q)$ is symmetric and positive definite.
2. The matrix $2C(q, \dot{q}) - \dot{M}(q)$ is skew-symmetric.

From the perspective of the robot manipulator, the following impedance model is presented for the safe of HRI.

$$M_d\ddot{x} + C_d\dot{x} + G_d(x - x_d) = F, \tag{5}$$

where $M_d \in \mathbb{R}^{n \times n}$, $C_d \in \mathbb{R}^{n \times n}$, $G_d \in \mathbb{R}^{n \times n}$ are the desired inertia, damping, and stiffness matrices, respectively, and x_d denotes the desired trajectory of robot manipulator.

Remark 1: M_d, C_d , and G_d are defined under different interaction behavior requirements.

To discuss the interaction behavior, it is necessary to take the environment dynamics into consideration. Two forms of environment models are presented.

$$M_e\ddot{x} + C_e\dot{x} + G_e x = -F, \tag{6}$$

$$C_e\dot{x} + G_e x = -F, \tag{7}$$

where M_e, C_e, G_e are unknown mass, damping, and stiffness of environment models, respectively. Eq. (6) represents a mass-damping-stiffness system and (7) represents a damping-stiffness system. The difference between (6) and (7) depends on the weight of the interactive environment model. For the sake of generality, we take the mass into consideration in robot–environment model. Thus, from (5) and (6), we have

$$(M_e + M_d)\ddot{x} + (C_e + C_d)\dot{x} + G_e x + G_d(x - x_d) = 0. \tag{8}$$

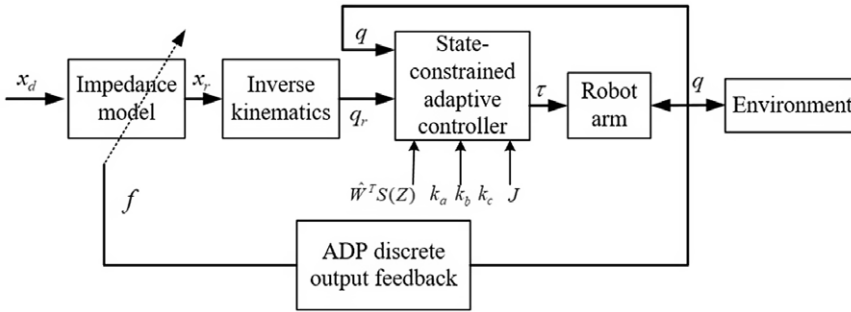


Figure 1. System structure of double-loop control.

3. Adaptive optimal control with discrete output feedback

3.1. Overall structure

The system diagram of overall structure in Fig. 1 is first designed and includes outer and inner loops. The outer loop of the system is to generate the virtual desired trajectory x_r according to the interaction force f and impedance model $Z(\cdot)$. The inner loop with an ABLF controller is designed to track the virtual desired trajectory precisely. The main purpose of the controller is to make the robot comply with the human’s objective, while guaranteeing safety and tracking accuracy.

In general, the desired impedance model in the Cartesian space is written as

$$f = Z(x_d, x_r), \tag{9}$$

where x_d is the desired trajectory and x_r is the reference trajectory in Cartesian space. Then, the virtual desired reference trajectory q_r can be obtained by using inverse kinematics. The outer loop of the system is to generate the virtual desired trajectory according to the interaction force f and the impedance model $Z(\cdot)$. The OPFB ADP method is proposed in the outer loop to determine the optimized impedance parameters of the unknown robot–environment dynamics model.

The main purpose for the inner loop is to guarantee the tracking performance of trajectory q_r . For the security of robot arm, ABLF-based controller is designed to constrain the state and speed of robot. Then, the RBFNN is employed to fit the unknown robot dynamics in the constrained controller.

3.2. OPFB ADP method in outer loop

To optimize the robot–environment interaction, an impedance adaptation method considering the environment dynamics is developed. In this paper, the robot and environmental system are taken into consideration to achieve the desired interaction performance by minimizing the following cost function:

$$\Gamma = \int_t^\infty [\dot{x}^T Q_1 \dot{x} + (x - x_d)^T Q_2 (x - x_d) + f^T R f] d\tau, \tag{10}$$

where Q_1, Q_2 are positive definite, describing the weight of velocity and tracking errors, respectively. R is the weight of the interaction force torque. The state variable is defined as

$$\xi = [\dot{x}^T, x^T, s_1^T, s_2^T]^T, \tag{11}$$

where s_1 and s_2 are two states constructed in the model, which can be written as

$$\begin{cases} \dot{s}_1 = F_1 s_1 \\ x_d = G_1 s_1 \end{cases}, \quad \begin{cases} \dot{s}_2 = F_2 s_2 \\ x_e = G_2 s_2 \end{cases}, \tag{12}$$

where F_1 and G_1 are two known matrices, F_2 and G_2 are unknown matrices. Two linear systems in (12) are utilized to determine the desired trajectory x_d and the varying position of the contact trajectory x_e .

According to (12), the cost function (10) can be rewritten as

$$\begin{aligned} \Gamma &= \int_t^\infty \left[\dot{x}^T Q_1 \dot{x} + \begin{bmatrix} x^T & x_d^T \end{bmatrix} \begin{bmatrix} Q_2 & -Q_2 \\ -Q_2 & Q_2 \end{bmatrix} \begin{bmatrix} x^T \\ x_d^T \end{bmatrix} + f^T R f \right] d\tau \\ &= \int_t^\infty [y^T Q y + f^T R f] d\tau, \end{aligned} \tag{13}$$

where $Q = \begin{bmatrix} Q_1 & \mathbf{0} \\ \mathbf{0} & Q_2' \end{bmatrix}$ with $Q_2' = \begin{bmatrix} Q_2 & -Q_2 G_1 \\ -G_1^T & G_1^T Q_2 G_1 \end{bmatrix}$.

For the convenience of calculation, a discrete output feedback method is introduced to solve adaptive dynamic programming problems. First, the continuous-time state-space function can be written as

$$\dot{\xi} = A \xi + B u, \quad y = C \xi \tag{14}$$

where $\xi \in \mathbb{R}^{n \times n}$ is the state, and $y \in \mathbb{R}^p$ is the measured output, and $u \in \mathbb{R}^m$ is the control input. In real application, u represents the interaction force f . Assume that (A, B) is controllable and (A, C) is observable. The matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ include the unknown environment dynamics due to the unmeasurable state s_2 , where

$$\begin{aligned} A &= \begin{bmatrix} -M^{-1}(C_d + C_e) & -M^{-1}G_e & 0 & M^{-1}G_e G_2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & F_1 & 0 \\ 0 & 0 & 0 & F_2 \end{bmatrix}, \\ B &= [-M^{-1} \ 0 \ 0 \ 0]^T, \quad M = M_d + M_e. \end{aligned}$$

The output matrix C is defined as

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Then, a discrete model of (14) using the zero-order hold method is obtained by taking periodic sampling

$$\xi_{k+1} = A_d \xi_k + B_d u_k, \quad y_k = C \xi_k, \tag{15}$$

where $A_d = e^{Ah}$, $B_d = (\int_0^h e^{A\tau} d\tau) B$, and $h > 0$ is a small sampling period. The stabilizing control policy is given as $u_k = \mu(\xi_k)$, which can minimize the performance index function:

$$V^\mu(\xi_k) = \sum_{i=k}^\infty (y_i^T Q_d y_i + u_i^T R_d u_i) \tag{16}$$

with weighting matrices $Q_d \geq 0$, $R_d > 0$, and $(A_d, \sqrt{Q_d}C)$ is observable and (A_d, B_d) is controllable.

Since the solution of (16) is similar to the LQR problem, we can find a positive definite matrix $P_d \in \mathbb{R}^{n \times n}$ to rewrite (16) into the following form.

$$V^\mu(\xi_k) = \xi_k^T P_d \xi_k. \tag{17}$$

Then, the LQR Bellman equation is obtained as follows:

$$\xi_k^T P_d \xi_k = y_k^T Q_d y_k + u_k^T R_d u_k + \xi_{k+1}^T P_d \xi_{k+1}. \tag{18}$$

To find the optimal control, combining (15) and (18) yields

$$\xi_k^T P_d \xi_k = y_k^T Q_d y_k + u_k^T R_d u_k + (A_d \xi_k + B_d u_k)^T P_d (A_d \xi_k + B_d u_k). \tag{19}$$

To determine the minimizing control, setting the derivative with respect to u_k to zero can obtain

$$u_k = -(R_d + B_d^T P_d B_d)^{-1} B_d^T P_d A_d \xi_k = -K_d^* \xi_k. \tag{20}$$

According to ref. [31], the Riccati equation can be derived as

$$A_d^T P_d A_d - P_d + C^T Q_d C - A_d^T P_d B_d (R_d + B_d^T P_d B_d)^{-1} B_d^T P_d A_d = 0. \tag{21}$$

For the unmeasurable states, an output feedback method is introduced to parameterize the internal state in terms of the filtered inputs and outputs of the system.

Based on the discretized system model (15), the state ξ_k is reconstructed with the previous N inputs and outputs.

$$\xi_k = E_u \bar{u}_{k-1,k-N} + E_y \bar{y}_{k-1,k-N} = [E_u \quad E_y] \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \end{bmatrix} = \Theta \bar{z}_{k-1,k-N}, \tag{22}$$

$$\bar{u}_{k-1,k-N} = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-N} \end{bmatrix} \in \mathbb{R}^{mN}, \quad \bar{y}_{k-1,k-N} = \begin{bmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-N} \end{bmatrix} \in \mathbb{R}^{pN}, \tag{23}$$

where $\bar{u}_{k-1,k-N}$ and $\bar{y}_{k-1,k-N}$ are the measured input and output sequences over the time interval $[k - N, k - 1]$, respectively, and $\bar{z}_{k-1,k-N} = [\bar{u}_{k-1,k-N}^T, \bar{y}_{k-1,k-N}^T] \in \mathbb{R}^l, l = N(m + p), \Theta = [E_u, E_y]$, and E_u, E_y are parameter matrices of two sequences, respectively, which is derived in ref. [31].

Substituting (22) into (17) yields

$$V^\mu(\xi_k) = \bar{z}_{k-1,k-N}^T \begin{bmatrix} E_u^T \\ E_y^T \end{bmatrix} P_d [E_u \quad E_y] \bar{z}_{k-1,k-N} \equiv \bar{z}_{k-1,k-N}^T \bar{P}_d \bar{z}_{k-1,k-N}, \tag{24}$$

where $\bar{P}_d = \Theta^T P_d \Theta \in \mathbb{R}^{(m+p)N \times (m+p)N}$. Then, the LQR Bellman equation is written in the form of measured data:

$$\bar{z}_{k-1,k-N}^T \bar{P}_d \bar{z}_{k-1,k-N} = y_k^T Q_d y_k + u_k^T R_d u_k + \bar{z}_{k,k-N+1}^T \bar{P}_d \bar{z}_{k,k-N+1}. \tag{25}$$

The optimal control can be determined online timely using temporal difference (TD) RL method to minimize the following Bellman TD error online [31]. Then, the Bellman TD error is defined based on Eq. (14), which can be rewritten as

$$e_k = -\bar{z}_{k-1,k-N}^T \bar{P}_d \bar{z}_{k-1,k-N} + y_k^T Q_d y_k + u_k^T R_d u_k + \bar{z}_{k,k-N+1}^T \bar{P}_d \bar{z}_{k,k-N+1}. \tag{26}$$

The optimal policy is obtained by minimizing value function in terms of the measured data.

$$\mu(\xi_k) = \underset{\mu_k}{arg \min} (y_k^T Q_d y_k + u_k^T R_d u_k + \bar{z}_{k,k-N+1}^T \bar{P}_d \bar{z}_{k,k-N+1}). \tag{27}$$

Decomposing $\bar{z}_{k,k-N+1}^T \bar{P}_d \bar{z}_{k,k-N+1}$, we have

$$\bar{z}_{k,k-N+1}^T \bar{P}_d \bar{z}_{k,k-N+1} = \begin{bmatrix} u_k \\ \bar{u}_{k-1,k-N+1} \\ \bar{y}_{k-1,k-N} \end{bmatrix}^T \begin{bmatrix} p_0 & p_u & p_y \\ p_u^T & P_{22} & P_{23} \\ p_y^T & P_{32} & P_{33} \end{bmatrix} \begin{bmatrix} u_k \\ \bar{u}_{k-1,k-N+1} \\ \bar{y}_{k-1,k-N} \end{bmatrix}, \tag{28}$$

where $p_0 \in \mathbb{R}^{m \times m}, p_u \in \mathbb{R}^{m \times m(N-1)}, p_y \in \mathbb{R}^{m \times pn}$. Then, differentiating with respect to u_k to perform minimization in (27) yields

$$u_k = -(R_d + p_0)^{-1} (p_u \bar{u}_{k-1,k-N+1} + p_y \bar{y}_{k,k-N+1}). \tag{29}$$

Theorem 1. [15]: Let $\delta^j, (j = 1, 2, \dots)$ be the maximum difference between $V^j(\xi_k)$ and $V^*(\xi_k)$. When adding a discount factor $\gamma (0 \leq \gamma \leq 1)$ to the VI algorithm, the convergence rate of the algorithm to the optimal value P^* is

$$\delta^{j+1} \leq \gamma \delta^j. \tag{30}$$

Considering the convergence speed in the iteration process of control law, a discount factor is added to the controller. Then LQR Bellman equation with discount factor is expressed as

$$\bar{z}_{k-1,k-N}^T \bar{P}_d \bar{z}_{k-1,k-N} = y_k^T Q_d y_k + u_k^T R_d u_k + \gamma \bar{z}_{k,k-N+1}^T \bar{P}_d \bar{z}_{k,k-N+1}. \tag{31}$$

The optimal control law with discount factor is written as follows:

$$u_k^* = -(R_d/\gamma + p_0)^{-1} (p_u \bar{u}_{k-1,k-N+1} + p_y \bar{y}_{k,k-N+1}). \tag{32}$$

Thus, the optimal interaction force is obtained as $f = u_k^*$.

3.3. ABLF-based NN inner loop controller

Referring to (1), and taking $x_1 = q, x_2 = \dot{q}$, the robot dynamic system can be rewritten as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= M^{-1}(x_1)[\tau + J^T(x_1)f - C(x_1, x_2)x_2 - G(x_1)], \\ y &= x_1, \end{aligned} \tag{33}$$

where $x_1 = [x_{11}, x_{12}, \dots, x_{1n}]^T$, and the desired trajectory is $x_d(t) = [q_{d1}(t), q_{d2}(t), \dots, q_{dn}(t)]^T$. We need to ensure all states and outputs are within the limits when tracking the desired trajectory, *i.e.*, $|x_{1i}| < k_{c1i}, |x_{2i}| < k_{c2i}, \forall t \geq 0$ with k_{c1i}, k_{c2i} being positive constants, where $k_{c1} = [k_{c11}, k_{c12}, \dots, k_{c1n}]^T, k_{c2} = [k_{c21}, k_{c22}, \dots, k_{c2n}]^T$ are positive constant vectors.

Lemma 1: There exists any positive constant vector $k_b \in \mathbb{R}$, for $\forall x \in \mathbb{R}$ in the interval $|x| < |k_b|$, then the following inequality holds

$$\ln \frac{k_b^{2p} - x^{2p}}{k_b^{2p} - x^{2p}} \leq \frac{x^{2p}}{k_b^{2p} - x^{2p}}, \tag{34}$$

where p is a positive integer.

For the convenience of expression, the controller proposed in the inner loop is expressed in a continuous way. In the real experiment, the torque calculated from the asymmetric constrained controller is exerted to the robot in a discrete way. So it is corresponding to the discrete outer loop.

In order to achieve the requirements of tracking constraint, an ABLF-based controller is employed in the system. The tracking errors are designed as $z_1 = [z_{11}, z_{12}, \dots, z_{1n}]^T = x_1 - x_d, z_2 = [z_{21}, z_{22}, \dots, z_{2n}]^T = x_2 - \alpha$, and we have $\dot{z}_1 = x_2 - \dot{x}_d$. The virtual control α is defined as

$$\alpha = \dot{x}_d - T, \tag{35}$$

where

$$T = \begin{bmatrix} (k_{a1} + z_{11})(k_{b1} - z_{11})k_1z_{11} \\ (k_{a2} + z_{12})(k_{b2} - z_{12})k_2z_{12} \\ \vdots \\ (k_{an} + z_{1n})(k_{bn} - z_{1n})k_nz_{1n} \end{bmatrix}. \tag{36}$$

Positive constants k_a and k_b denote the lower and upper bounds of tracking error where $-k_a < z_{1,2} < k_b$, and $k_i (i = 1, 2, \dots, n)$ denotes a positive constant. Taking the derivation of z_1 and z_2 with respect to time, we have

$$\dot{z}_1 = z_2 + \alpha - \dot{x}_d = z_2 - T, \tag{37}$$

$$\dot{z}_2 = M^{-1}(x_1)[\tau + J^T(x_1)f - C(x_1, x_2)x_2 - G(x_1)] - \dot{\alpha}. \tag{38}$$

The ABLF is defined as follows:

$$V_1 = \frac{h_1(z_1)}{2p} \sum_{i=1}^n \log \frac{k_{bi}^{2p}}{k_{bi}^{2p} - z_{1i}^{2p}} + \frac{1 - h_1(z_1)}{2p} \sum_{i=1}^n \log \frac{k_{ai}^{2p}}{k_{ai}^{2p} - z_{1i}^{2p}}, \tag{39}$$

where $k_{ai} = x_{di} - \underline{k}_{ci}$, $k_{bi} = \bar{k}_{ci} - x_{di}$, $\underline{k}_{ai} \leq k_{ai} \leq \bar{k}_{ai}$, $\underline{k}_{bi} \leq k_{bi} \leq \bar{k}_{bi}$, and $h_i(z_1)$ is given as

$$h_i(z_1) = \begin{cases} 1 & z_i > 0 \\ 0 & z_i \leq 0 \end{cases}. \tag{40}$$

The time derivative of (39) is

$$\begin{aligned} \dot{V}_1 &= h_1(z_1) \sum_{i=1}^n \frac{z_{1i}^{2p-1} \dot{z}_{1i}}{k_{bi}^{2p} - z_{1i}^{2p}} + (1 - h_1(z_1)) \sum_{i=1}^n \frac{z_{1i}^{2p-1} \dot{z}_{1i}}{k_{ai}^{2p} - z_{1i}^{2p}} \\ &= -h_1(z_1) \sum_{i=1}^n k_i z_{1i}^{2p} + h_1(z_1) \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{bi}^{2p} - z_{1i}^{2p}} - (1 - h_1(z_1)) \sum_{i=1}^n k_i z_{1i}^{2p} + (1 - h_1(z_1)) \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{ai}^{2p} - z_{1i}^{2p}} \\ &= -\sum_{i=1}^n k_i z_{1i}^{2p} + h_1(z_1) \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{bi}^{2p} - z_{1i}^{2p}} + (1 - h_1(z_1)) \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{ai}^{2p} - z_{1i}^{2p}}. \end{aligned} \tag{41}$$

Then, a second Lyapunov function is designed as

$$V_2 = V_1 + \frac{1}{2} z_2^T M(x_1) z_2. \tag{42}$$

Take the time derivation of V_2 to obtain

$$\dot{V}_2 = \dot{V}_1 + z_2^T [\tau + J^T(x_1)f - C(x_1, x_2)\alpha + G(x_1) - M(x_1)\dot{\alpha}]. \tag{43}$$

When $z_2 = [0, 0, \dots, 0]^T$, $\dot{V}_2 = -\sum_{i=1}^n k_i z_{1i}^{2p} < 0$. According to Lyapunov stability theory, the system is asymptotically stable. On the contrary, according to the property of the Moore-Penrose pseudoinverse, if $z_2 \neq [0, 0, \dots, 0]^T$, $z_2^T(z_2^T)^+ = 1$. K_2 is a designed positive gain matrix with $K_2 = K_2^T > 0$. Then the model-based controller is designed as

$$\begin{aligned} \tau_0 &= -J^T(x_1)f + C(x_1, x_2)\alpha + G(x_1) + M(x_1)\dot{\alpha} - K_2 z_2 \\ &\quad - h_1(z_1)(z_2^T)^+ \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{bi}^{2p} - z_{1i}^{2p}} - (1 - h_1(z_1))(z_2^T)^+ \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{ai}^{2p} - z_{1i}^{2p}}. \end{aligned} \tag{44}$$

Substituting (44) into (43) yields

$$\dot{V}_2 = -\sum_{i=1}^n k_i z_{1i}^{2p} - z_2^T K_2 z_2 = -(z_1^T)^p K_1 z_1^p - z_2^T K_2 z_2 < 0. \tag{45}$$

where $K_2 = K_2^T = \text{diag}([k_1, k_2, \dots, k_n]) > 0$ is a diagonal matrix.

Eq.(44) is model-based, in real applications, the dynamic parameters $C(x_1, x_2)$, $G(x_1)$, $M(x_1)$, and f are difficult to be obtained. To address uncertainties in the dynamic model, an adaptive NN method is employed to fit the uncertain terms. The adaptive NN control is proposed as

$$\begin{aligned} \tau_1 &= -h_1(z_1)(z_2^T)^+ \sum_{i=1}^n \frac{k_i z_{1i}^{2p}}{k_{bi}^{2p} - z_{1i}^{2p}} - (1 - h_1(z_1))(z_2^T)^+ \sum_{i=1}^n \frac{k_i z_{1i}^{2p}}{k_{ai}^{2p} - z_{1i}^{2p}} - h_1(z_1)(z_2^T)^+ \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{bi}^{2p} - z_{1i}^{2p}} \\ &\quad - (1 - h_1(z_1))(z_2^T)^+ \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{ai}^{2p} - z_{1i}^{2p}} - K_2 z_2 - \hat{W}^T S(Z) - J^T(x_1)f. \end{aligned} \tag{46}$$

where $S(\cdot)$ is the basis function, $\hat{W} \in R^{l \times n}$ is the estimation weight in which l is the node number of NN, and $Z = [x_1^T, x_2^T, \alpha^T, \dot{\alpha}^T]^T$ is the input of the NN basis function. With the property of approximating any nonlinearity, $\hat{W}^T S(Z)$ is the estimation of $W^{*T} S(Z)$, while the $W^{*T} S(Z)$ represents the real value of unknown term in model-based control (45) with

$$W^{*T} S(Z) + \varepsilon = -C(x_1, x_2)\alpha - G(x_1) - M(x_1)\dot{\alpha}. \tag{47}$$

The adaptive law is designed as

$$\dot{\hat{W}}_i = \Gamma_i \left[S_i(\mathbf{Z})z_{2i} - \mathbf{e}_i |z_{2i}| \dot{\hat{W}}_i \right]. \tag{48}$$

Then, the BLF V_3 is designed as

$$V_3 = V_3 + \frac{1}{2} \sum_{i=1}^n \tilde{W}_i^T \Gamma_i^{-1} \tilde{W}_i, \tag{49}$$

where $\tilde{W}_i = \hat{W}_i - W_i^*$ and $\tilde{W}_i, \hat{W}_i, W_i^*$ are the NN weight error, approximation, and ideal value, respectively. Differentiating (49), we have

$$\begin{aligned} \dot{V}_3 = & - \sum_{i=1}^n k_i z_{1i}^{2p} + h_1(\mathbf{z}_1) \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{bi}^{2p} - z_{1i}^{2p}} + (1 - h_1(\mathbf{z}_1)) \sum_{i=1}^n \frac{z_{1i}^{2p-1} z_{2i}}{k_{ai}^{2p} - z_{1i}^{2p}} \\ & + \sum_{i=1}^n \tilde{W}_i^T \Gamma_i^{-1} \dot{\hat{W}}_i + z_2^T [\tau_1 + J^T(\mathbf{x}_1)\mathbf{f} - \mathbf{C}(\mathbf{x}_1, \mathbf{x}_2)\boldsymbol{\alpha} - \mathbf{G}(\mathbf{x}_1) - \mathbf{M}(\mathbf{x}_1)\dot{\boldsymbol{\alpha}}]. \end{aligned} \tag{50}$$

If $\mathbf{z}_2 \neq [0, 0, \dots, 0]^T$, according to (46)–(48), we have

$$\begin{aligned} \dot{V}_3 \leq & - \sum_{i=1}^n k_i z_{1i}^{2p} - z_2^T \mathbf{K}_2 z_2 - h_1(\mathbf{z}_1) \sum_{i=1}^n \frac{k_i z_{1i}^{2p}}{k_{bi}^{2p} - z_{1i}^{2p}} - (1 - h_1(\mathbf{z}_1)) \sum_{i=1}^n \frac{k_i z_{1i}^{2p}}{k_{ai}^{2p} - z_{1i}^{2p}} \\ & + \sum_{i=1}^n \left[\tilde{W}_i^T S_i(\mathbf{Z})z_{2i} - e_i |z_{2i}| \tilde{W}_i^T \hat{W}_i \right] + z_2^T \varepsilon(\mathbf{Z}) + z_2^T [-\hat{W}^T S(\mathbf{Z}) + \mathbf{W}^{*T} S(\mathbf{Z})] \\ \leq & -z_2^T \left(\mathbf{K}_2 - \frac{3}{4} \mathbf{I} \right) z_2 - h_1(\mathbf{z}_1) \sum_{i=1}^n k_i \ln \frac{k_{bi}^{2p}}{k_{bi}^{2p} - z_{1i}^{2p}} - (1 - h_1(\mathbf{z}_1)) \sum_{i=1}^n k_i \ln \frac{k_{ai}^{2p}}{k_{ai}^{2p} - z_{1i}^{2p}} \\ & + \frac{1}{2} \|\varepsilon(\mathbf{Z})\|^2 + \sum_{i=1}^n \frac{e_i^2}{4} (\|\mathbf{W}_i^*\|^4 + \|\tilde{W}_i\|^4 - 2\|\mathbf{W}_i^*\|^2 \|\tilde{W}_i\|^2) \leq -\rho V_3 + \mathbf{C}, \end{aligned} \tag{51}$$

where

$$\rho = \min \left(\min(2k_i), \min(2(1 - h_1(\mathbf{z}_1))k_i), \frac{2\lambda_{\min}(\mathbf{K}_2 - \frac{3}{4}\mathbf{I})}{\lambda_{\max}(\mathbf{M})}, \min \left(\frac{z_i^2 \|\mathbf{W}_i^*\|^2}{\lambda_{\max}(\Gamma_i^{-1})} \right) \right), \tag{52}$$

$$\mathbf{C} = \frac{1}{2} \|\varepsilon(\mathbf{Z})\|^2 + \sum_{i=1}^n \left(\frac{z_i^2}{4} \|\mathbf{W}_i^*\|^4 + \frac{z_i^2}{4} N^4 \right). \tag{53}$$

Here $\lambda_{\min}(\bullet)$ and $\lambda_{\max}(\bullet)$ denote the minimum and maximum eigenvalues of the matrix, respectively, and

$$\min(2k_i) > 0, \lambda_{\min} \left(\mathbf{K}_2 - \frac{3}{4} \mathbf{I} \right) > 0, N = \left(\frac{\lambda_{\max}(\Gamma_i)}{\lambda_{\min}(\Gamma_i)} \right)^{\frac{1}{2}} \frac{\|\mathbf{s}_i\|}{z_i} + \|\mathbf{W}_i^*\| \tag{54}$$

with $\|S_i(\mathbf{Z})\| \leq \|\mathbf{s}_i\|$ and $\mathbf{s}_i > 0$.

Theorem 2 Consider the dynamic system described in (1). If the initial conditions satisfy $|x_{1i}(0)| < k_{c1i}, |x_{2i}(0)| < k_{c2i}$, the control law (46) ensures that all error signals are semi-globally uniformly bounded. Then the position and velocity constraints are not violated, that is, $\forall t > 0, |x_{1i}(t)| \leq k_{c1i}, |x_{2i}(t)| \leq k_{c2i}$. The closed-loop error signals $\mathbf{z}_1, \mathbf{z}_2$ and $\tilde{\mathbf{W}}$ remain in compact sets $\Omega_{z_1}, \Omega_{z_2}, \Omega_w$, respectively.

$$\begin{aligned} \Omega_{z_1} := & \left\{ \mathbf{z}_1 \in \mathbb{R}^n \mid -\sqrt[2p]{k_{ai}^{2p} (1 - e^{-D})} \leq z_{1i} \leq \sqrt[2p]{k_{bi}^{2p} (1 - e^{-D})}, i = 1, \dots, n \right\} \\ \Omega_{z_2} := & \left\{ \mathbf{z}_2 \in \mathbb{R}^n \mid \|\mathbf{z}_2\| \leq \sqrt{\frac{D}{\lambda_{\min}(\mathbf{M})}} \right\}, \Omega_w := \left\{ \tilde{\mathbf{W}} \in \mathbb{R}^{l \times n} \mid \|\tilde{\mathbf{W}}\| \leq \sqrt{\frac{D}{\lambda_{\min}(\Gamma^{-1})}} \right\}, \end{aligned} \tag{55}$$

Table I. Parameters of the 2-DOF robot manipulator.

Parameter	Description	Value
m_1	Mass of link 1	2.0kg
l_1	Length of link 1	0.35m
I_1	Inertia of link 1	$6.125 \times 10^{-2} \text{kg} \cdot \text{m}^2$
m_2	Mass of link 2	0.85kg
l_2	Length of link 2	0.31m
I_2	Inertia of link 2	$2.402 \times 10^{-2} \text{kg} \cdot \text{m}^2$

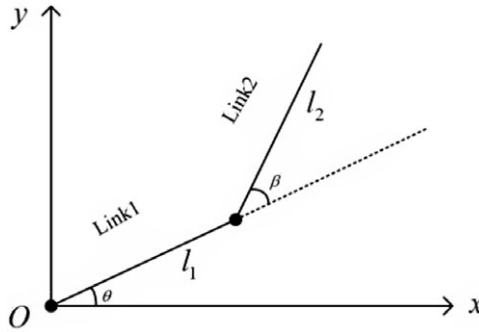


Figure 2. Diagram of the 2-DOF robot system.

where $D = 2(V_3(0) + C/\rho)$.

Proof: Multiplying $e^{\rho t}$ on both sides of (51), we have

$$e^{\rho t} \dot{V}_3 \leq -\rho V_3 e^{\rho t} + C e^{\rho t}, \tag{56}$$

and

$$d(e^{\rho t} V_3)/dt \leq C e^{\rho t} \tag{57}$$

Then, we have

$$V_3 \leq \left(V_3(0) - \frac{C}{\rho} \right) e^{-\rho t} + \frac{C}{\rho} \leq V_3(0) + \frac{C}{\rho}. \tag{58}$$

According to (58) and the property of inequality in ref. [32], we have

$$\begin{aligned} -\frac{1}{2} \sum_{i=1}^n \ln \frac{k_{ai}^{2p}}{k_{ai}^{2p} - z_{1i}^{2p}} &\leq V_3(0) + C/\rho \leq \frac{1}{2} \sum_{i=1}^n \ln \frac{k_{bi}^{2p}}{k_{bi}^{2p} - z_{1i}^{2p}} \\ \frac{1}{2} \|z_2\|^2 &\leq \frac{V_3(0) + C/\rho}{\lambda_{\min}(M)} \\ \frac{1}{2} \|\tilde{W}\|^2 &\leq \frac{V_3(0) + C/\rho}{\lambda_{\min}(\Gamma^{-1})} \end{aligned} \tag{59}$$

4. Simulations

In the simulations, a two-link robotic manipulator simulation platform is constructed as shown in Fig. 2 and the parameters of the robot are defined in Table I. The simulations of inner loop and outer loop are exerted separately to validate their own effectiveness.

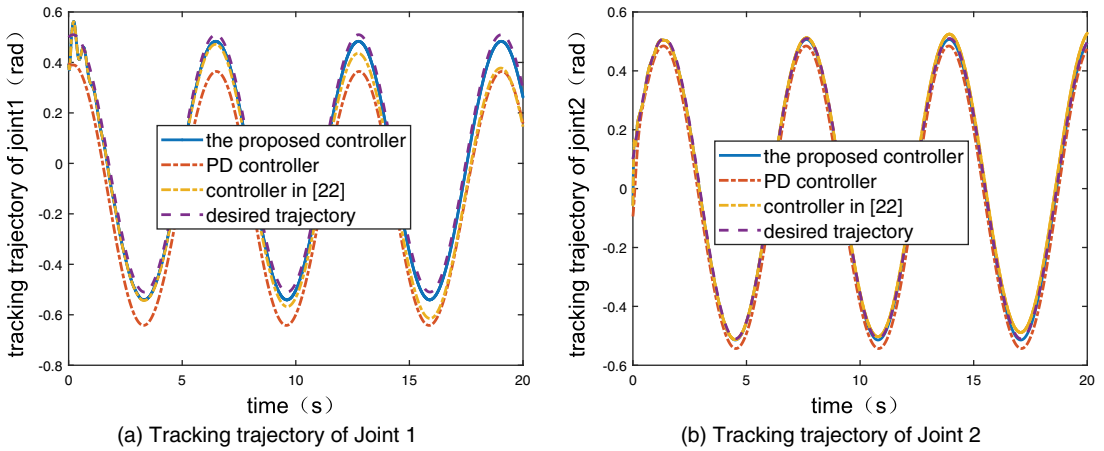


Figure 3. Tracking performance of different controllers of Joints 1 and 2.

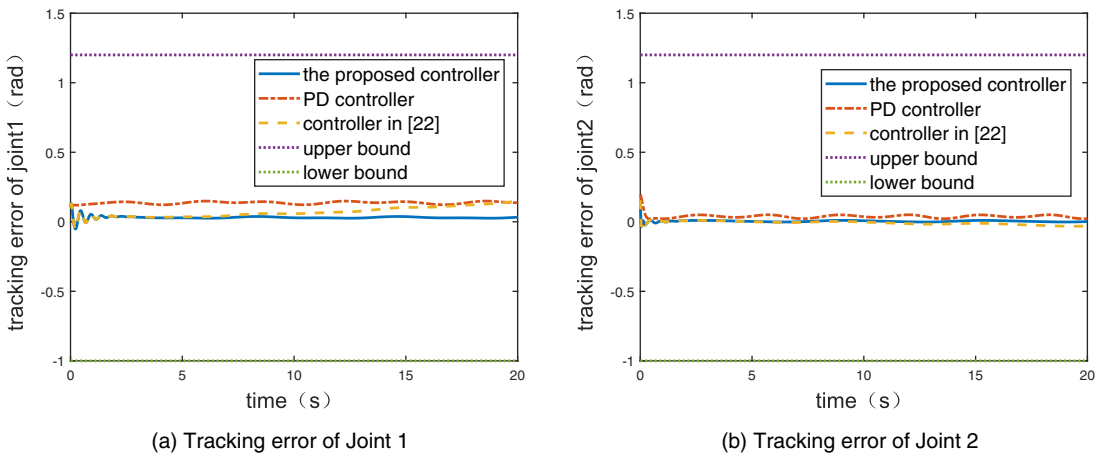


Figure 4. Tracking error of different controllers of Joints 1 and 2.

4.1. Inner loop

In order to validate the effectiveness and feasibility of the designed asymmetric constrained controller. Simulations using the proposed controller, PD controller, and controller in ref. [33] are carried in MATLAB.

The initial positions of the robot are given as $q_1(0) = [0.4, -0.1](rad)$, $q_2(0) = [0, 0](rad/s)$. The interaction force with end-effector is $f(t) = [\sin(t) + 1, \cos(t) + 0.5](N)$, where $t \in [0, t_f]$ and $t_f = 20s$. The desired trajectory is set as $q_r = [0.1 \sin(t) + 0.5 \cos(t), 0.1 \cos(t) + 0.5 \sin(t)](rad)$. The limits of tracking errors are $k_a = [-1.0, -1.0](rad)$, $k_b = [1.2, 1.2](rad)$, and the physical limit of robot joints is $k_c = [1.8, 1.8](rad)$.

For PD control, the controller is designed as $\tau = -G_1(q_1 - q_r) - G_2(q_2 - \dot{q}_r)$ where $G_1 = diag(50, 30)$ and $G_2 = diag(30, 3)$.

For the proposed controller, Eq. (46) is the control law applied in the inner loop. $S(\cdot)$ is defined as a Gaussian function. As for the gain of the NN, σ is chosen as 0.2, the number of the NN nodes is 2^8 , and the centers are selected in the area of $[-1, 1] \times 5$. The variance of centers is chosen as 0.5, and the initial value of the NN weight matrix is defined as 0. The control parameters are set as $K_2 = diag(3, 3)$, $k_1 = k_2 = 50$, $\Gamma_1 = \Gamma_2 = 100I_{256 \times 256}$ after regulating.

The tracking results are shown in Figs. 3 and 4. From Fig. 4, the tracking errors of PD controller remain relatively large, and the tracking errors of the controller in ref. [33] are smaller. It is seen that

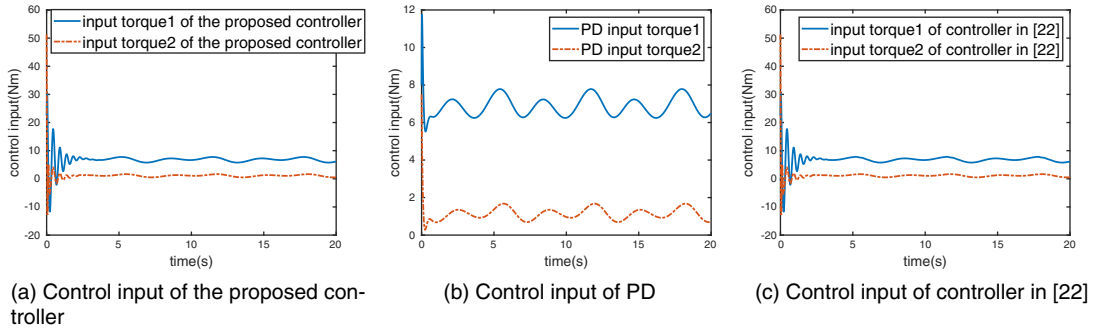


Figure 5. Control input of three control methods.

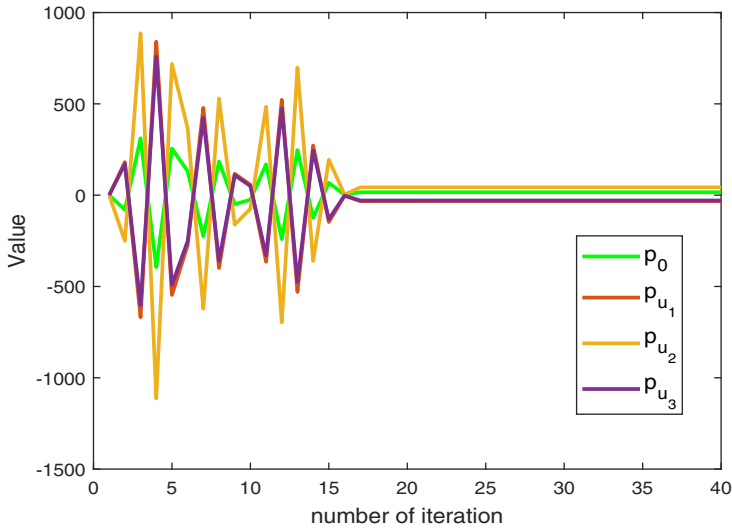


Figure 6. Convergence of the parameters p_0 and p_u .

the tracking errors are almost close to zero using our proposed method, which is better than the other methods. The comparison results show that our proposed method improves the tracking performance compared with PD controller and the controller in ref. [33]. The control input of the three methods is shown in Fig. 5. The torque of the proposed controller is relatively stable.

4.2. Outer loop

In the outer loop, a robot manipulator with two revolute joints physically interacting with the human/environment is considered. The simulation is implemented in MATLAB.

In the simulation, the sample interval is set to 1 ms. The adaptation procedure includes three steps. Firstly, the exploration noise is added to the initial control input to guarantee the PE conditions [10], which is chosen as $e_k = \sum_{\omega=1}^{10} (0.06/\omega) \sin(k\omega)$. Then, optimal impedance learning is conducted until the criterion is satisfied, which is $|\bar{P}_d^{j+1} - \bar{P}_d^j| < 0.001$. Finally, the optimized impedance model is acquired through iterations.

The parameters in (10) are defined as $Q_1 = 1, Q_2 = 0.3, R = 0.1$. In the simulation, the environment model is selected as $\ddot{x} + 10\dot{x} + 100(x - x_e) = -F$. The environment position and robot desired trajectory in the Cartesian space are given by (12), with $F_1 = -1, G_1 = 0.3, F_2 = -1, G_2 = 1$. All the elements of the initial matrix \bar{P}_d are selected as 0.01, and γ is set as 0.3. Figures 6 and 7 show the convergence of \bar{P}_d, p_0, p_y, p_u reach convergence after 17 steps.

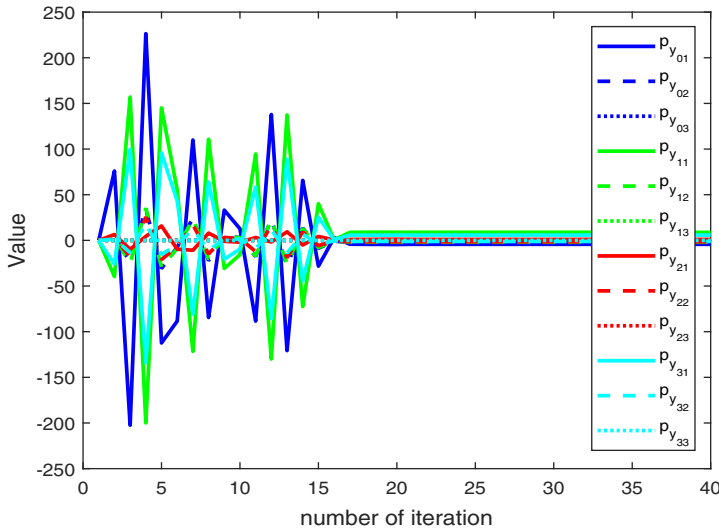


Figure 7. Convergence of the parameters p_{y0} , p_{y1} , p_{y2} and p_{y3} .

The optimal values of \bar{P}_d are shown as follows:

$$\begin{aligned}
 p_0^* &= 15.9258, \quad p_{u1}^* = -32.8697, \quad p_{u2}^* = 42.5815, \quad p_{u3}^* = -29.1587 \\
 p_{y0}^* &= [-4.2895, -2.2678, 0], \quad p_{y1}^* = [8.8085, -2.0308, 0] \\
 p_{y2}^* &= [1.0536, -1.5435, 0], \quad p_{y3}^* = [5.7937, -1.1972, 0].
 \end{aligned}$$

Then, the optimal matrix P_d is obtained from \bar{P}_d according to (24) as follows:

$$P_d = \begin{bmatrix} -7.5863 & -18.5025 & 9.2325 & 1.9694 \\ -18.5025 & -4.2168 & -18.8880 & 11.1250 \\ 9.2325 & -18.8880 & 19.6568 & -3.7325 \\ 1.9694 & 11.1250 & -3.7325 & -1.7415 \end{bmatrix}. \tag{60}$$

For comparison, the ideal optimal matrix P_d^* is calculated by using the model information as follows:

$$P_d^* = \begin{bmatrix} -7.26 & -18.3032 & 9.2331 & 1.8680 \\ -18.3032 & -3.9880 & -18.9033 & 11.0891 \\ 9.2331 & -18.9033 & 19.64 & -3.875 \\ 1.8680 & 11.0891 & -3.875 & -1.731 \end{bmatrix}. \tag{61}$$

As we know, the matrix P_d^* is the optimal solution to minimize the cost function. From the two matrices, it is obvious that the optimal matrix from our method is close to the real solution, which validates the correctness of the proposed algorithm.

Remark 2: Since this method is model-free and data-based, the parameters have few influences on the results. There is almost no need to tune them. In order to further improve the system performance in practical applications, the control parameters can be fine-tuned by adjusting the parameters in the inner and outer loops separately.

(a) Direct interaction



(b) Indirect interaction

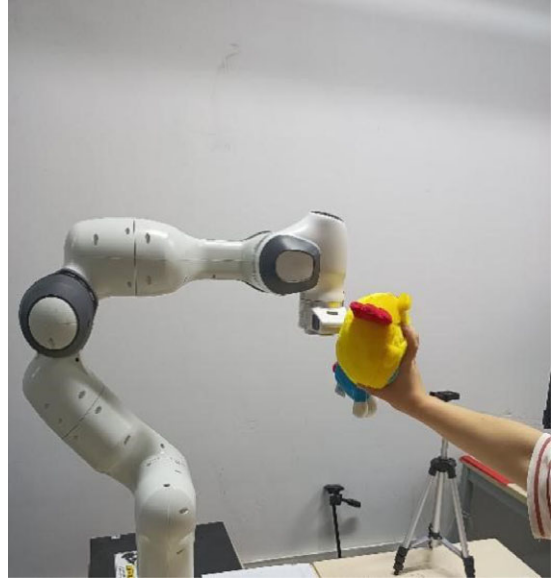


Figure 8. Human–robot collaboration task.

5. Experiment

The experiments are carried out for the robot (Franka Emika Panda, Franka Emika GmbH, Germany) shown in Fig. 8. The robot has 7 degrees of freedom (DOFs) with seven flexible joints equipped with angle and torque sensors. The communication frequency is 1000 Hz. The robot has three control modes including velocity mode, position mode, and torque mode. The operation system is Ubuntu 16.04 with the platform ROS Kinetic. When human moves the robot, it will generate a desired trajectory. Then the virtual trajectory is calculated through the impedance model in real time. There are two independent control nodes in ROS which are employed for calculating and controlling the robot, and the communication frequency between the two nodes is 200 Hz. The first node is used to calculate the adaptive constraint controller and publish the control result to the second node. Then, the second node subscribes to the information from the robot and publishes the control signals to control the robot.

In order to verify the effectiveness of the proposed controller, a trajectory tracking task is conducted. For the convenience of the experiment, a desired trajectory is defined by human in the outer loop. As shown in the Fig. 1, the virtual desired trajectory x_r , which calculated from impedance model is the input of the inner loop. It is adapted online on the basis of the human interaction with the robot. The parameters of the controller are chosen as follows: $S(\cdot)$ is the Gaussian function, $l = 2^8$ is the node numbers of NN, and the centers of each node are selected in the area of $[-1, 1]$, the center variance is set as 1.5. The initial weight \hat{W} of NN is 0, the controller parameters $K_2 = \text{diag}(10, 10, 10, 10, 10, 10, 10)$ and the constraint error of tracking are set as $k_{b1} = 2, k_{b2} = 2, k_{b3} = 2, k_{b4} = 3, k_{b5} = 2, k_{b6} = 2.6, k_{b7} = 1.5, k_{a1} = -1.2, k_{a2} = -1.2, k_{a3} = -1.5, k_{a4} = -2, k_{a5} = -2, k_{a6} = -2, k_{a7} = -1.8, \sigma_i = 0.02, \Gamma_i = 200I_{16 \times 16}, (i = 1, 2, \dots, 7)$.

Figure 9 shows the tracking trajectories of the three methods, it is obvious that the tracking performance of the proposed controller is better than others. Figure 10 shows the tracking errors in each direction, where green line is the zero baseline. Compared with the other two methods, the trajectory error of our proposed controller is smaller. The main verification metric for the accuracy of the controller is the tracking error. The average error of three controllers is provided in Fig. 11. Table II shows the average error and error variance of the three methods. It is obvious that the average error and error variance of the proposed controller are smaller than the other two methods. Moreover, we present the

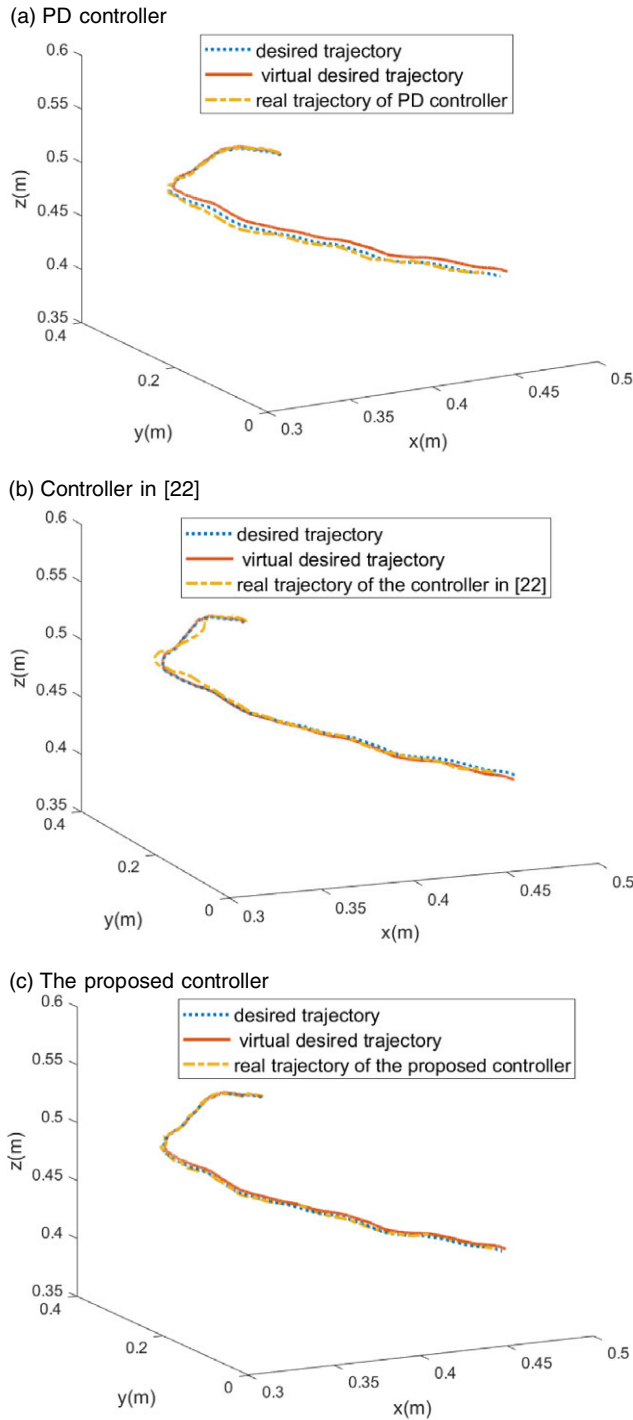


Figure 9. Tracking trajectory of the Franka robot.

index of smoothness index (as described in ref. [17]) of three controllers in Fig. 12, which demonstrates that the proposed controller is smoother than the other two methods. The calculation input torque result of the adaptive constraint controller is shown in Fig. 13.

Table II. Error comparison of three controllers.

Controllers	Average error	Error variance
PD controller	0.0084	1.0567
Controller in ref. [33]	0.0092	1.0564
The proposed controller	0.0034	0.1758

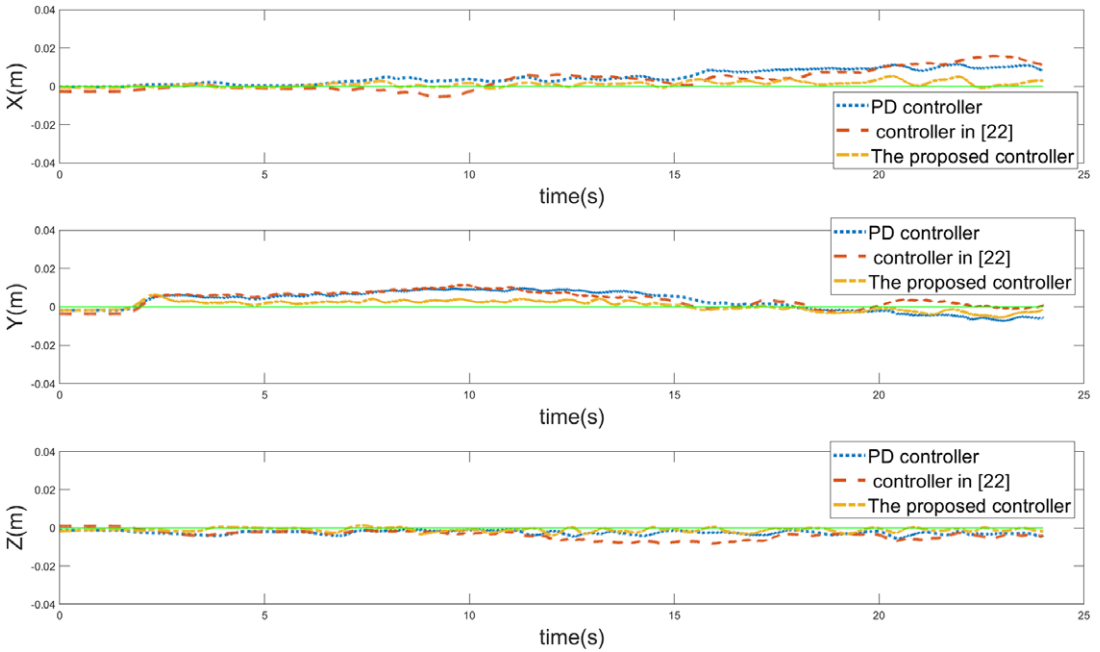


Figure 10. The tracking errors of different methods.

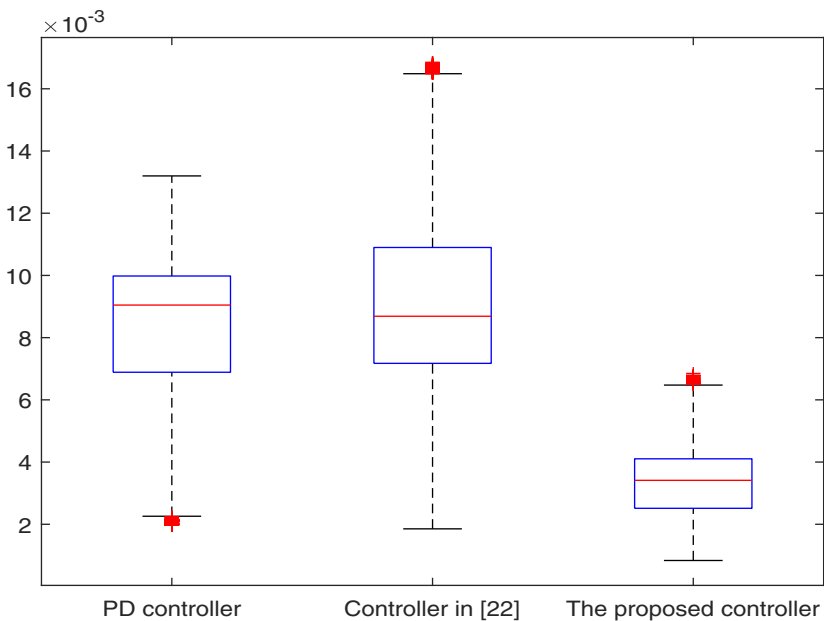


Figure 11. Average error of three controllers.

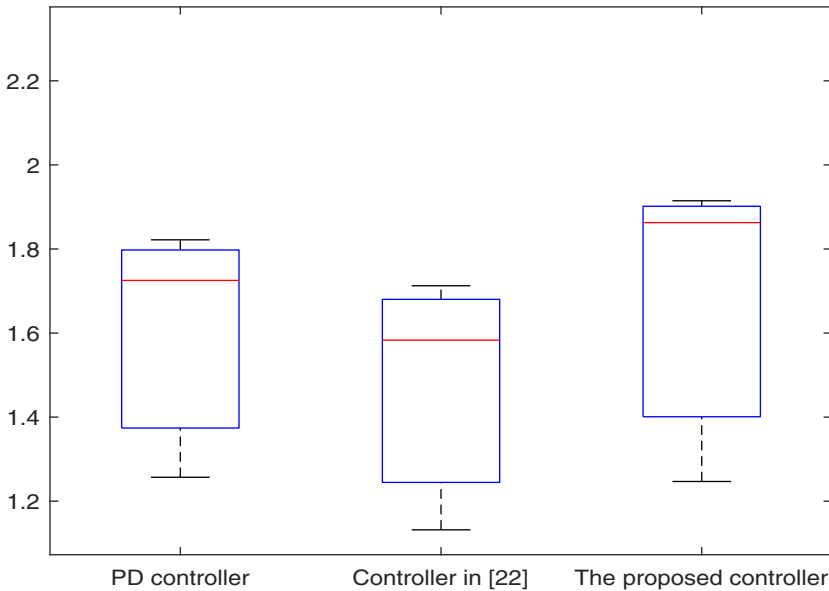


Figure 12. Smoothness of three controllers.

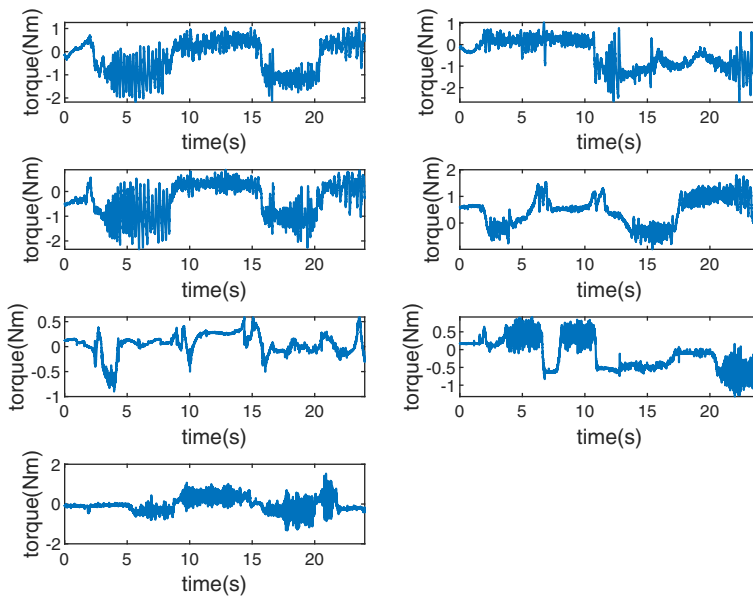


Figure 13. Input torques of Joints 1–7.

The calculated control torque is exerted in the robot when a HRI task is conducted on the Franka robot. The outer loop aims at optimizing the parameters of impedance model. For the cost function (10), the parameters Q_1 , Q_2 and R are set to 1. The stopping criterion is $|\bar{\mathbf{P}}_d^{j+1} - \bar{\mathbf{P}}_d^j| < 0.01$, and the discount factor γ is selected as 0.3. The initial value of p_0 is set as 2, and the initial values of \mathbf{p}_{y0} is set as $[-2 \ 10 \ 10]$. The initial values of \mathbf{p}_u are set as 1, and then all the other elements of the initial matrix $\bar{\mathbf{P}}_d$ are 0. The sampling interval is set as 10 ms.

Figures 14 and 15 show that the parameters \mathbf{p}_y , \mathbf{p}_0 and \mathbf{p}_u reach convergence after 1000 iteration. The optimal values of $\bar{\mathbf{P}}_d$ are shown as follows:

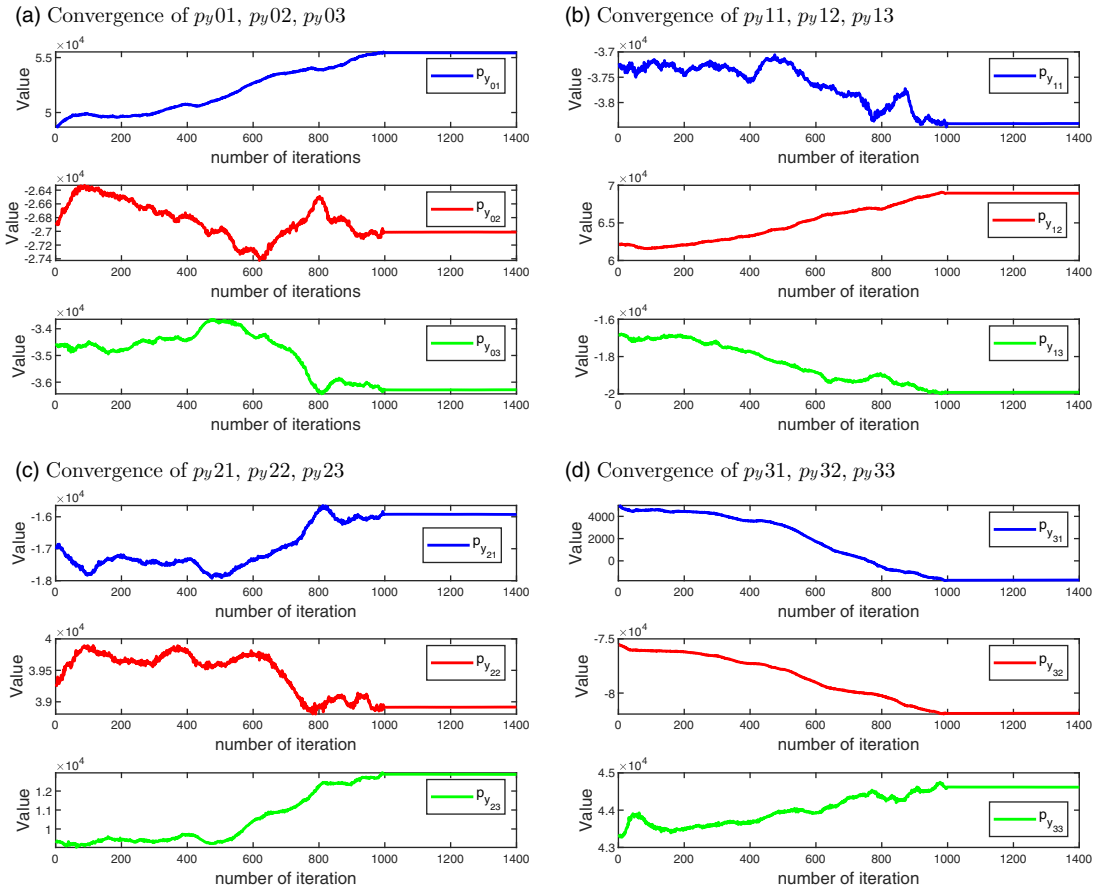


Figure 14. Convergence of the parameters p_y in matrix \bar{P}_d .

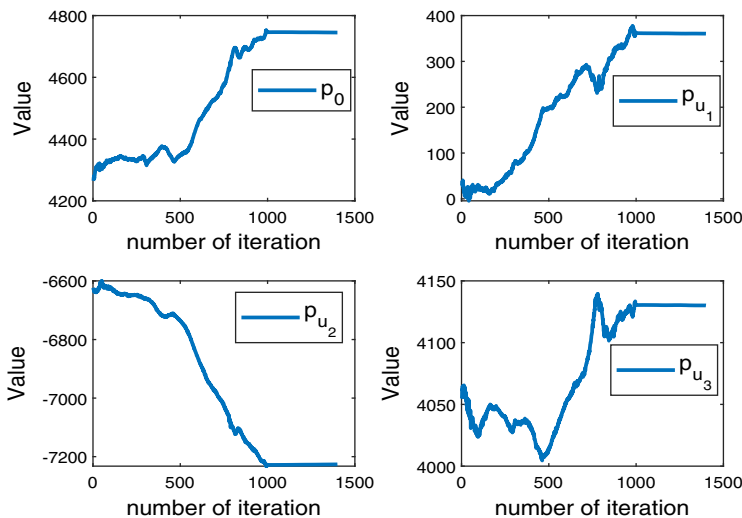


Figure 15. Convergence of the parameters p_0, p_u in matrix \bar{P}_d .

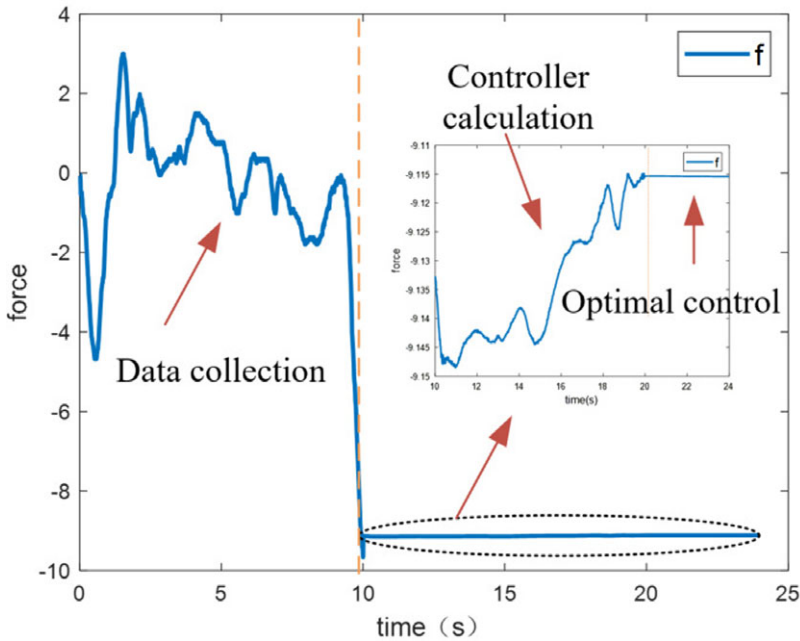


Figure 16. Interaction force of three processes.

$$\begin{aligned}
 p_0^* &= 4746.2, p_{u1}^* = 360.8, p_{u2}^* = -7727.4, p_{u3}^* = 4130.3 \\
 \mathbf{p}_{y0}^* &= [5.54, -2.7, -3.63] \times 10^4, \mathbf{p}_{y1}^* = [-3.84, 6.89, -1, 99] \times 10^4 \\
 \mathbf{p}_{y2}^* &= [-1.59, 3.89, 1.29] \times 10^4, \mathbf{p}_{y3}^* = [-0.173, -8.19, 4.46] \times 10^4.
 \end{aligned}$$

Figure 16 shows the interaction force, where the interval 0–10s is the data collection stage, and the interval 10–20s is the calculation stage. After 20s, the optimal control is exerted in the end-effector of the robot which is more stable to complete the given task.

6. Conclusion

In this paper, to realize safe HRI, a novel impedance adaptation method is proposed in unknown environment. A double-loop system is constructed in which the inner loop improves the tracking accuracy with the safety of interaction and the outer loop fits the uncertainty of model. An adaptive dynamic programming with discrete output feedback is proposed in the outer loop to solve the problem of uncertain environment position. A discount factor is added for faster convergence speed. In the interaction process, the robot is limited in safe space through an ABLF-based inner loop controller. Meanwhile, the RBFNN compensates for the unknown dynamics in the design of the controller. Simulation and experimental results verify the effectiveness of the proposed controller in HRI system. In future works, a reduced-order observer will be added to reduce the dimensionality of states so as to increase the speed of calculation.

Author contributions statement. Xinyi Yu and Huizhen Luo conceived and designed the study. Huizhen Luo and Shuanwu Shi did the experiments. Wei Yan and Linlin Ou modify the article and verified correctness.

Financial support. This work was supported by National Key R & D Program of China (2018YFB1308400) and Natural Science Foundation of Zhejiang province (LY21F030018).

Conflicts of interest. The authors declare no conflicts of interest.

Ethical approval. This research did not require any ethical approval.

References

- [1] L. Rovedal, G. Pallucca, N. Pedrocchi, F. Brafhin and L. M. Tosatti, "Iterative learning procedure with reinforcement for high-accuracy force tracking in robotized tasks," *IEEE Trans. Ind. Inform.* **14**(4), 1753–1763 (2018).
- [2] M. Zhang, A. Mcdaid, J. A. Veale, Y. Peng and S. Q. Xie, "Adaptive trajectory tracking control of a parallel ankle rehabilitation robot with joint-space force distribution," *IEEE Access* **7**, 85812–85820 (2019).
- [3] K. Murata, H. Yamazoe, M. G. Chung and J. H. Lee, "Elderly Care Training Robot for Quantitative Evaluation of Care Operation Development of Robotic Elbow Joint to Imitate Elderly People's Elbow," **In: IEEE/SICE International Symposium on System Integration** (2017).
- [4] M. Sharifi, S. Behzadipour and G. Vossoighi, "Nonlinear model reference adaptive impedance control for human-robot interactions," *Control Eng. Pract.* **32**, 9–27 (2014).
- [5] M. Dohring and W. Newman, "The Passivity of Natural Admittance Control Implementations," **In: 20th IEEE International Conference on Robotics and Automation** (2003) pp. 3710–3715.
- [6] N. Hogan, "Impedance control: An approach to manipulation: Part II-Implementation," *J. Dyn. Syst. Meas. Control* **107**(1), 8–16 (1985).
- [7] J. E. Colbaugh and N. Hogan, "Robust control of dynamically interacting systems," *Int. J. Control* **48**(1), 65–88 (1988).
- [8] W. S. Newman, "Stability and performance limits of interaction controllers," *J. Dyn. Syst. Meas. Control* **114**(4), 563–570 (1992).
- [9] S. P. Buerger and N. Hogan, "Complementary stability and loop shaping for improved human-robot interaction," *IEEE Trans. Robot* **23**(2), 232–244 (2007).
- [10] D. Vrabe and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Netw.* **22**(3), 237–246 (2009).
- [11] Y. Jiang and Z. P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica* **48**(10), 2699–2704 (2012).
- [12] L. Roveda, J. Maskani, P. Franceschi, A. Arash and N. Pedrocchi, "Model-based reinforcement learning variable impedance control for human-robot collaboration," *J. Intell. Robot. Syst.* **100**, 417–433 (2020).
- [13] S. A. A. Rizvi and Z. L. Lin, "Reinforcement learning-based linear quadratic regulation of continuous-time systems using dynamic output feedback," *IEEE Trans. Cybern.* **50**(11), 4670–4679 (2020).
- [14] W. Gao, Y. Jiang, Z. P. Jiang and T. Chai, "Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming," *Automatica* **72**(3), 37–45 (2016).
- [15] X. Liu, S. S. Ge, F. Zhao and X. S. Mei, "Optimized impedance adaptation of robot manipulator interacting with unknown environment," *IEEE Trans. Control Syst. Technol.* **29**(1), 411–419 (2021).
- [16] A. Al-Tamimi, F. L. Lewis and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica* **43**(3), 473–481 (2007).
- [17] L. Roveda, S. Haghshenas, M. Caimmi, N. Pedrocchi and L. M. Tosatti, "Assisting operators in heavy industrial tasks: on the design of an optimized cooperative impedance fuzzy-controller with embedded safety rules," *Front. Robotics AI* **6**(75), 1–31 (2019).
- [18] M. R. Ahmed and I. Kalaykov, "Static and Dynamic Collision Safety for Human Robot Interaction Using Magnetorheological Fluid Based Compliant Robot Manipulator," **In: IEEE International Conference on Robotics and Biomimetics** (2010) pp. 370–375.
- [19] M. D. A. Fonseca, B. V. Adorno and P. Fraisse, "Task-space admittance controller with adaptive inertia matrix conditioning," *J. Intell. Robot. Syst.* **101**(2), 1 (2021).
- [20] F. Vicentini, N. Pedrocchi, M. Beschi, M. Giussani and G. Fogliazza, "PIROS: Cooperative, Safe and Reconfigurable Robotic Companion for CNC Pallets load/unload Stations," **In: Bringing Innovative Robotic Technologies From Research Labs to Industrial End-users** (Springer, Cham, 2020) pp. 57–96.
- [21] K. B. Ngo, R. Mahony and Z. P. Jiang, "Integrator Backstepping Using Barrier Functions for Systems with Multiple State Constraints," **In: IEEE Conference on Decision Control/European Control Conference (CCD-ECC)** (2005) pp. 8306–8312.
- [22] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath and P. Tabuada, "Control Barrier Functions: Theory and Applications," **In: European Control Conference** (2019) pp. 3420–3431.
- [23] S. Zhang, S. T. Xiao and W. L. Ge, "Approximation-based Control of an Uncertain Robot with Output Constraints," **In: International Conference on Intelligent Control and Automation Science**, vol. 46 (2013) pp. 51–56.
- [24] W. He, Y. H. Chen and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Trans. Cybern.* **46**(3), 620–629 (2016).
- [25] H. Yu, T. T. Xie, S. Paszczynski and B. M. Wilamowski, "Advantages of radial basis function networks for dynamic system design," *IEEE Trans. Ind. Electron.* **58**(12), 5438–5450 (2011).
- [26] W. He and Y. T. Dong, "Adaptive fuzzy neural network control for a constrained robot using impedance learning," *IEEE Trans. Neur. Netw. Learn.* **29**(4), 1174–1186 (2018).
- [27] Q. J. Yao, "Neural adaptive learning synchronization of second-order uncertain chaotic systems with prescribed performance guarantees," *Chaos Solitons & Fractals* **152** (2021).
- [28] Y. J. Liu, S. M. Lu, S. C. Tong, X. K. Chen, C. L. P. Chen and D. J. Li, "Adaptive control-based barrier Lyapunov functions for a class of stochastic nonlinear systems with full state constraints," *Automatica* **83**, 83–93 (2018).

- [29] Z. J. Li, J. Q. Liu, Z. C. Huang, Y. Peng, H. Y. Pu and L. Ding, “Adaptive impedance control of human-robot cooperation using reinforcement learning,” *IEEE Trans. Ind. Electron.* **64**(10), 8013–8022 (2017).
- [30] H. Modares, I. Ranaunga, F. L. Lewis and D. O. Popa, “Optimized assistive human-robot interaction using reinforcement learning,” *IEEE Trans. Cybern.* **46**(3), 655–667 (2016).
- [31] F. L. Lewis and K. G. Vamvoudakis, “Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data,” *IEEE Trans. Syst. Man Cybern. B Cybern.* **41**(1), 14–25 (2011).
- [32] S. Zhang, Y. T. Dong, Y. C. Ouyang, Z. Yin and K. X. Peng, “Adaptive neural control for robotic manipulator with output constraints and uncertainties,” *IEEE Trans. Netw. Learn. Syst.* **29**(11), 5554–5564 (2018).
- [33] H. H. Zheng and Z. D. Fang, “Research on tracking error of robot arm based on adaptive neural network control,” *Mach. Des. Manufact.* **6**, 139–141 (2019).
- [34] Y. X. Wu, R. Huang, Y. Wang and J. Q. Wang, “Adaptive tracking control of robot manipulators with input saturation and time-varying output constraints,” *Asian J. Control* **23**(3), 1476–1489 (2021).