International Journal of Intelligent Robotics and Applications (2020) 4:229–242
https://doi.org/10.1007/s41315-020-00132-5

REGULAR PAPER

Iterative learning of human partner's desired trajectory for proactive human–robot collaboration

Jingkang Xia[1] · Deqing Huang[1] · Yanan Li[2] · Na Qin[1]

Received: 10 October 2019 / Accepted: 23 April 2020 / Published online: 5 June 2020

Abstract
A period-varying iterative learning control scheme is proposed for a robotic manipulator to learn a target trajectory that is planned by a human partner but unknown to the robot, which is a typical scenario in many applications. The proposed method updates the robot's reference trajectory in an iterative manner to minimize the interaction force applied by the human. Although a repetitive human–robot collaboration task is considered, the task period is subject to uncertainty introduced by the human. To address this issue, a novel learning mechanism is proposed to achieve the control objective. Theoretical analysis is performed to prove the performance of the learning algorithm and robot controller. Selective simulations and experiments on a robotic arm are carried out to show the effectiveness of the proposed method in human–robot collaboration.

Keywords Period-varying iterative learning control · Human–robot interaction · Trajectory learning · Interaction force tracking

1 Introduction

Human–robot interaction (HRI) is a rapidly expanding research field with a great need for robotic assistance in various applications such as assembly, teleoperated surgery, heavy load transport and so on [1–6]. HRI has been mainly studied in two complementary branches, such as divisible shared control (DSC) that has been discussed in [7–9] and interactive shared control (ISC) in [10–12]. DSC enables clear division of the human partner's and the robot's subtasks so that they can work independently, while ISC allows for flexible intervention to improve the collaborative performance which is the focus of this paper. How to design a simple, effective and safe robotic controller has been a focus in the field of HRI [13], especially in the subfield of physical human–robot interaction (pHRI).

A critical yet typical problem in pHRI is how to obtain the human's desired trajectory or motion intention, as the knowledge of this information enables a robot to adapt its behaviour according to the human partner and to complete a task cooperatively [14, 15]. In the literature of pHRI, researchers have tried to extract this information from the human partner by utilizing the haptic information, especially the interaction force that can be measured from the force/torque sensor at the interaction point. In [10], neural networks are used to build up a mapping between the human's desired trajectory and the interaction force. In [11], the robot's desired trajectory is deformed according to the interaction force, while the human's desired trajectory is not explicitly estimated. A different approach is adopted in [12], which develops a controller for lifting a beam with human–robot collaboration, using a probabilistic framework based on Gaussian Mixture Models (GMM) and Gaussian Mixture Regression (GMR) to encode force/torque at the end-effector with the demonstrated velocity. After learning the robot could perform the collaborative lifting by conditioning the GMM on the perceived force/torque and generate the velocity response using GMR. The strategy of using interaction force as stimuli is also applied in [16] for teaching multiple solution trajectories by means of a Hidden Markov Model (HMM) and GMR. The perceived force/torque could be displayed to the human partner by using a haptic device, and the human partner demonstrates motion response. In addition, without using a force sensor, [17] proposes a human intent detection

✉ Yanan Li
yl557@sussex.ac.uk

[1] Southwest Jiaotong University, Chengdu 610031, People's Republic of China

[2] University of Sussex, Brighton BN1 9RH, UK

method by observing the impact of the physical interaction on the robot's control effort. From the control point of view, these works use the estimated human information to alter the robot's controller, but they do not evaluate how it affects the performance of the closed-loop system. This is important as human is involved in the system so the system stability should be ensured at all instances.

Impedance control provides a useful framework for analyzing a pHRI system [18], which is adopted as a basis of many works. In [19–21], adaptive control is developed to deal with uncertainty of the parameters in robot dynamics to achieve a target impedance model. In [22], a desired impedance model is learned for a robot through repetitive interaction with its environment. It is important to note that impedance control is useful for not only regulating the interaction between the robot and the environment, but also for establishing a human motor control model. Although describing an individual human's behavior in physical interaction with an environment is challenging, impedance control is found to be an effective and robust computational model [23, 24]. With a unified model of impedance control for robot control and human motor control, it is natural to use it for skill transfer from the human to the robot. In [25], by measuring the sensory outputs from the robot and the corresponding motor commands generated by the human tutor, a proper mapping between the selected muscle EMG and robot's end-effector impedance is established. Moreover, the coupled HRI system can be explicitly described within the same framework of impedance control, so it becomes possible to evaluate the system performance with estimation of the human state and adaptation of the robot's controller. In the literature, a similar idea is found for impedance-based force control [20], where the robot is under impedance control and its environment is treated as a spring model. In this way, the robot-environment system can be analyzed within the same control framework. Differently, in pHRI, the human dynamics cannot be described as a passive system by a spring model. In this paper, a human model with an actively planned trajectory is investigated.

On the other hand, it is found that many pHRI processes are repetitive, so it is a natural idea to introduce iterative learning control (ILC) for the robot to gain skills from the human partner. In the past decades, ILC has been developed for improving performance of systems that carry out repetitive operations [26–28]. Its main goal is to generate a control that tracks a given reference or rejects a periodic disturbance [29]. However, in the pHRI problem under study in this paper, the human's desired trajectory is unknown to the robot so the control objective is different from the traditional ILC. It will be shown that the control problem is formulated in a way that is different from conventional trajectory tracking, although ILC is employed.

Furthermore, due to human variance during repetitive pHRI, the same period cannot be guaranteed for every iteration. Therefore, this paper develops a period-varying ILC that ensures learning convergence in the presence of human uncertainty. In the literature of ILC, some recent works [30–35] investigate the problem of period-varying ILC which lay the foundation of our controller. In particular, [33] proposes an ILC scheme with an iteration-average operator for discrete-time linear systems where the iteration period could be randomly varying. [34] introduces an ILC scheme based on an iteratively moving average operator for nonlinear dynamic systems with randomly varying iterative periods. Different from the above two methods, [35] applies the adaptive design framework for continuous-time parametric nonlinear systems with varying iteration periods. Similar to other ILC methods as discussed above, these three methods also require a predefined desired output trajectory. In this paper, the human's unknown desired trajectory will be first learned by using a human limb model and by formulating a zero force regulation problem. Then, it will be tracked by the robot so it can reduce human's control effort.

Through the above discussion, this paper has the following contributions:

1. Based on a simplified human limb model, estimation of the human partner's desired trajectory is formulated as a force regulation problem.
2. Different from traditional ILC which has been mainly used for trajectory tracking, we develop ILC to achieve force regulation and estimation of human trajectory, enabling the robot to proactively cooperate with the human through repetitive pHRI.
3. A period-varying ILC is developed to deal with the problem of interactions with uncertain time durations due to human variance, so it can be applied to a wider range of pHRI applications.

The rest of the paper is arranged as follows: Sect. 2 introduces the main problem formulation, including the system description and the novel control objective. In Sect. 3, a non-standard ILC is designed to solve the problem of varying iterative periods and theoretical analysis is conducted to show the learning convergence and system stability. Furthermore, the proposed algorithm is testified by simulations and experiments in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Problem formulation

2.1 System description

In this paper, we consider a typical human–robot collaboration scenario that is composed of a robotic manipulator and its human partner. The human partner guides the robotic manipulator along a trajectory to complete a task, e.g. painting a surface with a certain curvature. This trajectory is determined by the human but cannot be preprogrammed to the robotic manipulator. The interaction force between the human hand and the robotic manipulator is measured by a force sensor at the end-effector of the robotic manipulator.

The dynamics model of the n-degrees-of-freedom (n-DOF) robotic manipulator is given as below:

$$H(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = u - J^T F_h \tag{1}$$

where $\theta, \dot{\theta}, \ddot{\theta} \in \mathbb{R}^n$ represent the joint position, velocity and acceleration vectors, respectively; $H(\theta) \in \mathbb{R}^{n\times n}$ is the symmetric positive definite mass matrix; $C(\theta, \dot{\theta})\dot{\theta} \in \mathbb{R}^n$, $G(\theta) \in \mathbb{R}^n$ denote the torques due to centrifugal and gravity; $u \in \mathbb{R}^n$ is the joint torque applied by robotic actuators; $J \in \mathbb{R}^{n\times n}$ is the Jacobian matrix that relates the joint velocity to the linear and angular velocities of the end-effector; and $F_h \in \mathbb{R}^n$ is the robot/human interaction force that can be measured by a force sensor.

It is often desirable to describe the manipulator dynamics in the Cartesian space for the convenience of analysis, when the interaction takes place at the end-effector. The robot dynamics in the Cartesian space are given by

$$H_x\ddot{X} + C_x\dot{X} + G_x = J^{-T}u - F_h \tag{2}$$

where $X \in \mathbb{R}^n$ represents the position of the end effector. The velocity and acceleration satisfy

$$\begin{aligned} \dot{X} &= J\dot{\theta} \\ \ddot{X} &= J\ddot{\theta} + \dot{J}\dot{\theta} \end{aligned} \tag{3}$$

Then, we obtain

$$\begin{aligned} H_x &= J^{-T}HJ^{-1} \\ C_x &= J^{-T}(C - HJ^{-1}\dot{J})J^{-1} \\ G_x &= J^{-T}G \end{aligned} \tag{4}$$

According to [20], dynamics of a robot's environment can be described by a spring model. This corresponds to the equilibrium point control model that describes human motor control [36], which is given below:

$$F_h = K_h(X - X_h), \tag{5}$$

where K_h is the $n \times n$ equivalent stiffness matrix and X_h is the desired trajectory of the human. With Eqs. (2) and (5), we have a full description of the system dynamics and are ready to discuss the control objective.

2.2 Control objective

In traditional control tasks, the reference trajectory of the robot is available for the control design. However, in this paper, it should be designed according to the desired trajectory that is determined by the human partner and unknown to the robot. In the rest of the paper, we propose an approach to estimate the human partner's desired trajectory and control the robot to track it.

For this purpose, impedance control is adopted with the following target impedance model:

$$M_d(\ddot{X} - \ddot{X}_r) + C_d(\dot{X} - \dot{X}_r) + K_d(X - X_r) = -F_h \tag{6}$$

where X_r is the robot's reference trajectory. M_d, C_d and G_d are inertia, damping and stiffness matrices that are specified by the control designer, respectively.

To make Eq. (2) identical to Eq. (6), the robot's joint torque is given by $u = J^T F$ with

$$\begin{aligned} F = {} & C_x\dot{X} + G_x + F_h - H_x M_d^{-1}\{C_d(\dot{X} - \dot{X}_r) \\ & + K_d(X - X_r) + F_h + \ddot{X}_r\} \end{aligned} \tag{7}$$

To estimate the human partner's desired trajectory, we observe from Eq. (5) that the interaction force F_h becomes 0 when the human's desired trajectory is perfectly tracked, i.e. $X = X_h$. This indicates that we can minimize F_h so that the robot's trajectory X gets close to X_h as much as possible. Based on this idea, the control objective becomes

$$F_h = 0 \tag{8}$$

The next question is how to design the reference trajectory X_r to achieve the above control objective. As discussed in the Introduction, this reference trajectory can be learned through repetitive interaction between the robot and the human. In particular, we will adopt ILC to iteratively update X_r so that the control objective $F_h = 0$ is eventually achieved.

The control flow is illustrated in Fig. 1, where i represents the number of iterations. The "Impedance Control" part shows that the robot dynamics (2) are controlled by the robot controller (7), which has inputs X_{r_i} and F_{h_i} measured by a force sensor in the "Measurement and Storage" part. In the "Update Reference Position" part, (5) is an internal human model that relates the position information and interaction force F_{h_i}, which is also used to derive the dynamics model (12). Then, the developed ILC (26) uses the dynamics model (12) to update the reference trajectory $X_{r_{i+1}}$ and store it for the use in the next iteration, which will be detailed in the following section (Fig. 1).

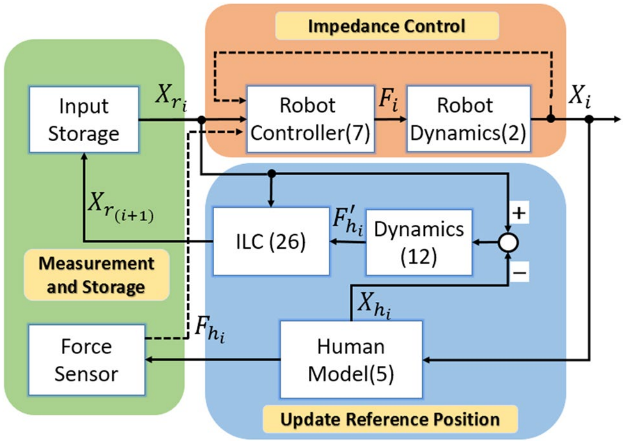


Fig. 1 Control flow

3 Controller design

3.1 System model transformation

For clarity of presentation, in the subsequent analysis we consider each Cartesian variable independently and replace M_d, C_d, K_d, E, F_h, F_r by m_d, c_d, k_d, e, f_h, f_r, respectively. In this way, the target impedance model in a single direction becomes

$$m_d(\ddot{x} - \ddot{x}_r) + c_d(\dot{x} - \dot{x}_r) + k_d(x - x_r) = -f_h \tag{9}$$

By using the Laplace transform of (9), we have

$$(m_d s^2 + c_d s + k_d)(x(s) - x_r(s)) = -f_h(s) \tag{10}$$

Using the environment model (5), $x(s)$ can be described by $x(s) = \frac{f_h(s)}{k_h} + x_h(s)$. Substituting for $x(s)$ in (10) yields

$$f_h(s) + \frac{k_h f_h(s)}{m_d s^2 + c_d s + k_d} = k_h(x_r(s) - x_h(s)) \tag{11}$$

The above equation combines the robot's impedance model and the human motor control model. By defining $f'_h(s) = f_h(s)/(m_d s^2 + c_d s + k_d)$, we have

$$m_d s^2 f'_h(s) + c_d s f'_h(s) + (k_d + k_h) f'_h(s) = k_h(x_r(s) - x_h(s)) \tag{12}$$

f'_h can be simply interpreted as a filtered signal of f_h. Then, we convert the system in Eq. (12) from the frequency domain back to the time domain, and obtain the time-domain state-space form as follows:

$$\begin{aligned} &\dot{\varphi}(t) = A\varphi(t) + Be(t), \quad \varphi(t) = \begin{bmatrix} f'_h(t) \\ \dot{f}'_h(t) \end{bmatrix}, \\ &A = \begin{bmatrix} 0 & 1 \\ \frac{-(k_d+k_h)}{m_d} & \frac{-c_d}{m_d} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{k_h}{m_d} \end{bmatrix}, \end{aligned} \tag{13}$$

where $e(t) = x_r(t) - x_h(t)$. In order to apply ILC, this state-space form is further written for each iteration as below:

$$\begin{cases} \dot{\varphi}_i(t) = A\varphi_i(t) + Be_i(t) \\ \rho_i(t) = C\varphi_i(t) \end{cases} \tag{14}$$

where $t \in [0, T_i]$ and $i \in [1, \mathrm{N}]$ is the iteration index. N is the total number of iterations and T_i is the trial length of the ith iteration. $C = [1, 1]$ and $e_i(t)$, $\varphi_i(t)$, $\rho_i(t)$ denote input, state and output of the system (14), respectively. Note that T_i may be different in a different iteration, as the human introduces uncertainty to the interaction. Therefore, traditional ILC with a fixed period cannot apply to this case. In the following subsection, we will develop an ILC with varying periods.

Moreover, it is important to emphasize that the control objective is to make the interaction force F_h be zero, indicating that the human partner's desired trajectory will be tracked. This is denoted as $\lim_{i\to\infty} f_{h_i}(t) = 0$ and $\lim_{i\to\infty} e_i(t) = 0$ in the framework of ILC. Thus, for the system (12), $e_d(t) = 0$ and $\rho_d(t) = 0$ are the desired input and output, respectively.

3.2 ILC design

The main problem about the ILC design for the system in Eq. (14) is that the period T_i is uncertain and usually different from a fixed period T_d. In order to address it, we first give some definitions and assumptions that will be used in the later design.

Definition 1 $\mathbf{P}[\cdot]$, $\mathbf{E}[\cdot]$ and $\mathbf{F}[\cdot]$ denote the occurrence probability of the event, the expectation of a random variable and the probability distribution function.

Assumption 1 *Assume that T_i is a stochastic variable, and its probability distribution function is*

$$\mathbf{F}_{T_i}(t) = \mathbf{P}[T_i \le t] = \begin{cases} 0, & t \in [0, T_{min}] \\ p(t), & t \in [T_{min}, T_{max}] \\ 1, & t > T_{max} \end{cases} \tag{15}$$

where $0 \le p(t) \le 1$ is a known continuous function. T_{min} and T_{max} are the minimum and maximum of T_i that can be obtained through multiple iterative experiments.

Assumption 2 $\varphi_i(0) = \varphi_d(0)$, *which indicates that the initial state is the same at every iteration.*

If $T_i = T_d$ and initial conditions are consistent, an open loop D-type ILC for the system (14) is usually designed as

$$e_{i+1}(t) = e_i(t) + \omega\dot{\mu}_i(t) \tag{16}$$

where $\mu_i(t) = \rho_i(t) - \rho_d(t) = f'_{h_i}(t) + \dot{f}'_{h_i}(t)$ is the tracking error on the interval $[0, T_d]$, $\dot{\mu}_i(t)$ is the derivative of $\mu_i(t)$ on

$[0, T_d]$, and ω is the learning rate. Considering that $x_h(t)$ does not change with the iteration, i.e. $x_{h_i}(t) = x_{h_{i+1}}(t)$, Eq. (16) can be rewritten as

$$x_{r_{i+1}}(t) = x_{r_i}(t) + \omega \dot{\mu}_i(t) \tag{17}$$

which gives an updating law for x_r. When the actual period $T_i \neq T_d$, which is the case of the non-standard ILC, the updating law (17) has to be redesigned.

Since T_i changes with iteration, two cases need to be discussed: $T_i < T_d$ and $T_i \geq T_d$. On the one hand, $T_i < T_d$ means that the ith iteration ends before the desired period, so both the output $\rho_i(t)$ and $\mu_i(t)$ on the time interval $[T_i, T_d]$ are missing, which thus cannot be used for learning. On the other hand, $T_i \geq T_d$ means that the ith iteration is still running after the desired period, the data from $(T_i, T_d]$ are redundant and useless. In order to deal with those missing data or redundant data in different cases, a sequence of stochastic variables is defined to satisfy Binomial distribution, so that a newly defined force error $\mu_i(t)$ is introduced to improve the traditional ILC.

The main improvement process has the following four steps:

1. Define the stochastic variable $\gamma_i(t)$ to satisfy the Binomial distribution.

   In this paper, we define $\gamma_i(t), t \in [0, T_{max}]$ as a stochastic variable satisfying Binomial distribution and taking binary values 1 and 0. $\gamma_i(t) = 1$ indicates that the control process of (14) can arrive at time t in the ith iteration. The probability of $\gamma_i(t) = 1$ is $q(t)$, where $0 < q(t) \leq 1$ is a distribution probability function of time t. $\gamma_i(t) = 0$ indicates that the control process of (14) cannot arrive at time t, which occurs with a distribution probability of $1 - q(t)$.
2. Compute the probability $\mathbf{P}[\gamma_i(t) = 1]$ and the expectation $\mathbf{E}[\gamma_i(t)]$.

   Because of $T_{min} \leq T_i \leq T_{max}$ the control process (14) will not stop before T_{min}. $\gamma_i(t) = 1$ is the inevitable event when $t \in [0, T_{min}]$, which implies that $q(t) = 1$, $\forall t \in [0, T_{min})$. For the scenario of $t \in [T_{min}, T_{max}]$, the event $\gamma_i(t) = 1$ represents that the control process (14) stops at or after time t, which means that $T_i \geq t$. $\mathbf{P}[T_i = t] = 0$ is applied, thus

$$\begin{aligned} \mathbf{P}[\gamma_i(t) = 1] &= \mathbf{P}[T_i \geq t] \\ &= 1 - \mathbf{P}[T_i \leq t] \\ &= 1 - F_{T_i}(t) \end{aligned} \tag{18}$$

   Combining Eqs. (15) and (18), we can obtain that

$$q(t) = 1 - F_{T_i} = \begin{cases} 1, & t \in [0, T_{min}] \\ 1 - p(t), & t \in [T_{min}, T_{max}] \\ 0, & t > T_{max} \end{cases} \tag{19}$$

   Because $\gamma_i(t)$ satisfies the Binomial distribution and taking binary values 1 and 0, the expectation $\mathbf{E}[\gamma_i(t)] = q(t) \cdot 1 + (1 - q(t)) \cdot 0 = q(t)$.
3. Define a modified force error.

   Denote the modified tracking error as

$$\mu_i^*(t) = \gamma_i(t)\mu_i(t), \quad t \in [0, T_d] \tag{20}$$

   Taking derivative on both sides of the equation leads to

$$\dot{\mu}_i^*(t) = \gamma_i(t)\dot{\mu}_i(t), \quad t \in [0, T_d] \tag{21}$$

   If $T_i < T_d$, Eq. (21) can be rewritten to

$$\dot{\mu}_i^*(t) = \begin{cases} \dot{\mu}_i(t), & t \in [0, T_i] \\ 0, & t \in (T_i, T_d] \end{cases} \tag{22}$$

   If $T_i \geq T_d$, Eq. (21) can be rewritten to

$$\dot{\mu}_i^*(t) = \dot{\mu}_i(t), \quad t \in [0, T_d] \tag{23}$$

4. Design a new ILC updating law.

   In order to compensate for the missing data and reduce the error of single learning, an iteratively moving average operator is introduced, as below

$$\Phi\{e_i(t)\} = \frac{1}{m+1}\sum_{j=0}^{m} e_{i-j}(t), \tag{24}$$

   for a sequence $e_{i-m}(t),\ e_{i-m+1}(t),\ e_{i-m+2}(t)\ldots,\ e_i(t)$ with $m \geq 1$, which includes only the recent $m + 1$ iterations that could provide more accurate control information for learning process. The ILC updating law is given as follows:

$$e_{i+1}(t) = \Phi\{e_i(t)\} + \sum_{j=0}^{m} \beta_j \dot{\mu}_{i-j}^*(t), \quad t \in [0, T_d] \tag{25}$$

   where the learning rates $\beta_j \in \mathbf{R}$, $j = 0, 1, 2, 3\ldots, m$ and $e_i(t) = 0, i < 0$. Considering Eq. (17), we develop the following updating law for x_r:

$$x_{r_{i+1}}(t) = \Phi\{x_{r_i}(t)\} + \sum_{j=0}^{m} \beta_j \dot{\mu}_{i-j}^*(t), \quad t \in [0, T_d] \tag{26}$$

With the above design, the main result of this paper is presented in the following theorem and its proof is in the Appendix.

Theorem 1 *For the system* (14) *and the ILC scheme* (26), *we choose the appropriate learning rates* β_j, $j = 0, 1, 2, 3 \ldots, m$ *so that for any constant* $0 \leq \epsilon < 1$,

$$\sum_{j=0}^{m} \alpha_j \leq \epsilon \tag{27}$$

where $\alpha_j = \sup_{t\in[0,T_d]} \left\{ |\frac{1}{m+1} - \beta_j CB| q(t) + \frac{1-q(t)}{m+1} \right\}$, *then the tracking error* $\mu_i(t)$, $t \in [0, T_d]$ *will converge to zero iteratively.*

Remark 1 While the probability distribution of T_i is unknown, it could be estimated based on preliminary experiments. Thus, the probability distribution function $F_{T_i}(t)$ in Assumption 1 is known and $q(t)$ is available for controller design that can be calculated by Eq. (19).

4 Simulations

4.1 Simulation I

In this section, we consider a single joint application scenario with an exoskeleton robot for gait tracking through iterative learning. Related works have been done for lower-limb exoskeleton for gait tracking and rehabilitation, such as [37, 38]. As shown in Fig. 2a, in the simulation we suppose that the interaction force could be measured by the force sensor mounted on the exoskeleton.

The desired joint trajectory of the user's gait is set as $q_h(t) = \frac{\pi t}{20} - \frac{\pi}{2}$, $t \in [0, 10]$ s and $T_d = 10$ s, indicating that the rotational joint is moving at a constant speed. The parameters of the desired impedance model are set as: $m_d = 0.1$ kg, $c_d = 25$ N/(m/s) and $k_d = 20$ N/m. The human leg's

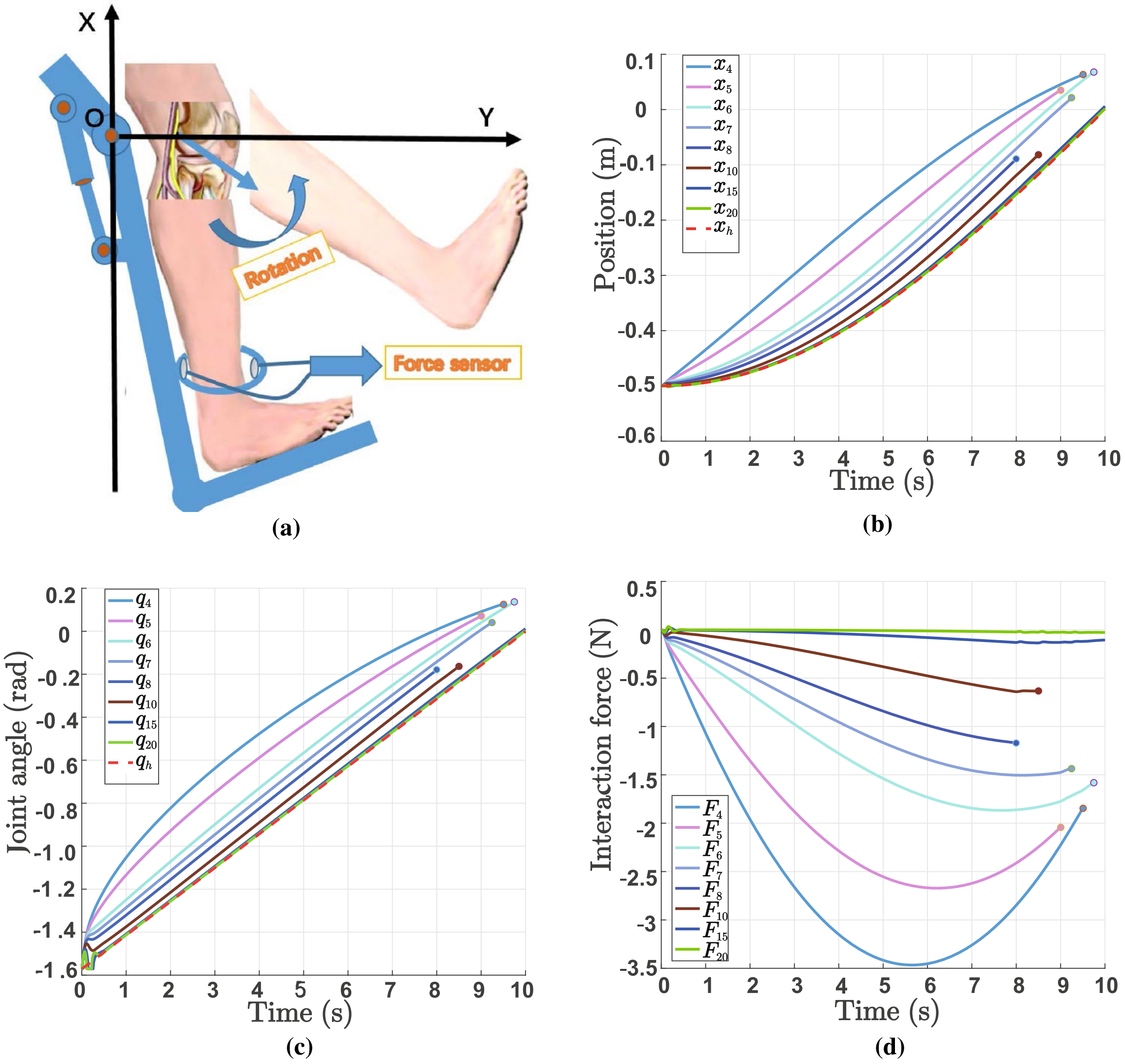


Fig. 2 **a** Simulated gait tracking scenario. **b** The process of learning a desired trajectory. **c** Change of the joint angles. **d** Interaction forces

stiffness is set as $k_h = 18$ N/m. In the Cartesian space, we take into account the change of the endpoint position in the X direction. The link length is 0.5 m, so according to the mapping between joint angles and endpoint position, we can obtain that the desired endpoint position with time is $x_h = -0.5cos(\frac{\pi t}{20})$.

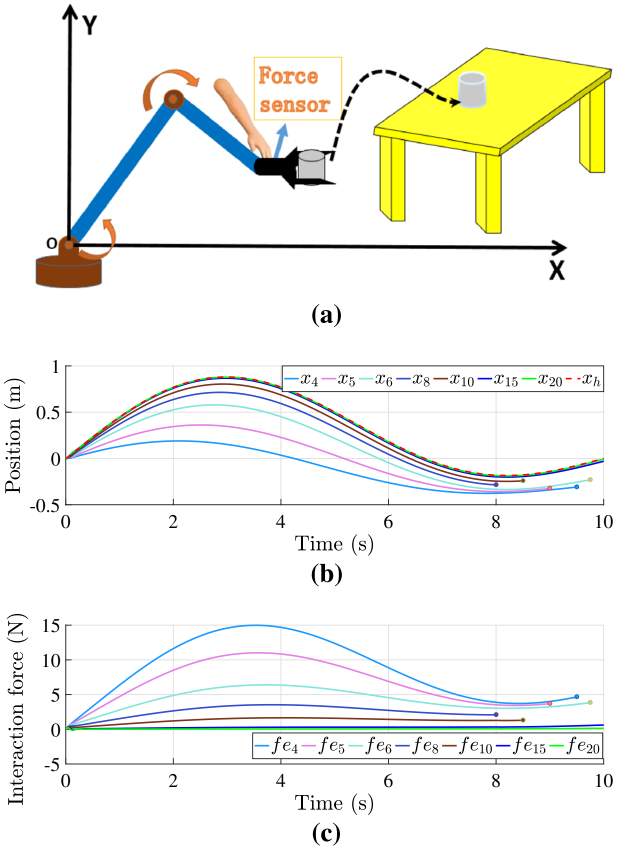
The reference trajectory before learning is $x_{r_0}(t) = (-0.5, 0)$ m, $t \in [0, 10]$ and the initial position is $(-0.5, 0)$ m for all iteration trails. The learning period of each iteration will change because of the user's influence, so the periods of the 4th, 5th, 6th, 7th, 8th and 10th are different from T_d. And we assume that the period obeys a normal distribution on the time range [6, 10] s with a mean of 8 and a standard deviation of 0.2. As is shown in Fig. 2b–d, the end of each iteration is marked with a dot. In order to obtain better simulation results, we take the first three results as the iteratively moving average operator's elements, so m is set as 3 and $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0.25$.

The tracking performance of the trajectory and joint angle are shown in Fig. 2b, c. It can be seen from the two figures that the actual position and joint angle trajectory (the solid lines in different colors) track the desired trajectory (the curve shown by the dotted red line) rapidly and precisely. In addition, the interaction force converges to zero after 20 learning iterations which can be observed from Fig. 2d. These results show that the whole learning process is also a skill transfer process. When the robot learns the user's gait, it becomes easier for the user to cooperate with the exoskeleton robot.

4.2 Simulation II

Different from simulation I, in this section, a 2-DOF manipulator is used to verify the proposed period-varying ILC algorithm. A typical trajectory learning process is considered in this simulation which can emulate an application of object manipulation, as is shown in Fig. 3a. The robot moves in the $X - Y$ plane with an initial position (0, 0)m. With the guidance of a human partner, the robot gradually learns to track the desired trajectory that is determined by the human partner. It is assumed that the interaction force is only exerted in the Y direction and the robot's speed in the X direction is 0.1 m/s. The desired trajectory in the Cartesian space is $x_h = \frac{sin0.2\pi t}{2} + \frac{sin0.1\pi t}{2} m, t \in [0, 10]$ and thus $T_d = 10$ s is the desired period. The parameters of the desired impedance model are set as: $m_d = 0.5$, $c_d = 28N/_{(m/s)}$ and $k_d = 26$ N/m. The human arm's stiffness is set as $k_h = 20$.

Without loss of generality, the input of the initial iteration is $x_{r_0}(t) = [0, 0], t \in [0, 10]$ s. In this simulation, we assume that the iteration period obeys the Gaussian distribution with a mean of 10 and a standard deviation of 0.2. We further set $m = 3$ and $\beta_0 = \beta_1 = \beta_2 = \beta_3 = 0.2$. The algorithm runs for 20 iterations.



(a)

(b)

(c)

Fig. 3 **a** Human–robot collaboration scenario for carrying a cup. **b** The evolution of the trajectory learning. **c** Interaction force

The tracking performance of the trajectory is shown in Fig. 3b, where the lengths of the 4th, 5th, 6th, 8th and 10th periods are different from the desired one. It is obvious that the robot can gradually reach the human's desired trajectory x_h. The interaction forces at the 4th, 5th, 6th, 8th, 10th, 15th and 20th are shown in Fig. 3c. It is observed that the interaction force is reduced to zero in the 20th iteration, so the robot fully learns the desired trajectory and it can eventually complete the task alone.

In addition, we change the value of k_h to represent the stiffness of different human users, varying from 10 to 28 N/m, i.e. $k_{h_{user1}} = 10$ N/m, $k_{h_{user2}} = 12$ N/m … $k_{h_{user10}} = 28$ N/m. According to the above task and parameters, simulations are designed to show the tracking performance of ten different users. Furthermore, we introduce traditional impedance control with zero stiffness for comparison. In order to facilitate comparison of the results, the values of inertia and damping parameters are the same as in the proposed ILC. For the proposed ILC, we record the different users' total iteration numbers, maximum errors and maximum interaction forces after convergence. For the impedance control, the maximum errors and maximum interaction forces are recorded.

The tracking performance and interaction force of impedance control with different users are shown in Fig. 4a, b where different colors represent different users, while it can be found that different users' tracking trajectories almost overlap. With the proposed method, the number of iterations is different for each user, which decreases as k_h increases in an appropriate range as shown in Fig. 5a. From Fig. 5b, c, it can be found that the proposed ILC achieves smaller maximum error and interaction force compared with impedance control for different users. Thus, the proposed method improves the tracking accuracy and dramatically reduces the interaction force so less human effort is required.

Simulation I and II demonstrate that the proposed method for tracking repeated trajectories is feasible. However, parameters of the desired impedance model and learning rates of the ILC are different, as the choice of parameters will affect the accuracy of tracking and the speed of convergence. Based on the desired impedance parameters in reference [22], we adjust them appropriately and find that they should not be chosen too large, otherwise the system will become unstable. For iterative learning rates, they must satisfy the condition in Theorem 1. It is worth noting that large learning rates β_i lead to faster convergence but larger tracking errors, whereas small learning rates will make the iteration number increase. The influence of the impedance parameters and the iterative learning rates have been extensively studied in the literature [22, 33, 34], respectively, so this paper does not discuss it further.

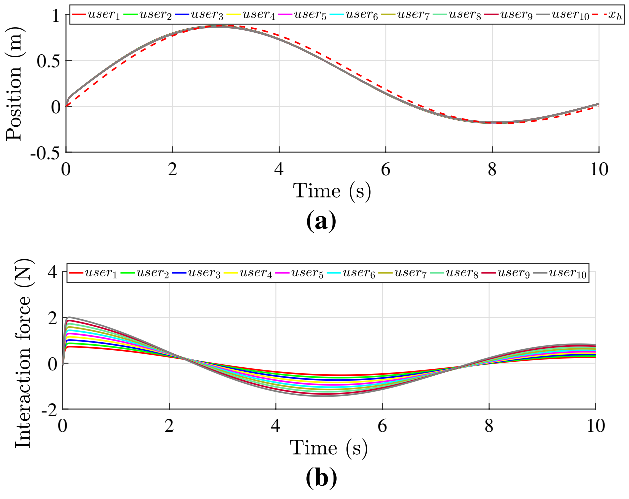


Fig. 4 **a** Ten users' trajectories by impedance control. **b** Ten users' interaction forces by impedance control

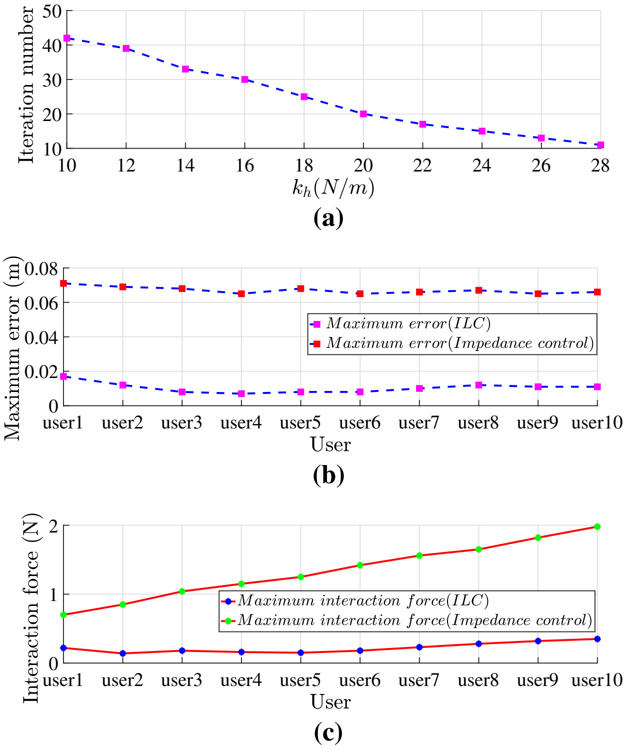


Fig. 5 **a** Iteration number with k_h varying from 10 to 28. **b** Maximum tracking error for different users. **c** Maximum interaction force for different users

5 Experiment verification

In this section, to verify the validity of the proposed algorithm we carry out an experiment on the Sawyer robot, which is a 7-DOF manipulator equipped with a motor encoder and a torque sensor in each joint. As is shown in Fig. 6, a human user guides the robot manipulator to track a circular trajectory according to his intent but this trajectory is unknown to the robot. This experiment setup emulates an industrial application where the robot learns a motion from the human partner through repetitive collaboration. In the experiment, joints 1 and 2 are used which make the experimental platform similar to that in the simulation in Fig. 3a.

The experiment process is shown in Fig. 6, with a circular trajectory that starts from $A(0.26, 0)$ m and has a diameter of 0.4 m. We set the desired period as $T_d = 20$ s that is estimated by preliminary experiments. In the X-axis direction, we set

$$x_r = \begin{cases} \frac{0.8}{T_d}t + 0.26, & 0 \le t < \frac{T_d}{2} \\ 0.66 - \frac{0.8}{T_d}t, & \frac{T_d}{2} \le t \le T_i \end{cases} \tag{28}$$

where T_i is the period of each iteration determined by the human user. In the Y-axis direction, there is a vertical

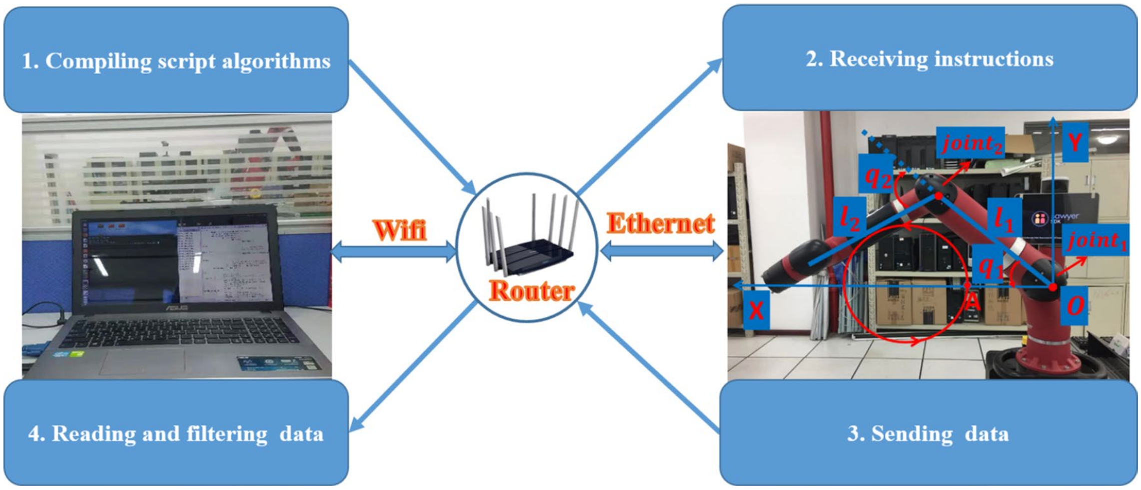

1. Compiling script algorithms
2. Receiving instructions
Wifi
Ethernet
Router
joint$_2$
Y
q_2
l_2
l_1
joint$_1$
q_1
X
A
O
4. Reading and filtering data
3. Sending data

Fig. 6 This figure shows four steps of the experiment: 1. Compiling script algorithms under ROS in Ubuntu system to control robotic manipulator and read joint information. 2. Taking script instructions and driving the motor under the Sawyer SDK mode. 3. The data of each joint's angle and torque are collected and sent to the computer. 4. Receiving the data of each joint and calling MATLAB to filter and fit the data. The robotic manipulator is connected with the computer through the router to transmit data and instructions. In the figure on the right, we simplify the Sawyer robot into a 2-DOF platform and establish a planar coordinate system

interaction force on the robot's end-effecter. One key issue of the experiment is to measure the interaction force, since there is no force sensor that can directly measure the force at the end-effector. Therefore, we convert the joint torque that can be measured by the torque sensor at each joint to obtain the approximate interaction force. The parameters used in the experiment are summarized in Table 1. Because the physical meaning of each parameter is introduced above, we do not repeat them here.

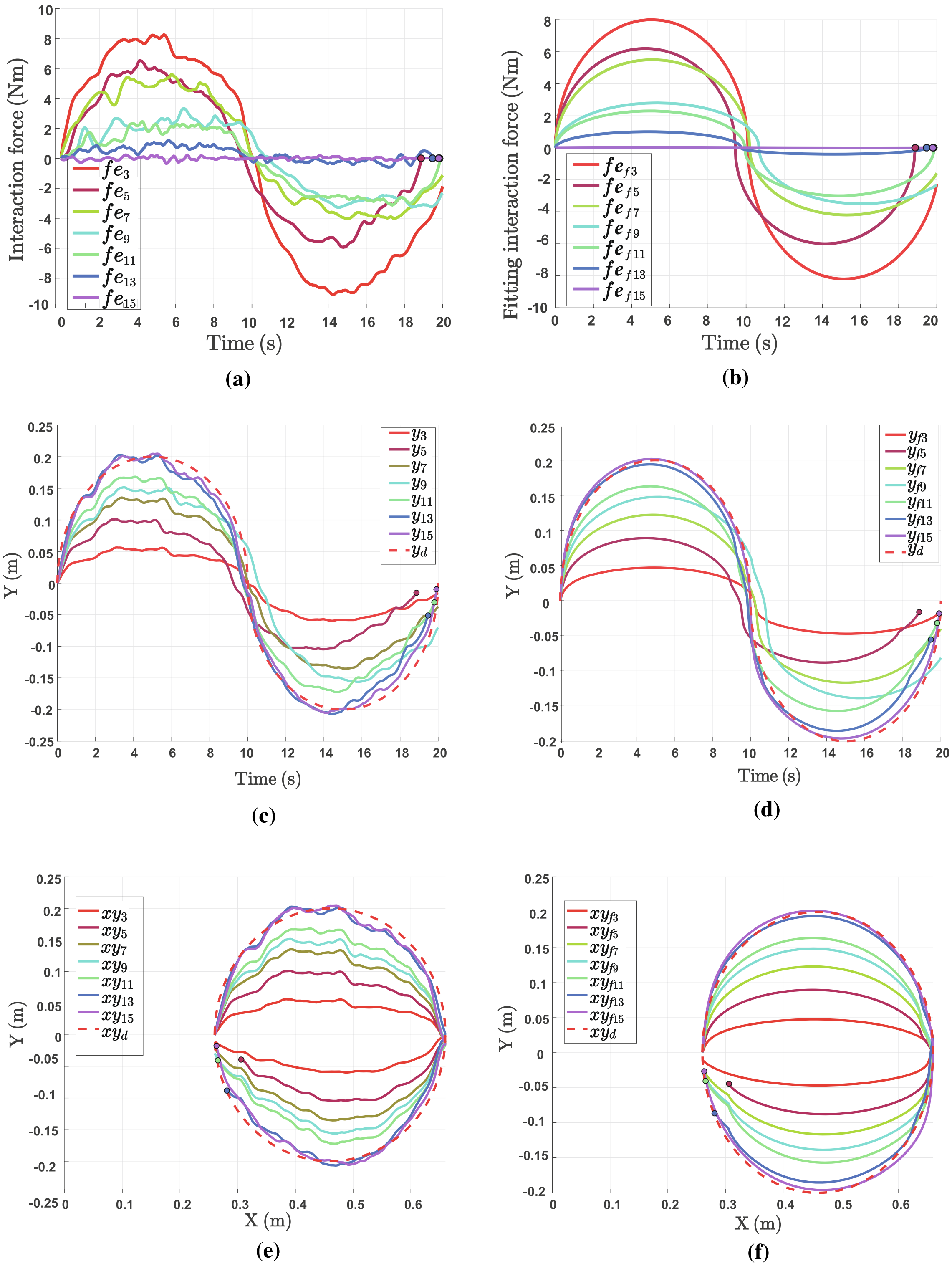
During the process of the experiment, due to continuous change of human–robot interaction force and the interference of friction in each joint, the learning period T_i changes accordingly. We record the learning periods of typical iterations in Table 2. As is shown in Fig. 7a, we present the change of the interaction force over the iterations. It is worth noting that due to the force measurement noise, the measured data must be smoothed through a filter. And then we take the interaction force into the designed ILC algorithm to constantly update the robot's reference trajectory. In order to control the motors, we convert the end-effector positions to joint angles through inverse kinematics. To further illustrate the effectiveness of the proposed algorithm, we also fit the filtered force data to get better tracking performance as is shown in Fig. 7b. Finally, the position change of the end-effector in the Y-direction and the trajectory in the vertical plane are given in Fig. 7d, f.

From Fig. 7c, e, it can be seen that the robotic manipulator accomplishes the trajectory tracking task under the effect of the actual interaction force, and the maximum error in the 15th iteration is less than 1 cm. In addition, we get a better tracking performance through the fitted interaction force as in Fig. 7d, f. Correspondingly, the interaction force gradually decreases to 0 and the trajectory gradually converges. These results clearly demonstrate that the robot is able to track unknown human user's desired trajectory and reduce human force, in presence of time-varying periods.

The proposed method is based on impedance control where the interaction force is modulated by refining the robot's reference trajectory. However, traditional impedance control generates a virtual reference trajectory but its real reference trajectory does not change. In comparison, the proposed ILC updates the robot's real reference trajectory and enables the robot to proactively move to the human partner's desired trajectory with less effort.

6 Conclusions

Unlike iterative learning control in conventional applications such as motion control, the reference trajectory cannot be programmed for the robot in the coupled system of human–robot interaction. Therefore, we introduce a simplified interaction force model to estimate the human partner's desired trajectory and apply the filtered interaction force to update the robot's reference trajectory. Stability of the non-standard iterative learning control system has been shown to be guaranteed by rigorous analysis, where the condition of a fixed period is relaxed for the iterative learning control design. The validity of the proposed method has been verified through two simulations of gait tracking and trajectory learning. A further experiment of trajectory learning on a physical robot proves its feasibility. These results show that the proposed method has a potential to



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Fig. 7 In this figure, different iterations are described by lines in different colors and the end of the trial with a period less than T_d is marked with a dot. The dotted red line represents the desired trajectory. **a** shows the actual interaction force. **b** shows the off-line filtered interaction force. **c**, **d** The curve of the end-effector position over time in the Y-axis direction. The trajectories of end-effector in the vertical plane are given in **e**, **f**. **d**, **f** Generated by fitting interaction force in **b** rather than directly filtering the data in **c**, **e**

Table 1 Experiment parameters

β_0	β_1	β_2	β_3	m
0.2	0.2	0.1	0.1	3.0
m_d (kg)	c_d (N/$_{\text{(m/s)}}$)	k_d (N/m)	k_h (N/m)	
2.0	25.0	26.0	0.25	

Table 2 Learning periods (s)

T_1	T_3	T_5	T_7	T_9	T_{11}	T_{13}	T_{15}
20.0	20.20	18.86	20.38	21.36	19.80	19.47	19.92

reduce human effort in human–robot collaboration and transfer human skills to robots, which can be applied to applications such as heavy load transport in automotive industry and compliant machining.

Appendix

Due to the control objectives of $f_h(t) = 0$ and $e(t) = 0$, we have $f_{h_d}(t) = 0$ and $\varphi_d(t) = 0$, then we obtain

$$\begin{aligned}\dot{\mu}_i(t) &= \dot{\rho}_i(t) - \dot{\rho}_d(t)\\ &= C\dot{\varphi}_i(t)\\ &= C[A\varphi_i(t) + Be_i(t)]\end{aligned} \tag{29}$$

From Eqs. (25) and (21), the following relationship is obtained

$$\begin{aligned}e_{i+1}(t) &= \Phi\{e_i(t)\} + \sum_{j=0}^{m} \beta_j\gamma_{i-j}(t)\dot{\mu}_{i-j}(t)\\ &= \Phi\{e_i(t)\} + \sum_{j=0}^{m} \beta_j\gamma_{i-j}(t)C[A\varphi_{i-j}(t)\\ &\quad + Be_{i-j}(t)]\\ &= \sum_{j=0}^{m}\left[\frac{1}{m+1} - \beta_j\gamma_{i-j}(t)CB\right]e_{i-j}(t)\\ &\quad + \sum_{j=0}^{m} \beta_j\gamma_{i-j}(t)CA\varphi_{i-j}(t)\end{aligned} \tag{30}$$

Taking norm on both sides of Eq. (30) yields

$$\begin{aligned}\|e_{i+1}(t)\| \leq &\sum_{j=0}^{m}\left|\frac{1}{m+1} - CB\beta_j\gamma_{i-j}(t)\right|\|e_{i-j}(t)\|\\ &+ \sum_{j=0}^{m} a|\beta_j|\gamma_{i-j}(t)\|\varphi_{i-j}(t)\|\end{aligned} \tag{31}$$

where a is the upper bound of $\|A\|$.

Taking norm on both sides of system Eq. (14) yields

$$\|\varphi_i(t)\| \leq \int_0^t a\|\varphi_i(\tau)\| + b\|e_i(\tau)\|d\tau \tag{32}$$

where $b \geq \| B \|$. By applying Bellman–Gronwall lemma, we have

$$\begin{aligned}\|\varphi_i(t)\| &\leq be^{at}\int_0^t \|e_i(\tau)\|d\tau\\ &= be^{at}\int_0^t \|e_i(\tau)\|e^{-\lambda\tau}e^{\lambda\tau}d\tau\\ &\leq be^{at}\|e_i(t)\|_\lambda \int_0^t e^{\lambda\tau}d\tau\\ &= be^{at}\frac{e^{\lambda t} - 1}{\lambda}\|e_i(t)\|_\lambda\end{aligned} \tag{33}$$

According to the definition of λ-norm, it can be obtained that $\|e_i(t)\|_\lambda = \sup_{t\in[0,T_d]}\{e^{-\lambda t}\|e_i(t)\|_\infty\}, \lambda > 0$. Then, substituting (33) into (31) implies that

$$\|e_{i+1}(t)\| \leq \sum_{j=0}^{m}\left|\frac{1}{m+1} - CB\beta_j\gamma_{i-j}(t)\right|\|e_{i-j}(t)\| \\ + abe^{at}\frac{e^{\lambda t}-1}{\lambda}\sum_{j=0}^{m}|\beta_j|\gamma_{i-j}(t)\|e_{i-j}(t)\|_\lambda \tag{34}$$

Introducing the expectation operator $\mathbf{E}[\cdot]$ to Eq. (34) and noting that only $\gamma_i(t)$ is the stochastic variable, we can rewrite Eq. (34) as

$$\|e_{i+1}(t)\| \leq \sum_{j=0}^{m}\mathbf{E}\left\{\left|\frac{1}{m+1} - CB\beta_j\gamma_{i-j}(t)\right|\right\}\|e_{i-j}(t)\| \\ + abe^{at}\frac{e^{\lambda t}-1}{\lambda}\sum_{j=0}^{m}|\beta_j|\mathbf{E}\{\gamma_{i-j}(t)\}\|e_{i-j}(t)\|_\lambda \tag{35}$$

Based on the definition of stochastic variable $\gamma_i(t)$ and the mathematical expectation, it follows

$$\begin{aligned}\mathbf{E}\left\{\left|\frac{1}{m+1} - CB\beta_j\gamma_{i-j}(t)\right|\right\} &= \left|\frac{1}{m+1} - 1\cdot\beta_j CB\right|q(t) \\ &\quad + \left|\frac{1}{m+1} - 0\cdot\beta_j CB\right|(1-q(t)) \\ &= \left|\frac{1}{m+1} - \beta_j CB\right|q(t) + \frac{1-q(t)}{m+1}\end{aligned} \tag{36}$$

then

$$\|e_{i+1}(t)\| \leq \sum_{j=0}^{m}\left\{\left|\frac{1}{m+1} - CB\beta_j\right|q(t) + \frac{1-q(t)}{m+1}\right\}\|e_{i-j}(t)\| \\ + abq(t)e^{at}\frac{e^{\lambda t}-1}{\lambda}\sum_{j=0}^{m}|\beta_j|\|F_{h_{i-j}}(t)\|_\lambda \tag{37}$$

From (37) and the definition of λ-norm, we obtain

$$\|e_{i+1}(t)\|_\lambda \leq \sum_{j=0}^{m}\sup_{t\in[0,T_d]}\left\{\left|\frac{1}{m+1} - CB\beta_j\right|q(t) + \frac{1-q(t)}{m+1}\right\}\|e_{i-j}(t)\|_\lambda \\ + abq(t)e^{aT_d}\frac{e^{\lambda T_d}-1}{\lambda}\sum_{j=0}^{m}|\beta_j|\|e_{i-j}(t)\|_\lambda \tag{38}$$

Define

$$\delta \triangleq abq(t)e^{aT_d}\frac{e^{\lambda T_d}-1}{\lambda}\sum_{j=0}^{m}|\beta_j| \tag{39}$$

From *Theorem 1* with (38) and (39), we have

$$\|e_{i+1}(t)\|_\lambda \leq \left(\sum_{j=0}^{m}\alpha_j + \delta\right)max\{\|e_i\|_\lambda, \|e_{i-1}\|_\lambda, \ldots, \|e_{i-m+1}\|_\lambda\} \tag{40}$$

where $0 < q(t) < 1$ is applied. When λ is sufficiently large, δ can be made as small as possible. According to Eq. (27), it yields that $\sum_{j=0}^{m}\alpha_j + \delta \leq \epsilon + \delta < 1$, so we have $\lim_{i\to\infty} e_i = 0$ and the control objective $x_d(t) = x_h(t)$ is achieved.

According to the convergence of e_i and the inequality (33), it is obvious that

$$\lim_{i\to\infty}\|\varphi_i(t)\| = 0 \tag{41}$$

so we can get $\lim_{i\to\infty} f_{h_i}(t) = 0$ which indicates that $\lim_{i\to\infty} x_i(t) = x_h(t)$ and the robot tracks the human's desired trajectory.

References

1. Sheridan, T.B.: Human–robot interaction: status and challenges. Human Fact. **4**, 58 (2016)
2. Bauer, A., Wollherr, D., Buss, M.: Human-robot collaboration: a survey. Int. J. Humanoid Rob. **5**(1), 47–66 (2008)
3. Tsarouchi, P., Makris, S., Chryssolouris, G.: Human-robot interaction review and challenges on task planning and programming. Int. J. Comput. Integr. Manuf. **29**(8), 916–931 (2016)
4. Brown, J.D., O'Brien, C.E., Leung, S.C., Dumon, K.R., Lee, D.I., Kuchenbecker, K.J.: Using contact forces and robot arm accelerations to automatically rate surgeon skill at peg transfer. IEEE Trans. Biomed. Eng. **64**(9), 2263–2275 (2017)
5. Jakopin, B., Mihelj, M., Munih, M.: An unobtrusive measurement method for assessing physiological response in physical human–robot interaction. IEEE Trans. Human Mach. Syst. **47**(4), 474–485 (2017)
6. Yu, H., Huang, S., Chen, G., Pan, Y., Zhao, G.: Human-robot interaction control of rehabilitation robots with series elastic actuators. IEEE Trans. Rob. **31**(5), 1089–1100 (2015)
7. Wang, W., Li, R., Chen, Y., Diekel, Z.Max, Jia, Y.: Facilitating human-robot collaborative tasks by teaching–learning-collaboration from human demonstrations. IEEE Trans. Autom. Sci. Eng. **16**(2), 640–653 (2018)
8. Zhang, Z., Wang, W., Chen, Y., Jia, Y., Peng, G.: Prediction of human actions in assembly process by a spatial–temporal end-to-end learning model. SAE Technical Paper, no. 2019-01-0509 (2019)
9. Chen, Y., Wang, W., Zhang, Z., Kroi, V., Jia, Y.: Modeling and learning of object placing tasks from human demonstrations in smart manufacturing. SAE Technical Paper, no. 2019-01-0700 (2019)
10. Li, Y., Ge, S.S.: Human–robot collaboration based on motion intention estimation. IEEE/ASME Trans. Mechatron. **19**(3), 1007–1014 (2013)
11. Losey, D., O'Malley, M.: Trajectory deformations from physical human–robot interaction. IEEE Trans. Rob. **34**(1), 126–138 (2017)
12. Evrard, P., Gribovskaya, E., Calinon, S., Billard, A., Kheddar, A.: Teaching physical collaborative tasks: object-lifting case study with a humanoid. In: IEEE-RAS International Conference on Humanoid Robots (2010)

13. Villani, V., Pini, F., Leali, F., Secchi, C.: Survey on human–robot collaboration in industrial settings: safety, intuitive interfaces and applications. Mechatronics (2018)
14. Rozo, L., Calinon, S., Caldwell, D.G., Jimenez, P., Torras, C.: Learning physical collaborative robot behaviors from human demonstrations. IEEE Trans. Rob. **32**(3), 513–527 (2016)
15. Cherubini, A., Passama, R., Crosnier, e a: Collaborative manufacturing with physical human–robot interaction. Robot. Comput. Integrat. Manuf. **40**, 1–13 (2016)
16. Rozo, L., Jimenez, P., Torras, C.: Robot learning from demonstration of force-based tasks with multiple solution trajectories. In: International Conference on Advanced Robotics (2011)
17. Erden, M.S., Tomiyama, T.: Human-intent detection and physically interactive control of a robot without force sensors. IEEE Trans. Rob. **26**(2), 370–382 (2010)
18. Hogan, N.: Impedance control: an approach to manipulation-part 1: Theory; part 2: Implementation; part 3: Applications. J. Dyn. Syst. Meas. Contr. **107**(1), 1–24 (1985)
19. Sam Ge, S., Li, Y., Wang, C.: Impedance adaptation for optimal robot–environment interaction. Int. J. Control **87**(2), 249–263 (2014)
20. Seraji, H., Colbaugh, R.: Force tracking in impedance control. Int. J. Robot. Res. **16**(2), 499–506 (1993)
21. Lu, W.S., Meng, Q.H.: Impedance control with adaptation for robotic manipulations. IEEE Trans. Robot. Autom. **7**(3), 408–415 (1991)
22. Li, Y., Ge, S.S.: Impedance learning for robots interacting with unknown environments. IEEE Trans. Control Syst. Technol. **22**(4), 1422–1432 (2014)
23. Burdet, E., Osu, R., Franklin, D.W., Milner, T.E., Kawato, M.: The central nervous system stabilizes unstable dynamics by learning optimal impedance. Nature **414**(6862), 446–449 (2001)
24. Noohi, E., Zefran, M., Patton, J.L.: A model for human-human collaborative object manipulation and its application to human–robot interaction. IEEE Trans. Robot. **32**(4), 1–17 (2016)
25. Peternel, L., Petri, T., Oztop, E., Babi, J.: Teaching robots to cooperate with humans in dynamic manipulation tasks based on multi-modal human-in-the-loop approach. Auton. Robots **36**, 1–14 (2014)
26. Tayebi, A.: Adaptive iterative learning control for robot manipulators (2004)
27. Aubin, P.M., Cowley, M.S., Ledoux, W.R.: Gait simulation via a 6-dof parallel robot with iterative learning control. IEEE Trans. Biomed. Eng. **55**(3), 1237–1240 (2008)
28. He, X., Qin, Z., Wu, X.: Adaptive iterative learning control of robot manipulators in the presence of environmental constraints. Control Theory Appl. **20**, 133–150 (2012)
29. Bristow, D.A., Tharayil, M., Alleyne, A.G.: A survey of iterative learning. IEEE Control Syst. **26**(3), 96–114 (2006)
30. Wang, L., Li, X., Shen, D.: Sampled-data iterative learning control for continuous-time nonlinear systems with iteration-varying lengths. Int. J. Robust Nonlinear Control **28**(8), 3073–3091 (2018)
31. Bu, X., Wang, S., Hou, Z., Liu, W.: Model free adaptive iterative learning control for a class of nonlinear systems with randomly varying iteration lengths. J. Franklin Inst. **356**(5), 2491–2504 (2019)
32. Liu, C., Shen, D., Wang, J.: Adaptive learning control for general nonlinear systems with nonuniform trial lengths, initial state deviation, and unknown control direction. Int. J. Robust Nonlinear Control **20**, 20 (2019)
33. Li, X., Xu, J., Huang, D.: An iterative learning control approach for linear systems with randomly varying trial lengths. IEEE Trans. Autom. Control **59**(7), 1954–1960 (2014)
34. Li, X., Xu, J.-X., Huang, D.: Iterative learning control for nonlinear dynamic systems with randomly varying trial lengths. Int. J. Adapt. Control Signal Process. **29**(11), 1341–1353 (2015)
35. Shen, D., Xu, J.: Adaptive learning control for nonlinear systems with randomly varying iteration lengths. IEEE Trans. Neural Netw. Learn. Syst. **30**(4), 1119–1132 (2019)
36. Feldman, A.G.: Once more on the equilibrium-point hypothesis (λ model) for motor control. J. Mot. Behav. **18**(1), 17–54 (1986)
37. Chen, Y., Lin, S., Li, R.H., Zhang, L.D.: Kinematics characteristic of exoskeleton walking robot for older people. Appl. Mech. Mater. **644**, 167–170 (2014)
38. Veneman, J.F., Rik, K., Hekman, E.E.G., et al.: Design and evaluation of the lopes exoskeleton robot for interactive gait rehabilitation. IEEE Trans. Neural Syst. Rehabil. Eng. **15**(3), 379–386 (2007)

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

Jingkang Xia received the B.S. degree in electrical Information and Engineering, from the School of Electrical Engineering, Southwest Minzu University, Chengdu, China, in 2018. He is currently pursuing the Ph.D. degree in control theory and application with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China. His current research interests include human-robot collaboration, robotics and intelligent control systems.

Deqing Huang received the B.S. and Ph.D. degrees in applied mathematics from the Mathematical College, Sichuan University, Chengdu, China, in 2002 and 2007, respectively, the Ph.D. degree in control engineering from the National University of Singapore (NUS), Singapore, in 2011. From 2013 to 2016, he was a Research Associate with the Department of Aeronautics, Imperial College London, London, U.K. In 2016, he joined the Department of Electronic and Information Engineering, Southwest Jiaotong University, Chengdu, as a Professor and the Department Head. His current research interests include modern control theory, fluid analysis and control, convex optimization, and robotics.

Yanan Li received the B.Eng. and M.Eng. degrees from the Harbin Institute of Technology, China, in 2006 and 2008, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2013. He is currently a Lecturer in control engineering with the Department of Engineering and Design, University of Sussex, Sussex, U.K. From 2015 to 2017, he was a Research Associate with the Department of Bioengineering, Imperial College London, London, U.K. From 2013 to 2015, he was a Research Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. His general research interests include human–robot interaction, robot control and control theory and applications.

Na Qin received the B.S. degree in electrical technology from the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China, in 2000, and the M.S. degree in electric power system and automation and the Ph.D. degree in electrical engineering from the School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China, in 2003 and 2014, respectively. She is currently an Associate Professor with the School of Electrical Engineering, Southwest Jiaotong University. Her research interests include intelligent information processing, fault diagnosis, pattern recognition, and intelligent systems.