Dynamic Motion Primitives-based Trajectory Learning for Physical Human-Robot Interaction Force Control

Xueyan Xing, Kamran Maqsood, Chao Zeng, Chenguang Yang, *Senior Member*, Shuai Yuan, and Yanan Li, *Senior Member*

***Abstract*—One promising function of interactive robots is to provide a specific interaction force to human users. For example, rehabilitation robots are expected to promote patients' recovery by interacting with them with a prescribed force. However, motion uncertainties of different individuals, which are hard to predict due to the varying motion speed and noises during motion, degrade the performance of existing control methods. This paper proposes a method to learn a desired reference trajectory for a robot based on dynamic motion primitives (DMPs) and iterative learning (IL). By controlling the robot to follow the generated desired reference trajectory, the interaction force can achieve a desired value. In our proposed approach, DMPs are first employed to parameterize the demonstration trajectories of the human user. Then a recursive least square (RLS)-based estimator is developed and combined with the Adam optimization method to update the trajectory parameters so that the desired reference trajectory of the robot is iteratively obtained by resolving the DMPs. Since the proposed method parameterizes the trajectories depending on the phrase variable, it removes the essential assumption of traditional IL methods that the iteration period should be invariant, and thus has improved robustness compared with the existing methods. Experiments are performed using an interactive robot to validate the effectiveness of our proposed scheme.**

***Index Terms*—Interaction force, physical human-robot interaction (pHRI), DMPs, iterative learning.**

I. Introduction

With the advent of interactive robots, they are expected to perform tasks according to needs of human users and realize efficient collaboration in different scenarios of physical human-robot interaction (pHRI). To do so, many strategies have been developed for interactive robots to adjust their control in view of human behaviors [1]–[5]. In many scenarios of pHRI, achieving a desired interaction force between a human user and a robot is expected, e.g. the requirements of a rehabilitation robot may include achieving a specific interaction force that is prescribed by a physiotherapist [6], [7]. Besides, control of interaction force is also needed by exoskeleton robots for the purpose of effective assistance [8]. Consequently, it is critical to design an effective robotic interaction force control. However, the uncertain motion and unknown motion behavior of different individuals raise challenges for control of HRI force.

Force control [9] and iterative learning control (ILC) [10], [11] with sensed force-feedback can be used for generating a desired interaction force. ILC-based force control offers good performance due to its model-free nature and capability of uncertainty compensation [12]. However, the existing ILC used for pHRI is restricted by the assumption that the movement speed should be fixed for each iteration, so its performance is sensitive to the unpredictable human behaviors. To remove this limitation, [13] proposes a control method to deal with the uncertainty caused by the varying iteration period by introducing a maximum execution time, so that the varying motion speed of the human user is allowed. However, the control method in [13] is based on the assumption of known iteration periods, which is avoided by [14]. In [14], with the proposed spacial ILC, robot's learning is space-based instead of time-based, and a desired interaction force is thus achieved with varying motion speed. To further reduce the time-related uncertainties, [15] designs a performance index function and iteratively updates the robot's parameterized trajectory for a desired interaction force. Although this method has good robustness against human motion uncertainties, it needs to know the shape of the robot's desired trajectory in advance, which limits its applications.

Dynamic motion primitive (DMPs) is a parameterized policy that can mathematically encode a movement into nonlinear dynamic systems, which can be combined with impedance/admittance control for force control. In [16], a model-free reinforcement learning scheme is combined with DMPs to learn the end-effector trajectories and variable impedance in force fields to endow robots with human behaviors for disturbance rejection. In [17], the motion and stiffness are learnt by the robot with DMPs during demonstrations, which enables the robot to autonomously execute the task according to task requirements. In [18], EMG signals are

This work was supported in part by the U.K. EPSRC under Grant EP/T006951/1, in part by the Chinese State Key Laboratory of Robotics and Systems (HIT) under Grant SKLRS-2022-KF-16, and in part by the Fundamental Research Funds for the Central Universities under Grant AUEA5740300122. (Xueyan Xing and Kamran Maqsood contributed equally to the work.) (Corresponding author: Yanan Li.)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by (the Sciences and Technology Cross-Schools Research Ethics Committee of Sussex University) (IF PROVIDED under Application No. ER/YL557/1).

Xueyan Xing, Kamran Maqsood, and Yanan Li are with the Department of Engineering and Design, University of Sussex, BN1 9RH Brighton, U.K. (e-mail: xx92@sussex.ac.uk; k.maqsood@sussex.ac.uk; hit.li.yn@gmail.com).

Chao Zeng is with the University of Hamburg, 22527 Hamburg, Germany (e-mail: chaozeng@ieee.org).

Chenguang Yang is with the Bristol Robotics Laboratory, University of the West of England, BS16 1QY Bristol, U.K. (e-mail: cyang@ieee.org).

Shuai Yuan is with the School of Astronautics, Harbin Institute of Technology, Harbin, 150001, China (e-mail: shuaiyuan@hit.edu.cn).

used for stiffness estimation, and trajectories and stiffness profiles are encoded in a unified way at the same time to achieve a more complete skill transfer process. A modified DMPs method is developed in [19] by adding a scaling factor and a force coupling term that is derived from the adaptive admittance control. With the modified strategy, trajectories can be produced on the curved surface with better force control performance. However, [16]–[19] largely concentrate on how to improve robots' behavior with DMPs by skill transfer. How to make robots efficiently cooperate with human users to achieve a task in pHRI is still a critical issue that needs to be further explored.

In pHRI, DMPs-based modeling is a powerful tool to effectively capture and reproduce human behaviors by human demonstrations. With the estimated human behaviors obtained by DMPs, robot can adjust its movement or impedance to meet the needs of humans with properly designed robotic controllers. The works in [20]–[22] focus on DMPs-based skill transfer from the perspective of trajectory learning and trajectory adaptation that do not involve force learning. In order to further improve the efficiency of pHRI, progress has also been made in force learning on the basis of DMPs. The force profile of humans is replicated and learned by the robot in [23] by minimizing a quadratic cost function of force error based on DMPs so that the robot can execute a cooperative task with human with a desired force. In [24], demonstration trajectory and force of human are first obtained by using DMPs and then a robotic ILC is proposed to compensate for the position offset for force adaptation in human-robot cooperation. These two works offer new ideas of combining DMPs and ILC to generate a desired interaction force to complete the task. However, [23] and [24] separately use DMPs and ILC, where DMPs are employed for the generation of the initial trajectory and ILC is used for the following update of the reference trajectory. Besides, [23] and [24] consider the force compensated by the designed robotic controller to complete a specific task instead of the interaction force, which becomes the major motivation of our proposed scheme.

Besides the iterative learning method, machine learning methods, e.g., imitation learning and reinforcement learning [25]–[27], which are commonly utilized for human-robot interaction, can be used to learn the human behavior based on DMPs. Compared with the existing machine learning methods, which come from the computer science field, the iterative learning method originates from the control field and its stability and performance can be rigorously analyzed. Different from the machine learning method that needs considerable offline demonstrations, the online iterative learning method can address uncertainties during human-robot interaction.

In this article, a DMPs-based high-level controller is developed to achieve a specific interaction force by formulating a reference trajectory of the robot. Here, we use the Adam optimization method [28] to minimize the force error based on a recursive least square (RLS)-based estimator. A low-level control is designed to realize trajectory tracking. The major contributions of this paper can be summarized as follows:

1) DMP is applied in this paper to encode the robotic reference trajectory, and thus the essential assumption of the traditional ILC, i.e., the iteration period is fixed [29] that is hard to guarantee in pHRI, is not required in our proposed strategy. Consequently, each iteration allows to have a different duration with our proposed method and the interaction force can arrive at the specific value regardless of human's motion speed.

2) Since the iterative update of the robot's reference trajectory depends on the phrase variable, our proposed algorithm has the capability against the time-related uncertainty and has better robustness than the traditional method that updates each position point on the robot's reference trajectory.

3) Compared with [15], the shape of the robot's desired reference trajectory does not need to be known as a priori, which significantly expands its application domains.

The rest of the paper is structured as follows. The methodology of the DMPs-based trajectory learning for human-robot interaction force control is introduced in Section II in detail, where Section II-A presents the design of the low-level robotic controller and a DMPs-based high-level controller is proposed in Section II-B including the design steps of the parameterization of the initial trajectory of the human user, the update of the parameters and the robot's reference trajectory. Section III shows the experimental results with comparisons and multiple subjects to prove the effectiveness of the proposed algorithm. Section IV discusses how to improve the robustness of the low-level controller and Section V finally summarizes this paper.

Notation: In this paper, $\Re^n$ and $\Re^{n\times m}$ stand for n-dimensional column vectors and $n\times m$ dimensional matrices, respectively. Vectors and matrices are represented as bold small and capital symbols, respectively, and they are not bold when $n=1$ for vectors and $n=m=1$ for matrices. The $n\times n$ identity matrix is denoted by $\boldsymbol{I}_n$. The circle operator $\circ$ denotes element-wise product of two vectors. For instance, if vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are defined as $\boldsymbol{a}=[a_1,\cdots,a_n]$ and $\boldsymbol{b}=[b_1,\cdots,b_n]$, we have $\boldsymbol{a}\circ\boldsymbol{b}=[a_1b_1,\cdots,a_nb_n]$.

II. Methodology

In this section, a DMPs-based trajectory learning method is designed for the interaction force control to meet some specific requirements as mentioned in the introduction section. With its help, the actual iteration force will converge to a desired value. The whole process is summarized by Fig. 1A, which illustrates that pHRI is realized via physical contact. Based on the measurable interaction force, the proposed DMPs-based trajectory learning method enables generation of a desired robot's reference trajectory, which first needs several offline demonstration trajectories of the human user, and then requires online pHRI for learning. Details of the designed DMPs-based trajectory learning method will be introduced in the following. By tracking the generated desired reference trajectory with a low-level control, a specific interaction force will be achieved between the human user and the robot so that the robot can assist the human user with a certain movement. The framework of the developed method is illustrated by Fig. 1B.

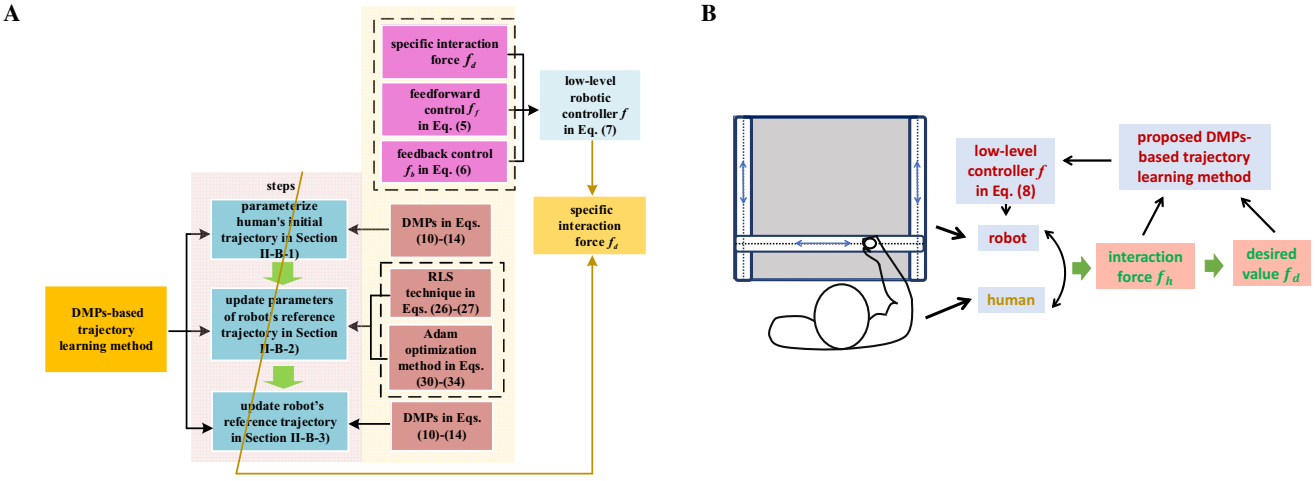

Fig. 1: A scenario of pHRI with our proposed scheme (A) and the proposed DMPs-based trajectory learning method (B).

A. Low-Level Robotic Controller Design

A low-level controller is first designed for reference trajectory tracking of the robot. To simplify the analysis, the following robot kinematics of 2D robot is considered

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}} \tag{1}$$

where $\boldsymbol{J}(\boldsymbol{q}) \in \Re^{2\times 2}$ is the Jacobian matrix that is square and invertible, $\boldsymbol{x} \in \Re^2$ is the position of the robot in the Cartesian space, and $\boldsymbol{q} \in \Re^2$ is the coordinate in the joint space.

The dynamics of the robot manipulator in the joint space can be expressed as

$$\boldsymbol{M}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{q})\,\boldsymbol{f_h} \tag{2}$$

in which $\boldsymbol{M}(\boldsymbol{q}) \in \Re^{2\times 2}$ denotes the inertia matrix of the manipulator, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \Re^{2\times 2}$ is a matrix representing the Coriolis and Centrifugal force, $\boldsymbol{G}(\boldsymbol{q}) \in \Re^2$ stands for the gravitational force, $\boldsymbol{\tau} \in \Re^2$ denotes the control torque, and $\boldsymbol{f_h} \in \Re^2$ is the force exerted by the human user.

Eq. (2) can be rewritten as

$$\bar{\boldsymbol{M}}(\boldsymbol{q})\,\ddot{\boldsymbol{x}} + \bar{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\,\dot{\boldsymbol{x}} + \bar{\boldsymbol{G}}(\boldsymbol{q}) = \boldsymbol{f} + \boldsymbol{f_h} \tag{3}$$

where $\bar{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{J}^{-\mathrm{T}}(\boldsymbol{q})\,\boldsymbol{M}(\boldsymbol{q})\,\boldsymbol{J}^{-1}(\boldsymbol{q})$, $\bar{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{J}^{-\mathrm{T}}(\boldsymbol{q})\left(\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{M}(\boldsymbol{q})\,\boldsymbol{J}^{-1}(\boldsymbol{q})\,\dot{\boldsymbol{J}}(\boldsymbol{q})\right)\boldsymbol{J}^{-1}(\boldsymbol{q})$, $\bar{\boldsymbol{G}}(\boldsymbol{q}) = \boldsymbol{J}^{-\mathrm{T}}(\boldsymbol{q})\,\boldsymbol{G}(\boldsymbol{q})$, and $\boldsymbol{f} = \boldsymbol{J}^{-\mathrm{T}}(\boldsymbol{q})\,\boldsymbol{\tau}$ represents the control force in the Cartesian space.

Define the tracking error as

$$\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x_r} \tag{4}$$

where $\boldsymbol{x_r} \in \Re^2$ is the reference trajectory of the robot. Combing a feedforward control term $\boldsymbol{f_f}$ designed to compensate for the undesired dynamics of the system

$$\boldsymbol{f_f} = \bar{\boldsymbol{M}}(\boldsymbol{q})\,\ddot{\boldsymbol{x}}_{\boldsymbol{r}} + \bar{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\,\dot{\boldsymbol{x}} + \bar{\boldsymbol{G}}(\boldsymbol{q}) - \boldsymbol{f_d} \tag{5}$$

and a feedback control term $\boldsymbol{f_b}$, which is also a PD control, to stabilize the system

$$\boldsymbol{f_b} = -\bar{\boldsymbol{M}}(\boldsymbol{q})\,\boldsymbol{K_d}\dot{\boldsymbol{e}} - \bar{\boldsymbol{M}}(\boldsymbol{q})\,\boldsymbol{K_p}\boldsymbol{e} \tag{6}$$

the robotic controller is proposed as

$$\boldsymbol{f} = \boldsymbol{f_f} + \boldsymbol{f_b} \tag{7}$$

where $\boldsymbol{f_d}$ is the specified interaction force, $\boldsymbol{K_d} \in \Re^{2\times 2}$ and $\boldsymbol{K_p} \in \Re^{2\times 2}$ are positive control gains, i.e., parameters of the PD feedback control term, which need to be designed for control.

Define the interaction force error as

$$\boldsymbol{f_e} = \boldsymbol{f_h} - \boldsymbol{f_d} \tag{8}$$

and then considering Eqs. (3) and (5)-(8) yields

$$\ddot{\boldsymbol{e}} + \boldsymbol{K_d}\dot{\boldsymbol{e}} + \boldsymbol{K_p}\boldsymbol{e} = \bar{\boldsymbol{M}}^{-1}(\boldsymbol{q})\,\boldsymbol{f_e}. \tag{9}$$

Apparently, by appropriately setting $\boldsymbol{K_d}$ and $\boldsymbol{K_p}$, if $\boldsymbol{f_e}$ is able to converge to zero with the help of the high-level control which generates a desired reference trajectory for the robot, which is denoted by $\boldsymbol{x_d}$, this designed low-level robotic control law in Eq. (9) ensures the stability of the interaction system.

Remark 1: In the feedforward control term in Eq. (5), $\ddot{\boldsymbol{x}}_{\boldsymbol{r}}$ is used to achieve the closed-loop system including position and force errors in Eq. (9) rather than $\ddot{\boldsymbol{x}}$, which is hard to be precisely obtained during human-robot interaction.

Remark 2: For a general case of a n-DoF (degree of freedom) robot where the Jacobian matrix is not square, the issue of Jacobian invertibility needs to be further considered in the design of low-level controller [30], which is not the focus of this paper and thus is not discussed here.

B. DMPs-based High-Level Controller

In this subsection, the proposed DMPs-based high-level controller is designed. Following the robot's desired trajectory $\boldsymbol{x_d}$, the specific interaction force between human and robot can be generated. To facilitate the analysis, only 1-DoF is considered in the following, and it can be extended to the case of multiple dimensions by introducing DMPs to each dimension as in [31].

1) Parameterizing the Initial Trajectory of the Human User: In this subsection, DMPs are adopted to encode and parameterize offline demonstration trajectories of the human user to get the user's initial trajectory, which is also his/her contact-free trajectory without any external force. The obtained trajectory parameters will be served as initial values of the iterative learning laws in the next part.

A typical discrete DMP for a 1-DoF trajectory can be described by

$$\tau\dot{\varrho} = \alpha\left(\beta\left(z_g - z\right) - \varrho\right) + \Psi\left(s\right) \tag{10}$$

$$\tau\dot{z} = \varrho \tag{11}$$

where z and ϱ are the position and the velocity of the DMP system, z_g is the target position, τ is a time constant used for temporal scaling, with which the duration of the motion can be adjusted, positive constants α and β that can be respectively regarded as the spring and damping coefficients determine the behavior of the DMP system, the phase variable s enables the forcing term $\Psi\left(s\right)$ to be active in a finite time window and it is obtained from the following canonical system

$$\tau\dot{s} = -cs \tag{12}$$

where the positive constant c is a decay factor. When s converges from the initial value, which is typically set as a positive constant, to zero with time, $\Psi\left(s\right)$ will correspondingly trend to zero and then the attractive point $[z_g, 0]$ can be achieved for $[z, \varrho]$.

The nonlinear function $\Psi\left(s\right)$ in Eq. (10) is expressed as

$$\Psi\left(s\right) = \frac{\sum\limits_{i=1}^{N}\xi_{0,i}\psi_i\left(s\right)}{\sum\limits_{i=1}^{N}\psi_i\left(s\right)}s\left(z_g - z_0\right) \tag{13}$$

which is a linear combination of N basis functions defined as

$$\psi_i\left(s\right) = \exp\left(-0.5b_i(s-\mu_i)^2\right), i = 1, \cdots, N \tag{14}$$

where b_i and μ_i, $i = 1, \cdots, N$, are positive constants that determine the width and the centers of $\psi_i\left(s\right)$, respectively, z_0 is the initial state of z at $t = 0$, and $\xi_{0,i}$ denotes the adjustable weights of $\psi_i(s)$ that can be obtained by minimizing the following error function

$$\min\left\|\Psi_{\text{target}}\left(s\right) - \Psi\left(s\right)\right\|^2 \tag{15}$$

with $\Psi_{\text{target}}\left(s\right)$ defined as

$$\Psi_{\text{target}}\left(s\right) = \tau\dot{\varrho}_{demo} - \alpha\left(\beta\left(z_g - z_{demo}\right) - \varrho_{demo}\right) \tag{16}$$

where z_{demo} and ϱ_{demo} are respectively the position and velocity of the human that can be collected from offline human demonstrations.

Remark 3: With DMPs described by the nonlinear differential equations in Eqs. (10) and (11) [32]–[34], a discrete movement, i.e., point-to-point movement, is generated, so that DMP in Eqs. (10) and (11) is also called discrete DMP. Observing Eqs. (10) and (11), it is known that they represent a simple second-order system (a damped spring model) with a PD-like controller and its velocity is modified by a linear combination of N basis functions expressed by a forcing term $\Psi\left(s\right)$ in Eq. (13). With $\Psi\left(s\right)$ being phasic, the simple second-order system becomes a point attractive system and the attractor landscape can be specified for the trajectory z towards the goal z_g.

Remark 4: Referring to [34], μ_i and b_i in Eq. (14) can be designed as $\mu_i = \exp\left(-c\frac{i-1}{N-1}\right)$ and $b_i = \frac{1}{\left(\mu_{i+1}-\mu_i\right)^2}$, which are adopted in the following experiment in Section III.

The weighted linear regression problem in Eq. (15) can be resolved by applying regression methods, e.g. locally weighted regression. With the solution $\boldsymbol{\xi}_0 = \left[\xi_{0,1}, \cdots, \xi_{0,N}\right]^{\mathrm{T}} \in \Re^N$, the initial trajectory of the human user ϱ can be achieved using Eqs. (10) and (11). By making use of the typical discrete DMPs, the initial trajectory of the human user is thus parameterized.

2) Updating Parameters of the Reference Trajectory of the Robot: In this step, we will iteratively update the parameter $\boldsymbol{\xi}_j$ of the forcing term $\Psi\left(s\right)$ online. Since the parameters of the forcing term $\Psi(s)$ are constants, $\boldsymbol{\xi}_j$ remain unchanged in the jth iteration and are iteratively updated according to their values in the previous iteration. With these parameters, a reference trajectory of the robot is generated with DMPs to achieve the specific interaction force.

According to human motor control [35], the interaction force from human user can be modeled as $\boldsymbol{f_h} = \boldsymbol{K_h}\left(\boldsymbol{x} - \boldsymbol{x_h}\right)$, where $\boldsymbol{x_h}$ is the desired position of the human user, and $\boldsymbol{K_h}$ is the stiffness of the human arm. Since $\boldsymbol{x}$ can follow $\boldsymbol{x_r}$ and $\boldsymbol{x_r}$ generated by Eqs. (35)-(37) is related to $\boldsymbol{\xi}_j$ in the jth iteration, $\boldsymbol{f_h}_{,j}$ can be conceived as a $\boldsymbol{\xi}_j$-related function. In other words, when the robot updates its reference trajectory by updating trajectory parameters, the interaction force will be correspondingly updated because of the change of the current position $\boldsymbol{x}$. If the specific interaction force is defined as f_d that allows to be a time-related function, the interaction force error in the jth iteration is then expressed as

$$f_e\left(\boldsymbol{\xi}_j\right) = f_h\left(\boldsymbol{\xi}_j\right) - f_d \tag{17}$$

where $j = 1, 2, \cdots, M$ with M being the iteration number, $\boldsymbol{\xi}_j = \left[\xi_{j,1}, \cdots, \xi_{j,N}\right]^{\mathrm{T}} \in \Re^N$, and f_h is the interaction force induced by the human that is collected online.

Then we develop a performance index to guide the iterative learning of $\boldsymbol{\xi}_j$ in terms of minimizing the interaction force error

$$g\left(\boldsymbol{\xi}_j\right) = \int_0^{T_j}\left\|f_e\left(\boldsymbol{\xi}_j\right)\right\|\mathrm{d}t \tag{18}$$

in which T_j represents the jth iteration duration.

If the gradient of the index function denoted by $\boldsymbol{\nabla} g\left(\boldsymbol{\xi}_j\right) \in \Re^N$ between iterations is known, the Adam method [28], i.e. a gradient-based optimization algorithm, is applied to find the optimal solution $\boldsymbol{\xi}^* = \lim\limits_{j\to\infty}\boldsymbol{\xi}_j$, $\boldsymbol{\xi}^* = \left[\xi_1^*, \cdots, \xi_N^*\right]^{\mathrm{T}} \in \Re^N$, that minimizes the index function in Eq. (16) as follows

$$\boldsymbol{\varepsilon}_j = \kappa_\varepsilon\boldsymbol{\varepsilon}_{j-1} + \left(1 - \kappa_\varepsilon\right)\boldsymbol{\nabla} g\left(\boldsymbol{\xi}_j\right) \tag{19}$$

$$\lambda_j = \kappa_\lambda \lambda_{j-1} + (1-\kappa_\lambda)\, \boldsymbol{\nabla} g^{\mathrm{T}}(\boldsymbol{\xi}_j)\, \boldsymbol{\nabla} g(\boldsymbol{\xi}_j) \tag{20}$$

$$\boldsymbol{\varepsilon}'_j = \frac{\boldsymbol{\varepsilon}_j}{1-\kappa_\varepsilon} \tag{21}$$

$$\lambda'_j = \frac{\lambda_j}{1-\kappa_\lambda} \tag{22}$$

$$\boldsymbol{\xi}_j = \boldsymbol{\xi}_{j-1} - \frac{\kappa_\xi}{\sqrt{\lambda'_j + \eta}} \boldsymbol{\varepsilon}'_j \tag{23}$$

where $\boldsymbol{\xi}^*$ is the parameter of the desired reference trajectory x_d, κ_ε , κ_λ and κ_ξ are positive constants, $\boldsymbol{\varepsilon}'_j \in \Re^N$ and $\lambda'_j \in \Re$ are bias-corrections of moments $\boldsymbol{\varepsilon}_j \in \Re^N$ and $\lambda_j \in \Re$ respectively, and η is a positive constant for avoiding singularity.

In Eq. (23), the initial iteration value of the trajectory parameter $\boldsymbol{\xi}_j$, which is also the initial trajectory parameter of the human user $\boldsymbol{\xi}_0$, is obtained from Section II-B-1. However, $\boldsymbol{\nabla} g(\boldsymbol{\xi}_j)$ in Eqs. (19) and (20) can not be directly calculated in the case of pHRI. To obtain $\boldsymbol{\nabla} g(\boldsymbol{\xi}_j)$, an estimator is developed to endow the performance index function $g(\boldsymbol{\xi}_j)$ with an explicit mathematical formulation and then $\boldsymbol{\nabla} g(\boldsymbol{\xi}_j)$ can be calculated online and used for the aforementioned learning process in Eqs. (19)-(23).

In this work, the estimator of $g(\boldsymbol{\xi}_j)$ is proposed as

$$\hat{g}(\boldsymbol{\xi}_j) = \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{\xi}_j)\, \boldsymbol{\theta}_j \tag{24}$$

where $\boldsymbol{\phi}(\boldsymbol{\xi}_j)$ is the regressor vector that can be designed using the m-order polynomial as

$$\boldsymbol{\phi}(\boldsymbol{\xi}_j) = \left[1, \boldsymbol{\xi}_j^{\mathrm{T}}, \boldsymbol{\xi}_j^{\mathrm{T}} \circ \boldsymbol{\xi}_j^{\mathrm{T}}, \cdots, \underbrace{\boldsymbol{\xi}_j^{\mathrm{T}} \circ \cdots \circ \boldsymbol{\xi}_j^{\mathrm{T}}}_{m}\right]^{\mathrm{T}} \tag{25}$$

in which $\underbrace{\boldsymbol{\xi}_j^{\mathrm{T}} \circ \cdots \circ \boldsymbol{\xi}_j^{\mathrm{T}}}_{m} = [(\xi_{j,1})^m, \cdots, (\xi_{j,N})^m] \in \Re^N$, $\boldsymbol{\phi}(\boldsymbol{\xi}_j) \in \Re^{Nm+1}$, m is the user-defined number of the polynomials, and the coefficients $\boldsymbol{\theta}_j \in \Re^{Nm+1}$ in Eq. (24) can be calculated applying the RLS technique [36] as

$$\boldsymbol{\sigma}_j = \frac{1}{\varsigma}\left(\boldsymbol{\sigma}_{j-1} - \frac{\boldsymbol{\sigma}_{j-1}\boldsymbol{\phi}(\boldsymbol{\xi}_j)\, \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{\xi}_j)\, \boldsymbol{\sigma}_{j-1}}{\varsigma + \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{\xi}_j)\, \boldsymbol{\sigma}_{j-1} \boldsymbol{\phi}(\boldsymbol{\xi}_j)}\right) \tag{26}$$

$$\boldsymbol{\theta}_j = \boldsymbol{\theta}_{j-1} + \frac{\boldsymbol{\sigma}_j \boldsymbol{\phi}(\boldsymbol{\xi}_j)}{\varsigma + \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{\xi}_j)\, \boldsymbol{\sigma}_j \boldsymbol{\phi}(\boldsymbol{\xi}_j)} \left(g(\boldsymbol{\xi}_j) - \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{\xi}_j)\, \boldsymbol{\theta}_{j-1}\right) \tag{27}$$

where $\varsigma > 0$ is a forgetting factor and $\boldsymbol{\sigma}_j \in \Re^{(Nm+1)\times(Nm+1)}$.

Remark 5: The real value of $g(\boldsymbol{\xi}_j)$ can be calculated according to Eq. (18), in which the interaction force error is available as the actual force can be measured by a force sensor while the desired force is given. The estimation of $g(\boldsymbol{\xi}_j)$ is calculated according to Eq. (24). Therefore, the accuracy of the estimator of $g(\boldsymbol{\xi}_j)$ can be evaluated by the estimation error. Besides, since the estimator of $g(\boldsymbol{\xi}_j)$ is designed based on RLS, its effectiveness in the presence of perturbations has been proved by [35], [37], [38].

We re-express the vector $\boldsymbol{\theta}_j$ obtained from Eqs. (26) and (27) as

$$\boldsymbol{\theta}_j = \left[\theta_{0,j}, \theta_{1,j}^1, \theta_{1,j}^2 \cdots, \theta_{1,j}^N, \cdots, \theta_{m,j}^1, \theta_{m,j}^2 \cdots, \theta_{m,j}^N\right]^{\mathrm{T}} \tag{28}$$

and thus the gradient of the estimated index performance function arrives at

$$\boldsymbol{\nabla}\hat{g}(\boldsymbol{\xi}_j) = \left[\sum_{k=1}^{m} k\theta_{k,j}^1 (\xi_{j,1})^{k-1}, \cdots, \sum_{k=1}^{m} k\theta_{k,j}^N (\xi_{j,N})^{k-1}\right]^{\mathrm{T}}. \tag{29}$$

In light of Eqs. (19)-(23), the Adam method is revised based on the estimated gradient $\boldsymbol{\nabla}\hat{g}(\boldsymbol{\xi}_j)$ as

$$\boldsymbol{\varepsilon}_j = \kappa_\varepsilon \boldsymbol{\varepsilon}_{j-1} + (1-\kappa_\varepsilon)\, \boldsymbol{\nabla}\hat{g}(\boldsymbol{\xi}_j) \tag{30}$$

$$\lambda_j = \kappa_\lambda \lambda_{j-1} + (1-\kappa_\lambda)\, \boldsymbol{\nabla}\hat{g}^{\mathrm{T}}(\boldsymbol{\xi}_j)\, \boldsymbol{\nabla}\hat{g}(\boldsymbol{\xi}_j) \tag{31}$$

$$\boldsymbol{\varepsilon}'_j = \frac{\boldsymbol{\varepsilon}_j}{1-\kappa_\varepsilon} \tag{32}$$

$$\lambda'_j = \frac{\lambda_j}{1-\kappa_\lambda} \tag{33}$$

$$\boldsymbol{\xi}_j = \boldsymbol{\xi}_{j-1} - \frac{\kappa_\xi}{\sqrt{\lambda'_j + \eta}} \boldsymbol{\varepsilon}'_j \tag{34}$$

with which $\lim_{j\to\infty} \boldsymbol{\xi}_j = \boldsymbol{\xi}^*$ and $\lim_{j\to\infty} g(\boldsymbol{\xi}_j) = 0$ can be achieved and the interaction force error iteratively tends to zero, which means $\lim_{j\to\infty} f_h(\boldsymbol{\xi}_j) = f_d$ holds.

Remark 6: Adam method has been extensively used to solve various problems, e.g., image processing, sparse problems, etc. Compared with the momentum method, Adam method updates the parameter with a decaying mean over the previous gradients, which contributes to more accurate fine-grained convergence. It also enables the method to straightforwardly correct for the bias arising from initialization to zero. Taking these advantages into consideration, Adam method is adopted to estimate the index performance function in this paper.

3) Updating the Robot's Reference Trajectory: Denote $\Psi(s)$ in the jth iteration with $\boldsymbol{\xi}_j$ as $\Psi_j(s)$ and the desired $\Psi(s)$, with which the specific interaction force can be achieved, as $\Psi_d(s) = \frac{\sum_{i=1}^{N} \xi_i^* \psi_i(s)}{\sum_{i=1}^{N} \psi_i(s)} s(z_g - z_0)$. Then according to Eqs. (10) and (11), we have the following DMPs-based high-level controller for the jth iteration

$$\tau \dot{\varrho}_j = \alpha(\beta(z_g - z_j) - \varrho_j) + \Psi_j(s) \tag{35}$$

$$\tau \dot{z}_j = \varrho_j \tag{36}$$

where $\Psi_j(s)$ is similar to Eq. (13) as

$$\Psi_j(s) = \frac{\sum_{i=1}^{N} \xi_{j,i} \psi_i(s)}{\sum_{i=1}^{N} \psi_i(s)} s(z_g - z_0) \tag{37}$$

z_j is the updated robot's reference trajectory denoted by $x_{r,j}$ in the jth iteration, i.e. $x_{r,j} = z_j$ holds in the jth iteration, and ξ_j is updated as shown in Section II-B-2.

In this paper, $x_{r,j}$ denotes the actual reference trajectory

of the robot generated by our proposed high-level controller in the jth iteration. x_d is the desired reference trajectory of the robot. If the robot can strictly follow x_d with the proposed high-level and low-level controllers without any tracking error, the desired human-robot interaction force can be received. During control process, we expect that x follows $x_{r,j}$ with the designed low-level controller and $x_{r,j}$ iteratively follows x_d with the designed high-level controller to achieve the desired interaction force. Then Theorem 1 is introduced.

Theorem 1: With the gradient descent estimator in Eqs. (24)-(27) and the low-level controller in Eqs. (5)-(7), the DMPs-based high-level controller in Eqs. (30)-(37) enables the human-robot interaction system described in Eq. (3) to achieve a desired interaction force, and the closed-loop system is asymptotically stable.

The proof of Theorem 1 is provided in Appendix.

With $\Psi_j(s)$ in Eq. (37), the updated robot's reference trajectory z_j can be attained by resolving Eqs. (35) and (36). The conclusive procedures of the proposed DMPs-based strategy are provided in Algorithm I.

Remark 7: As described in the introduction section, our proposed method has three major contributions. For contribution 1), it attributes to Eqs. (18), (30)-(34). From Eqs. (18), (30)-(34), it is known that an integral index of force error is utilized to update the parameters of the robot's reference trajectory in each iteration. In this way, the condition of fixed iteration period, which is caused by the point-to-point update, is removed. Contribution 2) corresponds to Eq. (13), from which it is known that the robot's reference trajectory is updated by updating the parameters of the forcing term $\Psi(s)$ that is defined on the phrase variable s rather than the time domain. As a result, the developed scheme has the capability against the time-related uncertainty, which leads to its good robustness. Contribution 3) corresponds to the DMPs in Eqs. (10)-(14). Since the DMPs can encode and parameterize demonstration trajectories of the human user in any pattern, the assumption of known shape of the robot's reference trajectory is removed in this paper.

Algorithm 1: Procedures of the proposed DMPs-based pHRI force control algorithm

Input: Desired interaction force f_d; coefficients for low-level control K_d, K_p; for DMPs α, β, c, τ, z_g, b_i, μ_i; for RLS-based estimator κ_ε , κ_λ and κ_ξ, η;

Output: Reference trajectory $x_{r,j}$;

```
begin
    Fit the DMPs model using the demonstration data of the human user to obtain the initial robot's reference trajectory with parameter vector ξ_0.
    for j = 1 to M do
        Obtain g(ξ_j) in Eq. (18) based on a force sensor;
        Apply the estimator in Eqs. (24)-(27) to obtain the explicit mathematical formulation of ĝ(ξ_j);
        Calculate ∇ĝ(ξ_j) as Eq. (29);
        Update the parameter ξ_j of the robot reference trajectory using the Adam method in Eqs. (30)-(34);
        With the updated parameter ξ_j, calculate Ψ_j(s) as Eq. (37) and update the robot's reference trajectory x_{r,j} = z_j based on DMPs described by Eqs. (35) and (36);
        Use low-level robotic controller f in Eqs. (5)-(7) to make the robot track the reference trajectory x_{r,j}.
    end
end
```

III. Experimental Results

Experiments have been conducted in this section to validate our proposed method, which is desired to generate a specific interaction force by updating the reference trajectory of the robot. An interactive robotic platform H-MAN customized for pHRI is used for the experiments.

As illustrated by Fig. 2A, H-MAN is a planar 2-DoF robot. Its operation area is a 342 mm $\times$ 330 mm square. A human user can interact with H-MAN by holding its handle that is mounted on a slider, and at its end, a 6-DoF force sensor is installed for measuring the pHRI force in real time. A GUI is designed in the experiments to display a reference trajectory to the human user, following which his/her demonstrations are recorded by the robot, and a parameterized initial trajectory of the human user can thus be synthesized off-line using DMPs. Then by asking the human user to track the same trajectory displayed in GUI online, the parameterized robot's reference trajectory is accordingly updated with our proposed scheme and the interaction force is desired to achieve the specific value with iterations.

In the following experimental results, we first show the performance of our proposed DMPs-based trajectory learning scheme. Then a comparison is made between the proposed and existing trajectory learning techniques. Multiple human subjects have been recruited to further validate the effectiveness of our proposed method.

A. Performance of Our Proposed DMPs-based Trajectory Learning Method

In this experiment, the performance of our proposed DMPs-based trajectory learning method is shown in 1D and 2D cases. The desired interaction forces are set to be assistive and resistive to emulate the different uses of our proposed controller. In practice, if the human user cannot complete a certain task due to some specific reasons, e.g., motor dysfunction or other diseases resulting in limb weakness, the robot providing assistive force will assist him/her with the task. On the other side, the robot providing specific resistive force will challenge the human user and contribute to his/her recovery in rehabilitation.

In the experiment, the initial trajectory of the human user is first obtained, based on which the robot will update its reference trajectory for the specific interaction force.

1) Step 1: Obtaining the initial trajectory of the human user: In this step, several demonstrations are conducted by the human user without any robot control. The human user is asked to hold the handle and repetitively follow the reference

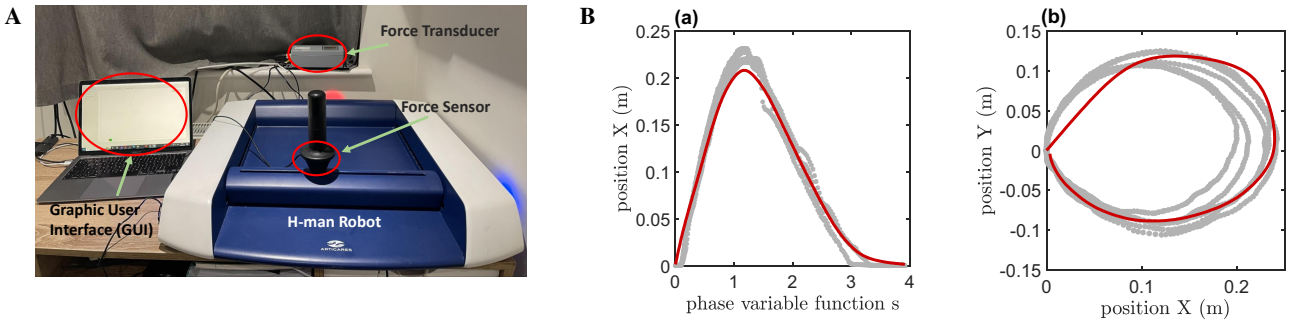

Fig. 2: The experimental setup - H-MAN (A), and demonstrations of the human user (grey lines) and the initial trajectory obtained with DMPs offline (red line) (B): (a) 1D case, (b) 2D case.

trajectory as shown in GUI for five times in a natural and comfortable way. Then the DMP technique is adopted to parameterize the initial trajectory of the human user offline and its parameters are selected based on experience. For 1D case, the DMP is designed with parameters $\tau = 10$, $\alpha = 5.66$, $\beta = 6.55$, and $z_g = 0.2$. The decay factor of the canonical system is designed as $c = 4.94$. For the 2D case, we design z_g as $z_{gx} = 0.2$ and $z_{gy} = \pm 0.1$, where the subscripts x and y denote the X and Y axes, respectively. Other parameters of DMPs are designed to be the same for x and y axes as $\tau = 10$, $\alpha = 3.30$, $\beta = 5.34$, $c = 4.94$.

By parameterizing the recorded five demonstrations of the human user for 1D and 2D cases, the initial trajectory is synthesized and presented in Fig. 2B, where demonstrations are shown in grey colour and the line in red colour denotes the initial trajectory fitting with DMPs. By observing Fig. 2B, it is known that the initial trajectory of the human user learned by DMPs has an irregular shape, which is hard to be learned by the method in [15].

By using DMPs with the parameters given above, parameters of human user's initial trajectory can be achieved for 1D case as $\boldsymbol{\xi}_0 = [8.7, 19, 29.5, 29.7, 9, 5, 13, 60.3, 14.9, 4.27]^{\mathrm{T}}$, where $\xi_{0,i}$ is the ith element of ξ_0, and for the x axis of 2D case as $\boldsymbol{\xi}_{x,0} = [1.39, 2.59, 2, 2.7, 38.9, 0.1, -0.57, -2.7, 15.1, -53]^{\mathrm{T}}$, for the y axis is $\boldsymbol{\xi}_{y,0} = [1.5, 28.3, 3.32, 5.87, -40, 2.2, 15.7, -30, -28, -22.6]^{\mathrm{T}}$. The weight $\boldsymbol{\xi}_0$ for 1D case, and $\boldsymbol{\xi}_{x,0}$ and $\boldsymbol{\xi}_{y,0}$ for 2D case obtained in this step are regarded as the initial values of the parameters of the robot's reference trajectory. They will be updated online in the following step.

2) Step 2: Updating parameters of the robot's reference trajectory: In this step, the human user needs to iteratively track 1D and 2D reference trajectories as shown in GUI that are the same as the ones in step 1, in his/her own way, for 11 iterations. According to the interaction force measured by the sensor in real time, the reference trajectory of the robot will be accordingly updated with our proposed scheme.

a) *For 1D case:* The desired resistive and assistive interaction forces are designed to be position-dependent in the following two cases:

case a: resistive force $f_d = -4x^{0.5}$ N;

case b: assistive force $f_d = 3x^{0.2}$ N.

The parameters of the proposed algorithm are designed as $\kappa_\varepsilon = 0.7$, $\kappa_\lambda = 4.7$, $\eta = 10^{-3}$, $\alpha = 0.74$, $\beta = 0.74$, and $\kappa_\varepsilon = 0.3$, $\kappa_\lambda = 8.5$, $\eta = 10^{-3}$, $\alpha = 0.56$, $\beta = 0.45$ for cases a and b, respectively.

The robot's reference trajectory, the interaction force error as well as the performance index $g(\boldsymbol{\xi}_j)$ in different iterations are presented in Fig. 3(a-c) for case 1 and Fig. 3(d-f) for case 2, respectively. As shown in Fig. 3, with trajectory parameters $\boldsymbol{\xi}_j$ that are updated with our proposed method, the robot's reference trajectory converges with time, which is in line with the theoretical analysis. After 9 iterations, the specific pHRI force f_d is achieved with a small force error as illustrated by Figs. 3b and 3e. Besides, the performance index $g(\boldsymbol{\xi}_j)$ also decreases with iterations and finally converges to zero. As a result, our proposed approach enables the interaction force to achieve the specific value for both resistance and assistance in 1D case.

b) For 2D case: The specific interaction force $\boldsymbol{f_d}$ is designed as:

case c: resistive force with $\|\boldsymbol{f_d}\| = 4$ N;

case d: assistive force with $\|\boldsymbol{f_d}\| = 3$ N.

The parameters of the proposed algorithm are designed to be the same for x and y axes as $\kappa_\varepsilon = 0.11$, $\kappa_\lambda = 4.70$, $\eta = 7 \times 10^{-3}$, $\alpha = 0.95$, $\beta = 0.63$ and $\kappa_\varepsilon = 0.30$, $\kappa_\lambda = 6.68$, $\eta = 7 \times 10^{-3}$, $\alpha = 0.61$, $\beta = 0.23$ for cases a and b, respectively.

The iteratively updated robot's reference trajectories are presented in Fig. 4a for case c and Fig. 4d for case d. By observing Figs. 4a and 4d, it is seen that the reference trajectories of the robot iteratively update towards opposite directions in cases c and d due to the different directions of the specific force $\boldsymbol{f_d}$. Besides, because of the resistance in case d, the human user does not return to the start point in the given time as shown in Fig. 4d.

According to experimental results shown in Figs. 2-4, the effectiveness of the proposed scheme is validated.

B. Comparison with a Traditional Control Strategy

To address the advantage of our developed method, the experiment results with an existing ILC method [12] are presented in this section for comparison.

Similar to our proposed scheme, the existing method focuses on regulating the robot's reference trajectory via a

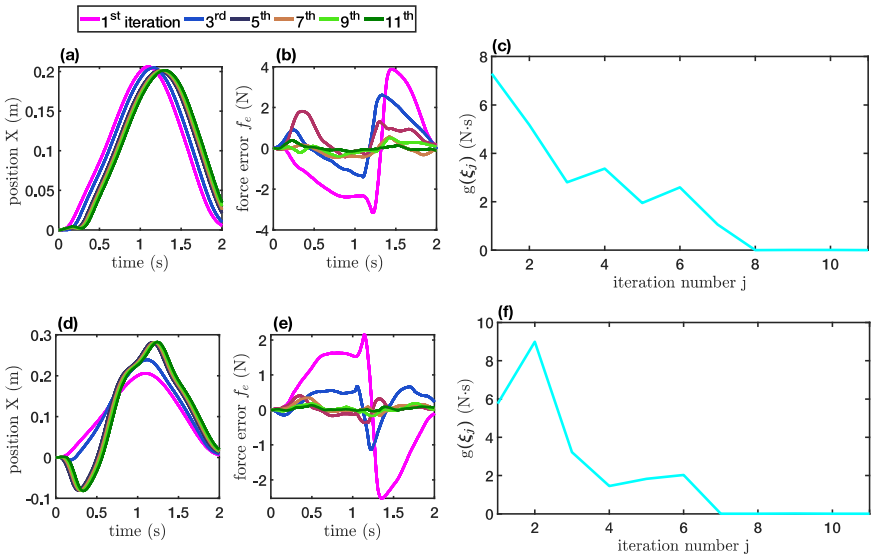

Fig. 3: Performance of our proposed scheme in cases a and b of 1D case: (a) robot's reference trajectory in case a, (b) interaction force error in case a, (c) performance index function $g(\boldsymbol{\xi}_j)$ in case a, (d) reference trajectory in case b, (e) interaction force error in case b, (f) $g(\boldsymbol{\xi}_j)$ in case b.

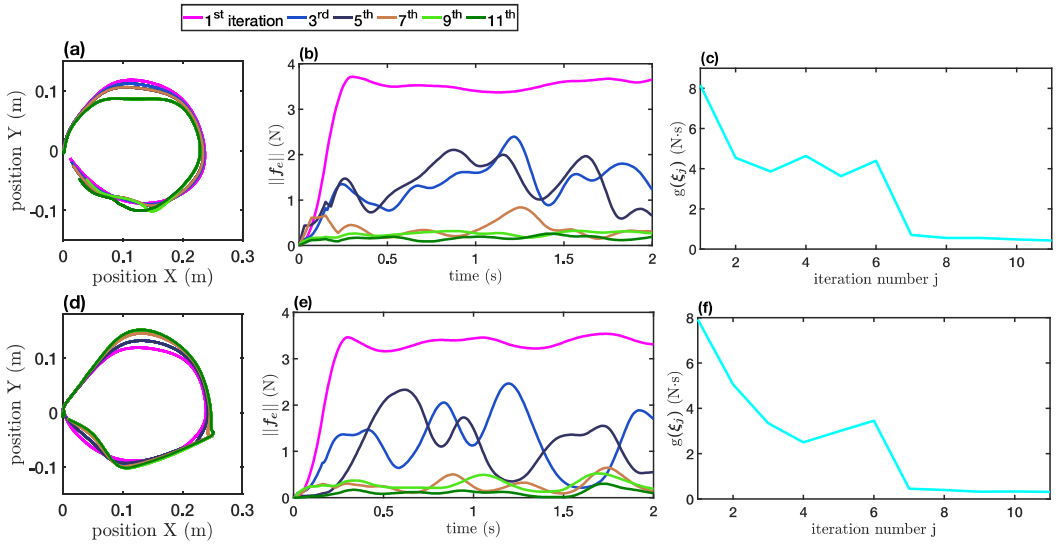

Fig. 4: Performance of our proposed scheme in cases c and d of 2D case: (a) robot's reference trajectory in case c, (b) norm of the interaction force error in case c, (c) performance index function $g(\boldsymbol{\xi}_j)$ in case c, (d) reference trajectory in case d, (e) norm of the interaction force error in case d, (f) $g(\boldsymbol{\xi}_j)$ in case d.

point-to-point approach for a specific interaction force. Its core idea is described as follows [12]

$$\boldsymbol{x}_{\boldsymbol{r},j} = \boldsymbol{x}_{\boldsymbol{r},j-1} + \boldsymbol{\alpha}\boldsymbol{f}_{\boldsymbol{h}} \tag{38}$$

where $\boldsymbol{\alpha}$ is a positive constant denoting the learning speed. In this experiment, we tune the parameter $\boldsymbol{\alpha} = 4.3\boldsymbol{I}_2$ to ensure desired control performance for comparison. In Eq. (38), the robot's reference trajectory is updated based on the information collected from the last iteration in the time domain.

With both existing and proposed strategies, the human user is asked to perform the experiment with hand trembling to explore the performance of our proposed method for the cases discussed in Section III-A. Interaction force errors in the 11th iteration of cases a and b in 1D case and cases c and d in 2D case are shown in Figs. 5A(a-d), respectively. From Fig. 5A, it is clear that the specific interaction force can be achieved with our proposed approach with a smaller force error despite disturbances caused by the trembling hand. Consequently, the proposed approach provides better performance as compared to the existing trajectory learning method in [12], and our method is more robust against the disturbance.

C. Multiple Human Subjects Experiments

To further test whether our proposed method is effective for different individuals, five human subjects (two females and three males whose ages are between 20 and 40) are recruited to perform the experiment with our proposed method and the existing method in [12] for the four cases as discussed before

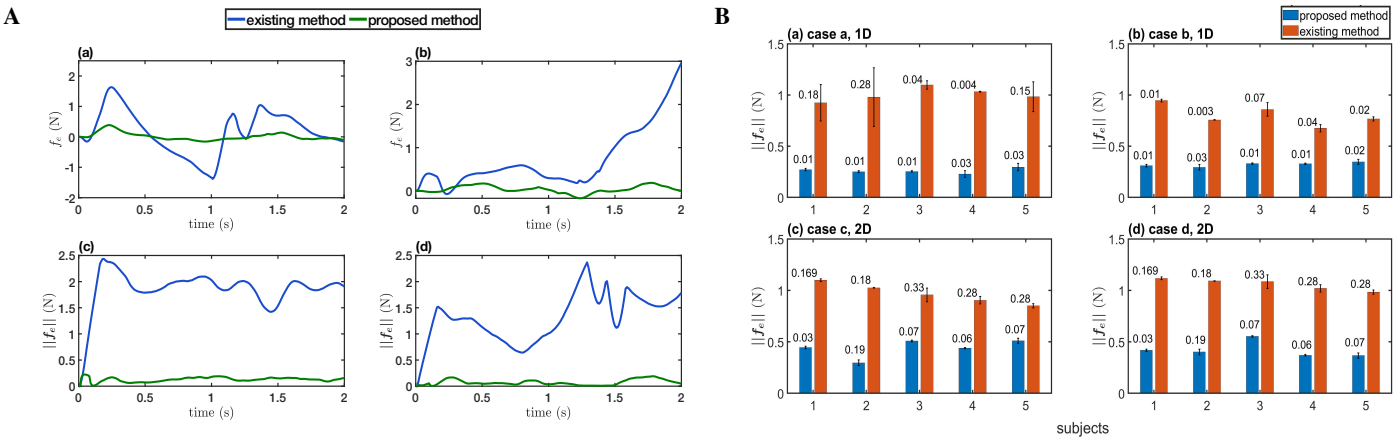

Fig. 5: Tracking error and its norm with the existing and the proposed methods in the last iteration of different cases (A): (a) case a of 1D case, (b) case b of 1D case, (c) case c of 2D case, (d) case d of 2D case, and interaction force errors of five human subjects with the existing and the proposed methods in the last iteration of different cases (B): (a) case a of 1D case, (b) case b of 1D case, (c) case c of 2D case, (d) case d of 2D case.

(cases a and b for 1D case, and cases c and d for 2D case), which is approved by the Sciences and Technology Cross-Schools Research Ethics Committee of Sussex University, with reference number ER/YL557/1. All the subjects need to practice until they are familiar with the experiment. After that, five demonstrations are done offline for step 1, and five pilots (each pilot includes 12 iterations) are performed online for step 2 of each case. Disturbances are introduced to the control signal during the experiment.

The average interaction force errors of five pilots of five human subjects for cases a and b in 1D case and cases c and d in 2D case with both existing and proposed methods are presented in Figs. 5B(a-d), respectively. Fig. 5B shows a considerable difference between the two methods. With the proposed approach, the norm of the interaction force error is reduced to less than $0.1\,\mathrm{N}$ in 1D case and $\approx< 0.2\,\mathrm{N}$ in 2D case. It shows that the proposed approach can overcome noises, and provide a specific force with less error and mostly shorter error bars, i.e., less deviation, for different individuals. Consequently, it caters to different individuals' requirements and can provide users with a specific assistive or resistive interaction force as needed.

IV. Discussions

In practice, $\boldsymbol{f_e}$ in Eq. (8) does not asymptotically converge to zero, as it is shown also in the results. The bounded steady-state error of $\boldsymbol{f_e}$ is caused by the external disturbance or the uncertain dynamics of the robot. If we denote the time-varying disturbance acting on the robot as $\boldsymbol{d} \in \Re^2$ and the state-related uncertainty as $\boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \Re^2$. Then the closed-loop system in Eq. (9) is rewritten as

$$\ddot{\boldsymbol{e}} + \boldsymbol{K_d}\dot{\boldsymbol{e}} + \boldsymbol{K_p}\boldsymbol{e} = \bar{\boldsymbol{M}}^{-1}(\boldsymbol{q})\left(\boldsymbol{f_e} + \boldsymbol{d} + \boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right). \quad (39)$$

Observing Eq. (39), it is known that only the boundedness of $\boldsymbol{e}$ can be guaranteed if $\boldsymbol{d}$ and $\boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are bounded. As a result, $\boldsymbol{f_e}$ will not asymptotically converge to zero.

To solve this problem, we can improve the robotic low-level controller by introducing a disturbance/uncertainty compensation term. To cancel the negative effect of $\boldsymbol{d}$ that satisfies $\|\boldsymbol{d}\| \leq D$, where D is a positive constant, a term $-D\,\mathrm{sgn}(\dot{\boldsymbol{e}})$ [39] can be added to the controller $\boldsymbol{f}$, where sgn is a signum function that can be replaced with a tanh-function [40] or a saturation function [41]. Compared with signum function, tanh and saturation functions enable smoother control performance without chattering. However, different from signum function, which endows the closed-loop system with asymptotical stability, tanh and saturation functions can only guarantee the stability of the system with a steady-state error, which can be adjusted by changing control parameters. Besides, a disturbance observer can be designed to estimate and compensate for the external disturbance [42].

To compensate for the undesired effect of the uncertainty, a neural network adaptive term [43] or a fuzzy adaptive term [44] can be introduced to $\boldsymbol{f}$ to approximate and compensate for $\boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}})$.

V. Conclusions

In this paper, a DMPs-based trajectory learning method is developed for interaction force control. The proposed algorithm aims at achieving a specific interaction force between human and robot. The merit of DMPs, namely capturing the characteristics of the system by parametrization, is integrated into our strategy so that the robot iteratively updates its reference trajectory by updating parameters of DMPs without any restriction of iterative time duration and against time-related uncertainty. To do so, a performance index function is designed, whose gradient is calculated by an RLS-estimator, and Adam method is adopted to search the optimal parameters of DMPs that can generate a desired reference trajectory of the robot. In this way, the actual interaction force can converge to a specific value with iterations. The experiment results with multiple subjects are conducted so that the effectiveness of the proposed scheme is validated.

APPENDIX

Combing Eqs. (35) and (36) in the revised paper, one gets

$$\tau^2 \ddot{z}_j + \alpha\tau \dot{z}_j - \alpha\beta \left(z_g - z_j\right) - \Psi_j\left(s\right) = 0. \tag{40}$$

With the desired forcing term $\Psi_d\left(s\right) = \frac{\sum_{i=1}^{N} \xi_i^* \psi_i(s)}{\sum_{i=1}^{N} \psi_i(s)} s\left(z_g - z_0\right)$, x_d for the specific interaction force satisfies the following dynamics

$$\tau^2 \ddot{x}_d + \alpha\tau \dot{x}_d - \alpha\beta \left(z_g - x_d\right) - \Psi_d\left(s\right) = 0. \tag{41}$$

Combing Eqs. (40) and (41), we have

$$\tau^2 \ddot{z}_{e,j} + \alpha\tau \dot{z}_{e,j} + \alpha\beta z_{e,j} = \Psi_{e,j}\left(s\right) \tag{42}$$

where $z_{e,j} = z_j - x_d$ and $\Psi_{e,j}\left(s\right) = \Psi_j\left(s\right) - \Psi_d\left(s\right)$.

Design a Lyapunov function of the system for the jth iteration as follows

$$V_j = \frac{1}{2}\alpha\beta z_{e,j}^2 + \frac{1}{2}\tau^2 \dot{z}_{e,j}^2. \tag{43}$$

Taking time derivative of Eq. (43) yields

$$\begin{aligned} \dot{V}_j &= \alpha\beta z_{e,j}\dot{z}_{e,j} + \dot{z}_{e,j}\left(\Psi_{e,j}\left(s\right) - \alpha\tau\dot{z}_{e,j} - \alpha\beta z_{e,j}\right) \\ &= -\alpha\tau\dot{z}_{e,j}^2 + \dot{z}_{e,j}\Psi_{e,j}\left(s\right). \end{aligned} \tag{44}$$

In view of the convergence property of the Adam method and the attenuation of s as reflected by Eq. (12), it is known that $\Psi_{e,j}\left(s\right)$ is bounded and $\Psi_{e,j}\left(s\right) \to 0$ holds when $j \to \infty$. Regarding $\Psi_{e,j}\left(s\right)$ as a virtual control input, from Eq. (44), one gets $\dot{V}_j = -\alpha\tau\dot{z}_{e,j}^2 \leq 0$ for $j \to \infty$. Define a set Ω as $\Omega = \left\{ \left(z_{e,j}, \dot{z}_{e,j}\right) \middle| \dot{V}_j = 0 \right\}$. If $\dot{V}_j = 0$, $\dot{z}_{e,j} = 0$ is satisfied. As a result, $\dot{z}_{e,j} = 0$ is necessary and the set Ω can be rewritten as $\Omega = \left\{ \left(z_{e,j}, \dot{z}_{e,j}\right) \middle| \dot{z}_{e,j} = 0 \right\}$. When $\dot{z}_{e,j} = 0$ holds, $\ddot{z}_{e,j} = 0$ holds, and $z_{e,j} = 0$ can be obtained for $j \to \infty$ from Eq. (42). As a result, using LaSalle's invariance theorem [45], it is known that the system is input-to-state stable and $\left(z_{e,j}, \dot{z}_{e,j}\right)$ will eventually converge to $(0, 0)$, i.e., $\lim_{j\to\infty} z_{e,j} = 0$ and $\lim_{j\to\infty} \dot{z}_{e,j} = 0$. It means that the reference trajectory of the robot can iteratively converge to the desired value in the sense of $x_d = \lim_{j\to\infty} z_j = \lim_{j\to\infty} x_{r,j}$, following which the specific interaction force can be finally generated. Therefore, the system stability with the proposed DMPs-based high-level controller is proved. To notice, the above proof process is inspired by [46], in which more details of the proof can be found.

REFERENCES

[1] W. He, C. Xue, X. Yu, Z. Li, and C. Yang, "Admittance-based controller design for physical human–robot interaction in the constrained task space," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1937–1949, 2020.

[2] Z. Li, B. Huang, Z. Ye, M. Deng, and C. Yang, "Physical human–robot interaction of a robotic exoskeleton by admittance control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9614–9624, 2018.

[3] L. Han, W. Xu, P. Kang, and H. Yuan, "Unified neural adaptive control for multiple humancrobotcenvironment interactions," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1166–1175, 2021.

[4] C. Yang, X. Liu, J. Zhong, and A. Cangelosi, "Human robot collaborative intelligence: Theory and applications," *Interaction Studies*, vol. 20, no. 1, pp. 1–3, 2019.

[5] Y. Zhuang, S. Yao, C. Ma, and R. Song, "Admittance control based on emg-driven musculoskeletal model improves the humancrobot synchronization," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1211–1218, 2019.

[6] T. Anwar and A. Al Juamily, "Adaptive trajectory control to achieve smooth interaction force in robotic rehabilitation device," *Procedia Computer Science*, vol. 42, pp. 160–167, 2014.

[7] A. Zhu, Y. Tu, W. Zheng, H. Shen, and X. Zhang, "Adaptive control of man-machine interaction force for lower limb exoskeleton rehabilitation robot," in *2018 IEEE International Conference on Information and Automation (ICIA)*, 2018, pp. 740–743.

[8] K. Anam and A. A. Al-Jumaily, "Active exoskeleton control systems: State of the art," *Procedia Engineering*, vol. 41, pp. 988–994, 2012, international Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

[9] H. Yu, S. Huang, G. Chen, Y. Pan, and Z. Guo, "Human–robot interaction control of rehabilitation robots with series elastic actuators," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1089–1100, 2015.

[10] A. Gams, T. Petric, B. Nemec, and A. Ude, "Learning and adaptation of periodic motion primitives based on force feedback and human coaching interaction," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 166–171.

[11] W. He, T. Meng, X. He, and C. Sun, "Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances," *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1524–1535, 2019.

[12] J. Zhang, C. C. Cheah, and S. H. Collins, "Experimental comparison of torque control methods on an ankle exoskeleton during human walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5584–5589.

[13] Y. Li, X. Zhou, J. Zhong, and X. Li, "Robotic impedance learning for robot-assisted physical training," *Frontiers in Robotics and AI*, vol. 6, pp. 1–13, 2019.

[14] L. Yang, Y. Li, D. Huang, J. Xia, and X. Zhou, "Spatial iterative learning control for robotic path learning," *IEEE Transactions on Cybernetics*, pp. 1–10, 2022.

[15] X. Xing, K. Maqsood, D. Huang, C. Yang, and Y. Li, "Iterative learning-based robotic controller with prescribed human-robot interaction force," *IEEE Transactions on Automation Science and Engineering*, pp. 1–14, 2021.

[16] F. Stulp and J. Buchli and A. Ellmer and M. Mistry and E. Theodorou and S. Schaal, "Reinforcement learning of impedance control in stochastic force fields," in *2011 IEEE International Conference on Development and Learning (ICDL)*, vol. 2. IEEE, 2011, pp. 1–6.

[17] L. Peternel and T. Petrič and J. Babič, "Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1497–1502.

[18] C. Yang and C. Zeng and C. Fang and W. He and Z. Li, "A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1193–1203, 2018.

[19] L. Han and H. Yuan and W. Xu and Y. Huang, "Modified dynamic movement primitives: robot trajectory planning and force control under curved surface constraints," *IEEE transactions on cybernetics*, 2022.

[20] J. Li, Z. Li, X. Li, Y. Feng, Y. Hu, and B. Xu, "Skill learning strategy based on dynamic motion primitives for human–robot cooperative manipulation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 105–117, 2020.

[21] Y. Yuan, Z. Li, T. Zhao, and D. Gan, "Dmp-based motion generation for a walking exoskeleton robot using reinforcement learning," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3830–3839, 2019.

[22] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[23] B. Nemec, F. J. Abu-Dakka, B. Ridge, A. Ude, J. A. Jørgensen, T. R. Savarimuthu, J. Jouffroy, H. G. Petersen, and N. Krüger, "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile," in *2013 16th International Conference on Advanced Robotics (ICAR)*. IEEE, 2013, pp. 1–7.

[24] B. Nemec, A. Gams, M. Deniša, and A. Ude, "Human-robot cooperation through force adaptation using dynamic motion primitives and iterative learning," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE, 2014, pp. 1439–1444.

[25] Y. Yuan, Z. Li, T. Zhao, and D. Gan, "DMP-based motion generation for a walking exoskeleton robot using reinforcement learning," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3830–3839, 2019.

[26] T. Davchev, K. S. Luck, M. Burke, F. Meier, S. Schaal, and S. Ramamoorthy, "Residual Learning from Demonstration: Adapting DMPs for Contact-rich Manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4488–4495, 2022.

[27] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Control, planning, learning, and imitation with dynamic movement primitives," in *Workshop on Bilateral Paradigms on Humans and Humanoids: IEEE International Conference on Intelligent Robots and Systems (IROS 2003)*, 2003, pp. 1–21.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] Z. Bien and J.-X. Xu, *Iterative learning control: analysis, design, integration and applications*. Springer Science & Business Media, 2012.

[30] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, vol. 17, pp. 1–19, 2004.

[31] C. Zeng, H. Su, Y. Li, J. Guo, and C. Yang, "An approach for robotic leaning inspired by biomimetic adaptive control," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1479–1488, 2021.

[32] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 2, 2002, pp. 1398–1403.

[33] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[34] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *arXiv preprint arXiv:2102.03861*, 2021.

[35] A. G. Feldman, "Once more on the equilibrium-point hypothesis (λ model) for motor control," *Journal of motor behavior*, vol. 18, no. 1, pp. 17–54, 1986.

[36] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Transactions on signal processing*, vol. 52, no. 8, pp. 2275–2285, 2004.

[37] R. M. Johnston and J. C. R. Johnson and R. R. Bitmead and B. D. Anderson, "Exponential convergence of recursive least squares with exponential forgetting factor," *Systems & Control Letters*, vol. 2, no. 2, pp. 77–82, 1982.

[38] S. Ljung and L. Ljung, "Error propagation properties of recursive least-squares adaptation algorithms," *Automatica*, vol. 21, no. 2, pp. 157–167, 1985.

[39] X. Yin and L. Pan, "Direct adaptive robust tracking control for 6 DOF industrial robot with enhanced accuracy," *ISA transactions*, vol. 72, pp. 178–184, 2018.

[40] G. Song, L. Cai, Y. Wang, and R. W. Longman, "A sliding-mode based smooth adaptive robust controller for friction compensation," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 8, no. 8, pp. 725–739, 1998.

[41] M. G. Ortega, M. Vargas, C. Vivas, and F. R. Rubio, "Robustness improvement of a nonlinear H_∞ controller for robot manipulators via saturation functions," *Journal of Robotic Systems*, vol. 22, no. 8, pp. 421–437, 2005.

[42] W.-H. Chen, D. J. Ballance, P. J. Gawthrop, J. OReilly, "A nonlinear disturbance observer for robotic manipulators," *IEEE Transactions on industrial Electronics*, vol. 47, no. 4, pp. 932–938, 2000.

[43] H. D. Patiño, R. Carelli, and B. R. Kuchen, "Neural networks for advanced control of robot manipulators," *IEEE Transactions on Neural networks*, vol. 13, no. 2, pp. 343–354, 2002.

[44] C. Ham, Z. Qu, and R. Johnson, "Robust fuzzy control for robot manipulators," *IEE Proceedings-Control Theory and Applications*, vol. 147, no. 2, pp. 212–216, 2000.

[45] F. Mazenc and D. Nesic, "Strong Lyapunov functions for systems satisfying the conditions of La Salle," *IEEE transactions on automatic control*, vol. 49, no. 6, pp. 1026–1030, 2004.

[46] D. H. Zhai, Z. Xia, H. Wu, and Y. Xia, "A Motion Planning Method for Robots Based on DMPs and Modified Obstacle-Avoiding Algorithm," *IEEE Transactions on Automation Science and Engineering*, 2022.

Xueyan Xing received the B.S. degree in automation from Northeastern University, Shenyang, China, in 2015, the M.S. degree in control engineering from the Harbin Institute of Technology, Harbin, China, in 2017, and the Ph.D. degree in control theory and control engineering from Beihang University, Beijing, China, in 2020. She was a research fellow with the Department of Engineering and Design, University of Sussex, Brighton, U.K from 2020 to 2022. Currently, she is a research fellow with the Continental-NTU Corporate Lab, Nanyang Technological University, Singapore. Her current research interests include physical human-robot interaction, distributed control and optimization, distributed Nash equilibrium seeking, and vibration control.

Kamran Maqsood is a control systems engineer at Wienerberger UK and a research associate at the University of Sussex, UK. He obtained his BSc in electronics and MSc in advanced electronics and control systems in 2013 and 2016, respectively from University College of Engineering and Technology, Bahawalpur, Pakistan. In 2023, he completed his PhD on control systems for robotics from the University of Sussex. His research interests encompass human-robot interaction, robot control, and control theory and applications.

Chao Zeng received the Ph.D. degree in pattern recognition and intelligent systems from the South China University of Technology, Guangzhou, China, in 2019. He visited the Department of Informatics, University of Hamburg (UHH), Hamburg, Germany, from 2018 to 2019. He has been a research associate with UHH since July 2020. His research interests include robot learning and control, and physical human-robot interaction.

Chenguang Yang (M'10-SM'16) received the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010, and postdoctoral training in human robotics from the Imperial College London, London, U.K. He was awarded UK EPSRC UKRI Innovation Fellowship and individual EU Marie Curie International Incoming Fellowship. As the lead author, he won the IEEE Transactions on Robotics Best Paper Award (2012) and IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award (2022). He is the Corresponding Co-Chair of IEEE Technical Committee on Collaborative Automation for Flexible Manufacturing, a Fellow of Institute of Engineering and Technology (IET), a Fellow of Institution of Mechanical Engineers (IMechE), and a Fellow of British Computer Society (BCS). His research interest lies in human robot interaction and intelligent system design.

Shuai Yuan received the B.Sc., M.Sc. degree in Mechanical Science and Engineering from Harbin Institute of Technology, Huazhong University of Science and Technology, China, in 2011 and 2014 respectively, and the Ph.D. degree (with cum laude) in Systems and Control in 2018 from Delft University of Technology, Netherlands. He joined Harbin Institute of Technology in September 2018 and is currently an associate professor in the School of Astronautics. His research interests include learning-based optimization, adaptive control, and applications in aerospace and robotics. He serves as an associate editor of International Journal of Adaptive Control and Signal Processing.

Yanan Li (M14-SM21) received the B.Eng. and M.Eng. degrees in automatic control from the Harbin Institute of Technology, China, in 2006 and 2008, respectively, and the Ph.D. degree in robotics from the National University of Singapore, Singapore, in 2013. 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. He is currently a Senior Lecturer in control engineering with the Department of Engineering and Design, University of Sussex, Brighton, U.K. His general research interests include human-robot interaction, robot control, and control theory and applications.