UNIVERSITY OF LEEDS

This is a repository copy of *Uncertainty Compensated High-Order Adaptive Iteration Learning Control for Robot-Assisted Upper Limb Rehabilitation*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/208954/

Version: Accepted Version

Article:
Ai, Q. orcid.org/0000-0003-4283-2289, Liu, Z. orcid.org/0000-0002-2261-2812, Meng, W. orcid.org/0000-0003-0209-8753 et al. (2 more authors) (2024) Uncertainty Compensated High-Order Adaptive Iteration Learning Control for Robot-Assisted Upper Limb Rehabilitation. IEEE Transactions on Automation Science and Engineering, 21 (4). 7004 -7015. ISSN 1545-5955

https://doi.org/10.1109/tase.2023.3335401

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Uncertainty Compensated High-Order Adaptive Iteration Learning Control for Robot-Assisted Upper Limb Rehabilitation

Qingsong Ai, *Member, IEEE*, Zemin Liu, Wei Meng, *Member, IEEE*, Quan Liu, *Member, IEEE*, and Sheng Q. Xie, *Senior Member, IEEE*

***Abstract*— Upper limb rehabilitation robot can assist stroke patients to complete daily activities to promote the recovery of upper-limb motor functions. However, the robot uncertainty and the patient's unconscious disturbance impose great difficulties on the high-performance trajectory tracking of the rehabilitation robot. In this paper, an uncertainty compensated high-order adaptive iterative learning controller (UCHAILC) is proposed to reduce the impact of uncertainty from inside and outside of the robot during the rehabilitation process. The nonlinear system is converted into a dynamic linearization model with uncertainty compensation, and the optimization criterion method is adopted to estimate the pseudo-partial derivative (PPD) parameters and the uncertainty respectively, then the previous iterations are used to update the current parameters through a high-order learning scheme. The convergence of UCHAILC is theoretically proved. Simulation and control experiments on a rehabilitation robot are given to validate the effectiveness of the proposed method, which is significant to improve the training security and physiotherapy effect of robot-assisted rehabilitation.**

***Note to Practitioners*—This paper was motivated by the need to assist stroke patients to restore motor function for executing daily activities. The inherent difficulties lie in reducing the tracking errors of rehabilitation robots caused by uncertainty and involuntary disturbance from patients to avoid secondary injury. The proposed UCHAILC can transform the complex nonlinear system into a dynamic linear model with uncertainty compensation, then the PPD parameters and uncertainty are estimated through high-order learning law. Theoretical analysis, simulation, and experiments verified the feasibility of the method. Furthermore, the proposed controller is not limited to the dynamic model and hardware driving mode of the robot system, which can be easily transplanted to other nonlinear control systems with uncertainties.**

***Index Terms*— Upper limb rehabilitation, uncertainty compensation, model-free control, iterative learning control.**

Manuscript received 22 June 2023; revised 4 September 2023; accepted 18 November 2023. This article was recommended for publication by Associate Editor S.-L. Chen and Editor L. Zhang upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 52275029 and Grant 52075398 and in part by the Key Research and Development Program of Hubei Province under Grant 2022BAA066. *(Corresponding author: Wei Meng.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Human Participants Ethics Committee of Wuhan University of Technology.

Qingsong Ai is with the School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China, and also with the School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China (e-mail: qingsongai@whut.edu.cn).

Zemin Liu, Wei Meng, and Quan Liu are with the School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China (e-mail: zeminliu@whut.edu.cn; weimeng@whut.edu.cn; quanliu@whut.edu.cn).

Sheng Q. Xie is with the School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT Leeds, U.K. (e-mail: s.q.xie@leeds.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2023.3335401.

Digital Object Identifier 10.1109/TASE.2023.3335401

I. Introduction

THE number of stroke patients is increasing as countries around the world gradually enter aging societies, and almost 85% of patients with stroke suffer from upper limb motor function disorders [1]. Repeated rehabilitation training can promote the recovery of injured motor neural centers. Traditional treatment method relies on long-term one-on-one repetitive assistance training from therapists [2], which is time-consuming and labor-intensive. Experiments demonstrate that robot-assisted rehabilitation is effective to encourage motor function recovery for stroke patients, which can provide high-intensity, repetitive, and task-specific treatments [3]. Due to the lack of upper limb motor ability, many stroke patients are unable to engage in common activities of daily living (ADL) by themselves, such as eating and drinking. It is significant to cooperate with ADL in rehabilitation training to promote the recovery of arm motor function [4]. Nevertheless, as a multi-variable, nonlinear, and time-varying system, the motion control of rehabilitation robots with uncertainty from inside and outside is challenging. The uncertainty factors of robot-assisted rehabilitation include model uncertainty [5], end-effector load and friction, etc. In addition, as the robot contacts with patients directly, the environmental uncertainty [6] caused by time-varying pathological statuses such as muscle tone or muscle cramps of the affected limb will further make it challenging for controller design. Besides, the trajectories of ADL are usually irregular and cannot be represented by common functions, which also imposes great difficulties for precise ADL trajectory tracking.

Since the appearance of rehabilitation robots, their motion control has attracted extensive attention to prevent second injuries due to tracking errors. There are many dynamic uncertainty factors leading to the instability and insecurity of the rehabilitation robot system. Currently, a majority of

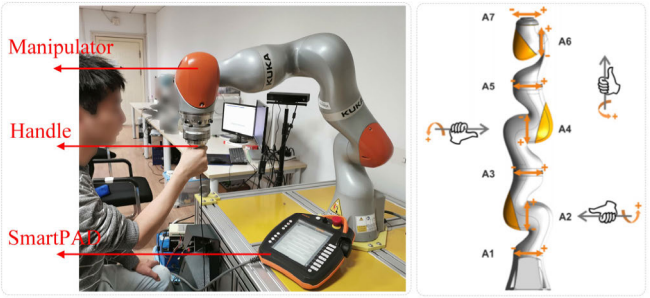
1545-5955 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

rehabilitation robots have adopted model-based control methods to guarantee trajectory tracking accuracy. Mojtaba et al. [7] designed a nonlinear model reference adaptive impedance controller for MIT-MANUS with uncertainties. David et al. [8] designed a robust motion control architecture based on cascade proportional-integral (PI) and sliding mode controller (SMC), which could compensate for the matched disturbances while reducing tracking error. Wu et al. [9] developed a neural-fuzzy adaptive controller (NFAC) strategy based on a radial basis function network to track predefined trajectories with parametric uncertainties and environmental disturbances. However, it is difficult for the complex nonlinear system to obtain accurate dynamic parameters. As the model parameters are time-varying, and the behavior of muscle tissues and tendons are highly nonlinear during rehabilitation training, which leads to the degradation performance of the model-based controller.

Data-driven control (DDC) is a viable method applicable to nonlinear control systems. The iterative learning control (ILC) [10] is a typical DDC method to improve the tracking performance by updating the current iteration parameters with the control knowledge from the previous iterations. Zhang et al. [11]designed an iterative trajectory learning scheme to compensate for unknown time-varying periodic disturbances. Zhu et al. [12] proposed a double iterative compensation learning controller to adjust the parameters to satisfy the patient's condition. Maqsood [13] et al. developed an iterative learning-based path control method to update the robot trajectory according to human movement. Nevertheless, these methods required prior information about the human-robot interaction model which are often unavailable during the rehabilitation process, so they cannot achieve satisfactory tracking accuracy for nonlinear system with uncertainty. Model-free adaptive iterative learning controller (MFAILC) is an extended DDC method proposed by Hou [14], which established an equivalent dynamic linearization model to substitute the original system, only using the input and output (I/O) data to estimate the pseudo-partial derivative (PPD) by repetitive operations. Zhao et al. [15] used partial-form dynamic linearization based MFAILC for a noncircular turning tool. Bu et al. [16] proposed a distributed MFAILC method for an unknown nonlinear multi-agent system to perform consensus tracking. Wang et al. [17] proposed a distributed disturbance compensation based MFAILC algorithm to achieve the consensus tracking of nonlinear multi-agent systems with unknown disturbance. Esmaeili et al. [18] designed model-free adaptive iterative learning integral terminal sliding mode controller to address the tracking issue of the multi-degree-of-freedom robot under external perturbations. We have also established a robust iterative feedback tuning control technique for repetitive training control of a compliant ankle rehabilitation robot [19], and a high-order pseudo-partial derivative based MFAILC has been proposed to achieve high-performance repetitive control of pneumatic artificial muscle (PAM) actuator [20]. However, MFAILC only relies on I/O data to build a dynamic linearization model, which requires a large number of iterations bringing about a slow convergence rate. The excessive consumption of iteration and time of ILC makes it impractical for robot-assisted rehabilitation.



(a) (b)

Fig. 1. (a) Robot-assisted rehabilitation scene; (b) 7- DOF manipulator model.

In this paper, we propose an uncertainty compensated high-order adaptive iteration learning control (UCHAILC) scheme for a seven degree-of-freedom (7)-DOF) rehabilitation robot. To satisfy the requirement of accurate tracking and fast convergence of irregular human-like trajectories, we design the uncertainty compensated controller to improve system robustness and adopt the high-order learning law [21] to utilize more knowledge of past iterations to boost the convergence speed of ILC. The main contributions of this work include: 1) A general dynamic linearization framework with uncertainty compensation is designed to represent the nonlinear robot system only utilizing I/O data, which does not refer to any robot model information and is not affected by the dynamic characteristics of the system; 2) the uncertainty and PPD parameters are estimated by the optimized criterion function techniques respectively to avoid the coupling of the estimation between uncertainty and PPD parameters; 3) a high-order learning law is utilized to redesign the PPD and control inputs, which employs more parameters of the previous iterations to calculate control parameters of current iteration to improve the convergence speed and achieve a better tracking performance.

The rest of this paper is arranged as follows: Section II introduces the human-like trajectory generation method for rehabilitation robots; the principle deduction of proposed UCHAILC and its convergence proof is presented in Section III; Simulation and tracking control experiment results of circular and ADL trajectories on rehabilitation robot are provided in Section IV, and Section V gives the discussion and conclusion.

II. Human-Like Trajectory Generation for Rehabilitation Robot

To assist patients in completing ADL tasks, we adopt the Gaussian mixture model (GMM) to generate a human-like trajectory for the rehabilitation robot. GMM can model the joint probability of input and output variables, and learn the probabilistic features from multi-demonstrations effectively.

A. Upper Limb Rehabilitation Robot

The upper limb rehabilitation robot system, as shown in Fig.1(a), is composed of the KUKA LBR iiwa R700

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

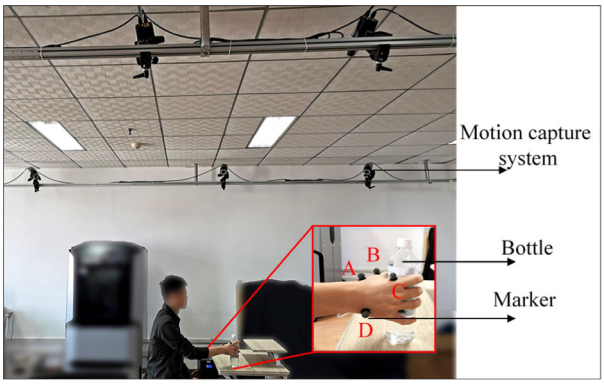


Fig. 2. Data collection scene setting.

manipulator and a handle used to connect with humans. This manipulator is a lightweight collaborative robot suitable for human-robot interaction tasks. In Fig. 1(b), this manipulator has seven DOFs and a flexible workspace, similar to a human arm, which can reach the same position/orientation in different configurations [22]. The *Sunrise.Servoing* package is utilized to implement real-time joint position control mode, and we use the KUKA Sunrise Toolbox [23] developed by Safeea to realize the online control signal input and real-time data acquisition (30Hz) of the robotic arm with MATLAB, which provide a great convenience in applying the control algorithm to the manipulator. The open-source code can be found at https://github.com/Modi1987/KST-Kuka-Sunrise-Toolbox. The robotic arm can assist patients to conduct various rehabilitation tasks including common circular trajectory tracking and irregular ADL tasks.

B. Data Collection and Preprocessing

Irregular ADL trajectories cannot be expressed by common functions, so we acquire ADL trajectories from healthy people to generate end-effector human-like trajectories for the robotic manipulator. In Fig. 2, the motion capture system (Qualisys, Sweden) is utilized to record the action sequence of healthy people's hands while drinking. Four markers (A, B, C, D) are set on the hand to record the position and orientation of the endpoint, as shown in the enlarged image. This trial has been approved by Human Participants Ethics Committee from Wuhan University of Technology, and written informed consent was obtained from each participant.

Next, the spatial position and direction of the hand can be calculated through coordinate transformation. In general, a rotation matrix is used to represent the position and orientation [24].

$$T = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\gamma c\beta & c\gamma s\beta s\alpha - s\gamma c\alpha & c\gamma s\beta c\alpha + s\gamma s\alpha & p_x \\ s\gamma c\beta & s\gamma s\beta s\alpha + c\gamma c\alpha & s\gamma s\beta c\alpha - c\gamma s\alpha & p_y \\ -s\beta & c\beta s\alpha & c\beta c\alpha & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where vector a, b, c represent orientation and vector p denotes position, α, β, γ are orientation angles. The position coordinates a, b, c are determined directly from the coordinates of the marked points, and orientation angles α, β, γ are calculated using the Euler angle formula.

The ADL demonstration path can be represented as a set of positions and directions $D = \{\{\xi^{n,m}\}_{n=1}^{N}\}_{m=1}^{M}$, N is the time length and M is the number of teaching trajectories, $\xi^{n,m} = [\tau^{n,m}\tau^{\dot{n},m}]$, $\tau^{n,m}$ denote the pose at the $n-$th time-step from the m-th demonstration, $\tau^{\dot{n},m}$ is the derivative of $\tau^{n,m}$.

$$\tau^{n,m} = \{\alpha^{n,m}, \beta^{n,m}, \gamma^{n,m}, p_x^{n,m}, p_y^{dn,m}, p_z^{dn,m}\} \quad (2)$$

C. Human-like Trajectory Generation

Then GMM [25] is adopted to fit and generate the end-effector human-like trajectories of the manipulator from the teaching trajectories in the above section. GMM is constructed based on the weighted sum of Gaussian component density, and GMM can be expressed as

$$p(t, \xi) \sim \sum_{z=1}^{Z} \pi_z \mathcal{N}(\mu_z, \Sigma_z) \quad (3)$$

where Z is the number of Gaussian components and π_z denotes the prior probability, $\sum_{z=1}^{Z} \pi_z = 1$, the mean and covariance of each Gaussian component are $\mu_z = \begin{bmatrix} \mu_{t,z} \\ \mu_{\xi,z} \end{bmatrix}$ and $\Sigma_z = \begin{bmatrix} \Sigma_{tt,z} & \Sigma_{t\xi,z} \\ \Sigma_{\xi t,z} & \Sigma_{\xi\xi,z} \end{bmatrix}$.

Gaussian mixture regression (GMR) is used to predict the conditional probability distribution:

$$p(\xi|t) = \sum_{z=1}^{Z} h_z(t) \mathcal{N}(\overline{\mu_z}(t), \overline{\Sigma_z}) \quad (4a)$$

with

$$h_z(t) = \frac{\pi_z \mathcal{N}(t|\mu_{t,z}, \Sigma_{tt,z})}{\sum_{i=1}^{Z} \pi_i \mathcal{N}(t|\mu_{t,i}, \Sigma_{tt,i})} \quad (4b)$$

$$\overline{\mu_z}(t) = \mu_{\xi,z} + \Sigma_{\xi t,z} \Sigma_{tt,z}^{-1}(t - \mu_{t,z}) \quad (4c)$$

$$\overline{\Sigma_z} = \Sigma_{\xi\xi,z} - \Sigma_{\xi t,z} \Sigma_{tt,z}^{-1} \Sigma_{t\xi,z} \quad (4d)$$

Due to the properties of multivariate Gaussian distributions, we can estimate $\mathbb{E}(\xi|t)$ and $\mathbb{D}(\xi|t)$:

$$\hat{\mu}_t = \mathbb{E}(\xi|t) = \sum_{z=1}^{Z} h_z(t) \overline{\mu_z}(t) \quad (5)$$

$$\hat{\Sigma}_t = \mathbb{D}(\xi|t) = \mathbb{E}(\xi\xi^T|t) - \mathbb{E}(\xi|t)\mathbb{E}^T(\xi|t) = \sum_{z=1}^{Z} h_z(t)(\overline{\mu_z}(t)\overline{\mu_z^T}(t) + \overline{\Sigma_z}) - \hat{\mu}_z\hat{\mu}_z^T \quad (6)$$

Furthermore, (4a) can be approximated as

$$p(\xi|t) \approx \mathcal{N}(\hat{\mu}_t, \hat{\Sigma}_t) \quad (7)$$

In this paper, we acquire five sets of demonstration trajectories of healthy people performing drinking tasks within 9 seconds and calculate the Cartesian positions and orientation angles as the teaching trajectories, which are indicated by the green curves in Fig. 3. Through GMM and GMR, the end-point trajectories of human are generated as the red curve.

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

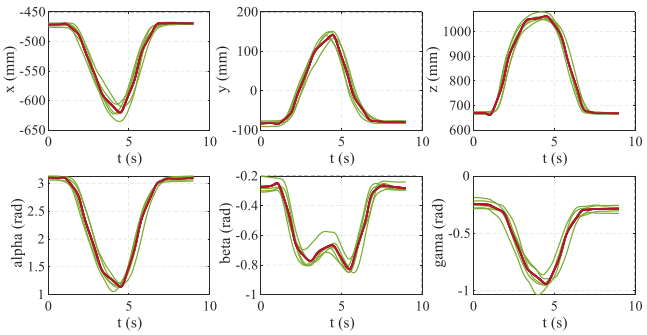


Fig. 3. End-point trajectories of humans in Cartesian space.

III. Uncertainty Compensated High-Order Adaptive Iteration Learning Control

A. Problem Formulation

Due to the uncertainty factors of human-robot interaction system during rehabilitation, we regard it as a discrete-time nonlinear system with uncertainty described as follows:

$$y_k(t+1) = f(y_k(t), \cdots, y_k(t-n_y), u_k(t), \cdots, u_k(t-n_u), d_k(t), \cdots, d_k(t-n_d)) \quad (8)$$

where $u_k(t)$ and $y_k(t)$ denote the system control input and measured output at time t of the k-th iteration; $d_k(t)$ is the generalized uncertainty including robot system uncertainties and external human-robot uncertain disturbances; n_y, n_u and n_d are unknown positive constants which represent the system order; $t \in 0, 1, 2\cdots, T-1$ is denoted as time interval; $f(\cdot)$ is defined as unknown nonlinear function.

The following assumptions are given to dynamically linearize the nonlinear system (8).

Assumption 1: The partial derivatives of $f(\cdot)$ with respect to control input $u_k(t)$ and uncertainty $d_k(t)$ are continuous.

Assumption 2: System (8) is generalized Lipschitz, i.e. if $\Delta u_k(t) \neq 0$ and $|\Delta d_k(t)| \neq 0$, satisfying $\Delta y_k(t+1) \leq b|\Delta u_k(t)|$ and $\Delta y_k(t+1) \leq c|\Delta d_k(t)|$, where $\Delta y_k(t+1) = y_k(t+1) - y_{k-1}(t+1)$, $\Delta u_k(t) = u_k(t) - u_{k-1}(t)$, $\Delta d_k(t) = d_k(t) - d_k(t-1)$, b, and c are positive constants.

Assumption 3: The uncertainty $d_k(t)$ is unknown and bounded. There is a constant d which satisfies $|d_k(t)| \leq d$.

Assumption 1 is a common condition for general nonlinear control systems. Assumption 2 means that a finite incremental input with bounded disturbance will lead to a finite incremental output, the constant b and c are determined by trials for qualitative analysis. Assumption 3 demonstrates that the uncertainty is bounded. Then, the following lemma demonstrates that a generalized discrete-time nonlinear system with uncertainty can be transformed into an equivalent dynamical linearized model.

Lemma 1 ([26]): For discrete-time nonlinear system (8) with Assumptions 1 – 3, if $|\Delta u_k(t)| \neq 0$ and $|\Delta d_k(t)| \neq 0$, then (8) can be converted into the following dynamic data model:

$$\Delta y_k(t+1) = \phi_k(t)\Delta u_k(t) + \psi_k(t)\Delta d_k(t) \quad (9)$$

where $\phi_k(t)$ and $\psi_k(t)$ are time-varying iteration-dependent PPD parameters, satisfying $|\phi_k(t)| \leq b$ and $|\psi_k(t)| \leq c$, respectively.

Equation (9) is the equivalence of system (8). When $\Delta u_k(t)$ and $\Delta d_k(t)$ are not large, Equation (9) can be regarded as a dynamic linear system with slowly time-varying parameters, that is, $\Delta u_k(t) \leq a$, a is a positive constant. This equation has practical implications in a control system with unknown factors, which is inevitable and often affects tracking performance.

B. Uncertainty Compensated High-Order Adaptive Iterative Learning Control

The goal of the controller is to find suitable bounded control input $u_k(t)$, so that the system output $y_k(t)$ can track the given desired trajectory $y_d(t)$. We estimate the uncertainty and the PPD parameters $\phi_k(t)$ and $\psi_k(t)$ to construct a model-free adaptive iterative learning controller with uncertainty compensation to guarantee the stable tracking of the upper limb rehabilitation robot.

To avoid the complexity of the subsequent equations, an intermediate variable is first defined

$$\omega_k(t) = \hat{\phi}_k(t)\Delta u_k(t) - \hat{\psi}_k(t)\Delta\hat{d}_k(t) \quad (10)$$

where $\hat{\phi}_k(t)$, $\hat{\psi}_k(t)$ and $\hat{d}_k(t)$ are the estimations of the PPD parameters $\phi_k(t)$, $\psi_k(t)$ and uncertainty $d_k(t)$, respectively.

Due to the time variability of uncertainty, a modified projection algorithm [27] is applied to design the criterion function for the estimation of uncertainty $d_k(t)$:

$$J(\Delta\hat{d}_k(t)) = |\Delta y_{k-1}(t+1) - \hat{\phi}_{k-1}(t)\Delta u_{k-1}(t) - \hat{\psi}_{k-1}(t)\Delta\hat{d}_k(t)|^2 + \delta_1|\Delta\hat{d}_k(t) - \Delta\hat{d}_{k-1}(t)|^2 \quad (11)$$

where δ_1 is a weighted constant. To minimize (11), let $\frac{\partial J(\Delta\hat{d}_k(t))}{\partial\Delta\hat{d}_k(t)} = 0$, the estimation of uncertainty is as follows:

$$\Delta\hat{d}_k(t) = \Delta\hat{d}_{k-1}(t) + \frac{\beta_1\hat{\psi}_{k-1}(t)(\Delta y_{k-1}(t+1) - \omega_{k-1}(t))}{\delta_1 + |\hat{\psi}_{k-1}(t)|^2} \quad (12)$$

where $\beta_1 \in (0, 1)$ is a step-size constant, and β_1 will influence the convergence speed of the learning process.

As the PPD parameters $\phi_k(t)$ and $\psi_k(t)$ are unknown and time-varying, a modified projection algorithm is also used to obtain the criterion functions for $\phi_k(t)$ and $\psi_k(t)$:

$$J(\hat{\phi}_k(t)) = |\Delta y_{k-1}(t+1) - \hat{\phi}_k(t)\Delta u_{k-1}(t) - \hat{\psi}_{k-1}(t)\Delta\hat{d}_{k-1}(t)|^2 + \delta_2|\hat{\phi}_k(t) - \hat{\phi}_{k-1}(t)|^2 \quad (13)$$

$$J(\hat{\psi}_k(t)) = |\Delta y_{k-1}(t+1) - \hat{\phi}_{k-1}(t)\Delta u_{k-1}(t) - \hat{\psi}_k(t)\Delta\hat{d}_{k-1}(t)|^2 + \delta_3|\hat{\psi}_k(t) - \hat{\psi}_{k-1}(t)|^2 \quad (14)$$

where δ_2 and δ_3 are the weighted constants that penalize the rate of change of parameters estimation.

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

To minimize (13) and (14), let $\frac{\partial J(\hat{\phi}_k(t))}{\partial \hat{\phi}_k(t)} = 0$ and $\frac{\partial J(\hat{\psi}_k(t))}{\partial \hat{\psi}_k(t)} = 0$, the PPD estimation algorithm can be obtained as follows.

$$\hat{\phi}_k(t) = \hat{\phi_{k-1}}(t) + \frac{\beta_2 \Delta u_{k-1}(t)(\Delta y_{k-1}(t+1) - \omega_{k-1}(t))}{\delta_2 + |\Delta u_{k-1}(t)|^2} \tag{15}$$

$$\hat{\psi}_k(t) = \hat{\psi_{k-1}}(t) + \frac{\beta_3 \Delta \hat{d_{k-1}}(t)(\Delta y_{k-1}(t+1) - \omega_{k-1}(t))}{\delta_3 + \left|\Delta \hat{d_{k-1}}(t)\right|^2} \tag{16}$$

where $\beta_2, \beta_3 \in (0, 1)$ are step factors, which can improve the generality of the PPD estimation algorithm.

To increase the tracking ability of the control scheme for the compensation to the time-varying parameter, and to restart the updating of PPD in terms of different operation points, the reset algorithm [28] is designed as follows:

$$\hat{\phi}_k(t) = \hat{\phi}_1, \quad \text{if} \left|\hat{\phi}_k(t)\right| \le \varepsilon_1, \text{or} |\Delta u_k(t)| \le \epsilon_1 \tag{17}$$

$$\hat{\psi}_k(t) = \hat{\psi}_1, \quad \text{if} \left|\hat{\psi}_k(t)\right| \le \varepsilon_2, \text{ or} \left|\Delta \hat{d}_k(t)\right| \le \epsilon_2 \tag{18}$$

$$\hat{d}_k(t) = \hat{d}_1, \quad \text{if} \left|\Delta \hat{d}_k(t)\right| > 2d \tag{19}$$

where $\hat{\phi}_1$ and $\hat{\psi}_1$ are the initial value of $\hat{\phi}_k(t)$ and $\hat{\psi}_k(t)$, respectively, ε_1,ε_2,ϵ_1 and ϵ_2 are the small positive constants, where $\hat{d}_1$ are the initial value of $\hat{d}_k(t)$, d is the boundary of uncertainty in Assumption 3.

From Equations (12) (15) and (16), we can update the uncertainty estimation value $\Delta \hat{d}_k(t)$ and the PPD estimation parameters $\hat{\phi}_k(t)$ and $\hat{\psi}_k(t)$ iteratively. Then, the following cost function is adopted for the controller to make $y_k(t)$ track desired output signal $y_d(t)$.

$$J(u_k(t)) = |y_d(t+1) - y_k(t+1)|^2 + \delta_4 |u_k(t) - u_{k-1}(t)|^2 \tag{20}$$

where $y_d(t+1)$ is the expected output and $\delta_4 > 0$ is weight factor. We define the tracking error as follows:

$$e_{k-1}(t+1) = y_d(t+1) - y_{k-1}(t+1) \tag{21}$$

Substituting (9) and (21) into (20), the cost function can be obtained:

$$\begin{aligned} J(u_k(t)) = {} & |e_{k-1}(t+1) - \phi_k(t)\Delta u_k(t) - \psi_k(t)\Delta d_k(t)|^2 \\ & + \delta_4 |u_k(t) - u_{k-1}(t)|^2 \end{aligned} \tag{22}$$

Let $\frac{\partial J(u_k(t))}{\partial u_k(t)} = 0$, and $\hat{\phi}_k(t)$, $\hat{\psi}_k(t)$ and $\Delta \hat{d}_k(t)$ are used to estimate $\phi_k(t)$, $\psi_k(t)$ and $\Delta d_k(t)$, then the control input is designed as follows:

$$u_k(t) = u_{k-1}(t) + \frac{\beta_4 \hat{\phi}_k(t) e_{k-1}(t+1)}{\delta_4 + \left|\hat{\phi}_k(t)\right|^2} - \frac{\beta_4 \hat{\phi}_k(t) \hat{\psi}_k(t) \Delta \hat{d}_k(t)}{\delta_4 + \left|\hat{\phi}_k(t)\right|^2} \tag{23}$$

where $\frac{\beta_4 \hat{\phi}_k(t) e_{k-1}(t+1)}{\delta_4 + \left|\hat{\phi}_k(t)\right|^2}$ can be considered as the tracking part of the control input, $-\frac{\beta_4 \hat{\phi}_k(t) \hat{\psi}_k(t) \Delta \hat{d}_k(t)}{\delta_4 + \left|\hat{\phi}_k(t)\right|^2}$ is the uncertainty compensation part, β_4 is step-size constant.

Then, the high-order learning scheme for PPD $\hat{\phi}_k(t)$, which uses the parameters from more than one past run to estimate the current iteration, is employed to redefine the criterion function (13) as follows:

$$\begin{aligned} J\left(\hat{\phi}_k(t)\right) = {} & \left|\Delta y_{k-1}(t+1) - \hat{\phi}_k(t)\Delta u_{k-1}(t) \right. \\ & \left. - \hat{\psi_{k-1}}(t)\Delta \hat{d_{k-1}}(t)\right|^2 \\ & + \delta_2 \left|\hat{\phi}_k(t) - \sum\nolimits_{i=1}^{m} \alpha_i \hat{\phi}_{k-i}(t)\right|^2 \end{aligned} \tag{24}$$

where m is the order, α_i is the weighting coefficient with $\sum_{i=1}^{m} \alpha_i = 1$, and $\alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_m$. High-order learning law can be used to improve the convergence speed [29].

Let $\frac{\partial J(\hat{\phi}_k(t))}{\partial \hat{\phi}_k(t)} = 0$, the high-order estimation algorithm of $\hat{\phi}_k(t)$ is expressed below:

$$\begin{aligned} \hat{\phi}_k(t) = {} & \frac{\beta_2 \Delta u_{k-1}(t)\left(\Delta y_{k-1}(t+1) - \hat{\psi_{k-1}}(t)\Delta \hat{d_{k-1}}(t)\right)}{\delta_2 + |\Delta u_{k-1}(t)|^2} \\ & + \frac{\delta_2 \sum_{i=1}^{m} \alpha_i \hat{\phi}_{k-i}(t)}{\delta_2 + |\Delta u_{k-1}(t)|^2} \end{aligned} \tag{25}$$

Similarly, the high-order scheme is extended to calculate control input to improve the tracking performance and the cost function (20) is rewrote as follows:

$$\begin{aligned} J(u_k(t)) = {} & |e_{k-1}(t+1) - \phi_k(t)\Delta u_k(t) - \psi_k(t)\Delta d_k(t)|^2 \\ & + \delta_4 \left|u_k(t) - \sum\nolimits_{j=1}^{n} \mu_j u_{k-j}(t)\right|^2 \end{aligned} \tag{26}$$

where n is the order, μ_j is the weighting coefficient with $\sum_{j=1}^{n} \mu_j = 1$ and $\mu_1 \ge \mu_2 \ge \cdots \ge \mu_j$.

Let $\frac{\partial J(u_k(t))}{\partial u_k(t)} = 0$, the control input can be obtained with the same calculation method.

$$\begin{aligned} u_k(t) = {} & \frac{\left(\hat{\phi}_k(t)\right)^2 u_{k-1}(t) + \delta_4 \sum_{j=1}^{n} \mu_j u_{k-j}(t)}{\delta_4 + \left|\hat{\phi}_k(t)\right|^2} \\ & + \frac{\beta_4 \hat{\phi}_k(t)(e_{k-1}(t+1) - \hat{\psi}_k(t)\Delta \hat{d}_k(t))}{\delta_4 + \left|\hat{\phi}_k(t)\right|^2} \end{aligned} \tag{27}$$

Since $\hat{\phi}_k(t)$ and $u_k(t)$ are calculated based on the previous iterations, high-order learning algorithms (25) and (27) can be applied to obtain $\hat{\phi}_k(t)$ and $u_k(t)$ when $k > m(orn)$. The original algorithms (15) and (23) are used when $1 < k \le m(orn)$. So $\hat{\phi}_k(t)$ and $u_k(t)$ can be expressed as in (28) and (29), shown at the bottom of the next page. Therefore, the structure diagram of the proposed UCHAILC is illustrated in Fig. 4.

C. Convergence Analysis

Assumption 4: The PPD parameters satisfy $\hat{\phi}_k(t) > 0 (\hat{\phi}_k(t) < 0)$ and $\hat{\psi}_k(t) > 0 (\hat{\psi}_k(t) < 0)$. It is assumed that $\hat{\phi}_k(t) > 0$ and $\hat{\psi}_k(t) > 0$ at any instant t and iteration k.

Assumption 4 is inspired by [27]. It is a common condition in practice that means that a change in the system output and the corresponding input satisfies the positive correlation.

Theorem 1: Consider nonlinear system (8) with uncertainty under the Assumptions 1 - 4. When the parameters $\Delta d_k(t)$, $\phi_k(t)$, and $\psi_k(t)$ are estimated by (12) (28) (16), then $\Delta \hat{d}_k(t)$, $\hat{\phi}_k(t)$ and $\hat{\psi}_k(t)$ are bounded if $\beta_1, \beta_2, \beta_3 \in (0.5, 1)$ and

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

δ_1,δ_2,δ_3, $\Delta u_k(\mathrm{t})$, $\Delta\hat{d}_k(t)$, and $\hat{\psi}_k(\mathrm{t})$ satisfy the following inequalities:

$$\begin{aligned}\delta_1 &> \frac{(\beta_1(a+2d))^2}{4(1-\beta_1)}, \delta_2 > \frac{(\beta_2(2d+c))^2}{4(1-\beta_2)},\\ \delta_3 &> \frac{(\beta_3(a+c))^2}{4(1-\beta_3)},\\ |\Delta u_k(t)| &> \sqrt{\frac{\delta_2}{2\beta_2-1}}, \left|\Delta\hat{d}_k(t)\right| > \sqrt{\frac{\delta_3}{2\beta_3-1}},\\ \left|\hat{\psi}_k(\mathrm{t})\right| &> \sqrt{\frac{\delta_1}{2\beta_1-1}}\end{aligned} \tag{30}$$

Proof: Define the parameter estimation errors as $\tilde{\Delta}d_k(\mathrm{t}) = \Delta\hat{d}_k(\mathrm{t}) - \Delta d_k(\mathrm{t})$, $\tilde{\phi}_k(\mathrm{t}) = \hat{\phi}_k(\mathrm{t}) - \phi_k(\mathrm{t})$, $\tilde{\psi}_k(\mathrm{t}) = \hat{\psi}_k(\mathrm{t}) - \psi_k(\mathrm{t})$. When $1 < k \le m$, substituting Eq. (12), (15) and (16) into the above parameter estimation errors, the estimation errors can be written as follows:

$$\begin{aligned}\begin{bmatrix}\tilde{\phi}_k(\mathrm{t})\\ \tilde{\psi}_k(\mathrm{t})\\ \tilde{\Delta}d_k(\mathrm{t})\end{bmatrix} &= \begin{bmatrix}M_{11} & M_{12} & M_{13}\\ M_{21} & M_{22} & M_{23}\\ M_{31} & M_{32} & M_{33}\end{bmatrix}\begin{bmatrix}\tilde{\phi_{k-1}}(\mathrm{t})\\ \tilde{\psi_{k-1}}(\mathrm{t})\\ \tilde{\Delta}d_{k-1}(\mathrm{t})\end{bmatrix}\\ &+ \begin{bmatrix}\phi_{k-1}(\mathrm{t}) - \phi_k(\mathrm{t})\\ \psi_{k-1}(\mathrm{t}) - \psi_k(\mathrm{t})\\ \Delta d_{k-1}(\mathrm{t}) - \Delta d_k(\mathrm{t})\end{bmatrix}\end{aligned} \tag{31}$$

where $M_{11} = 1 - \frac{\beta_2(\Delta u_{k-1}(t))^2}{\delta_2+(\Delta u_{k-1}(t))^2}$, $M_{12} = -\frac{\beta_2\Delta u_{k-1}(t)\Delta\hat{d_{k-1}}(\mathrm{t})}{\delta_2+(\Delta u_{k-1}(t))^2}$, $M_{13} = -\frac{\beta_2\Delta u_{k-1}(t)\psi_{k-1}(\mathrm{t})}{\delta_2+(\Delta u_{k-1}(t))^2}$, $M_{21} = -\frac{\beta_3\Delta\hat{d_{k-1}}(\mathrm{t})\Delta u_{k-1}(t)}{\delta_3+(\Delta\hat{d_{k-1}}(\mathrm{t}))^2}$, $M_{22} = 1 - \frac{\beta_3(\Delta\hat{d_{k-1}}(\mathrm{t}))^2}{\delta_3+(\Delta\hat{d_{k-1}}(\mathrm{t}))^2}$, $M_{23} = -\frac{\beta_3\Delta\hat{d_{k-1}}(\mathrm{t})\psi_{k-1}(\mathrm{t})}{\delta_3+(\Delta\hat{d_{k-1}}(\mathrm{t}))^2}$, $M_{31} = -\frac{\beta_1\hat{\psi_{k-1}}(\mathrm{t})\Delta u_{k-1}(t)}{\delta_1+(\hat{\psi_{k-1}}(\mathrm{t}))^2}$, $M_{32} = -\frac{\beta_1\hat{\psi_{k-1}}(\mathrm{t})\Delta d_{k-1}(\mathrm{t})}{\delta_1+(\hat{\psi_{k-1}}(\mathrm{t}))^2}$, $M_{33} = 1 - \frac{\beta_1(\hat{\psi_{k-1}}(\mathrm{t}))^2}{\delta_1+(\hat{\psi_{k-1}}(\mathrm{t}))^2}$.

Since $|\phi_k(t)| \le b$, $|\psi_k(t)| \le c$ and $|d_k(t)| \le d$, the L1 norm of the vector in (31) satisfies following inequality:

$$\left\|\begin{matrix}\phi_{k-1}(\mathrm{t}) - \phi_k(\mathrm{t})\\ \psi_{k-1}(\mathrm{t}) - \psi_k(\mathrm{t})\\ \Delta d_{k-1}(\mathrm{t}) - \Delta d_k(\mathrm{t})\end{matrix}\right\|_1 \le 2(b+c+2d) \tag{32}$$

To ensure the convergence of the iterative algorithm, the conditions for the iterative matrix M satisfying $\|M\| < 1$ are:

$$\frac{\beta_2|\Delta u_{k-1}(t)|(\left|\Delta\hat{d_{k-1}}(\mathrm{t})\right| + \left|\hat{\psi_{k-1}}(\mathrm{t})\right|)}{\delta_2+(\Delta u_{k-1}(t))^2} < 1 - \frac{\beta_2(\Delta u_{k-1}(t))^2}{\delta_2+(\Delta u_{k-1}(t))^2} \tag{33a}$$

$$\frac{\beta_3\left|\Delta\hat{d_{k-1}}(\mathrm{t})\right|(|\Delta u_{k-1}(t)| + \left|\hat{\psi_{k-1}}(\mathrm{t})\right|)}{\delta_3+(\Delta\hat{d_{k-1}}(\mathrm{t}))^2} < 1 - \frac{\beta_3(\Delta\hat{d_{k-1}}(\mathrm{t}))^2}{\delta_3+(\Delta\hat{d_{k-1}}(\mathrm{t}))^2} \tag{33b}$$

$$\frac{\beta_1\left|\hat{\psi_{k-1}}(\mathrm{t})\right|(|\Delta u_{k-1}(t)| + |\Delta d_{k-1}(\mathrm{t})|)}{\delta_1+(\hat{\psi_{k-1}}(\mathrm{t}))^2} < 1 - \frac{\beta_1(\hat{\psi_{k-1}}(\mathrm{t}))^2}{\delta_1+(\hat{\psi_{k-1}}(\mathrm{t}))^2} \tag{33c}$$

$$1 - \frac{\beta_2(\Delta u_{k-1}(t))^2}{\delta_2+(\Delta u_{k-1}(t))^2} < \frac{1}{2} \tag{33d}$$

$$1 - \frac{\beta_3(\Delta\hat{d_{k-1}}(\mathrm{t}))^2}{\delta_3+(\Delta\hat{d_{k-1}}(\mathrm{t}))^2} < \frac{1}{2} \tag{33e}$$

$$1 - \frac{\beta_1(\hat{\psi_{k-1}}(\mathrm{t}))^2}{\delta_1+(\hat{\psi_{k-1}}(\mathrm{t}))^2} < \frac{1}{2} \tag{33f}$$

Rearranging inequalities (33), we have (30). According to the property of norm, and $\|M\| \le \gamma < 1$, taking the norm to (31), we can obtain

$$\begin{aligned}\left\|\begin{matrix}\tilde{\phi}_k(\mathrm{t})\\ \tilde{\psi}_k(\mathrm{t})\\ \tilde{\Delta}d_k(\mathrm{t})\end{matrix}\right\|_1 &\le \gamma\left\|\begin{matrix}\tilde{\phi_{k-1}}(\mathrm{t})\\ \tilde{\psi_{k-1}}(\mathrm{t})\\ \tilde{\Delta}d_{k-1}(\mathrm{t})\end{matrix}\right\|_1 + (2b+2c+4d) \le \cdots\\ &\le \gamma^{k-1}\left\|\begin{matrix}\tilde{\phi}_1(\mathrm{t})\\ \tilde{\psi}_1(\mathrm{t})\\ \tilde{\Delta}d_1(\mathrm{t})\end{matrix}\right\|_1 + \frac{(1-\gamma^{k-1})(2b+2c+4d)}{1-\gamma}\end{aligned} \tag{34}$$

From (34), $\tilde{\phi}_k(\mathrm{t})$, $\tilde{\psi}_k(\mathrm{t})$ and $\tilde{\Delta}d_k(\mathrm{t})$ are bounded at any instant t and iteration k. Since $\phi_k(t)$, $\psi_k(t)$ and $\Delta d_k(\mathrm{t})$ are bounded according to assumptions 2 and 3, the upper boundedness of $\hat{\phi}_k(\mathrm{t})$, $\hat{\psi}_k(\mathrm{t})$ and $\Delta\hat{d}_k(\mathrm{t})$ can be guaranteed.

In addition, the PPD parameter $\phi_k(\mathrm{t})$ is estimated by a high-order learning algorithm (25) when $k > m$. Substituting the dynamic linearization model (9) into (25) and taking the absolute value at both sides, we have as in (35), shown at the bottom of the next page.

According to Lemma 1, $\phi_{k-1}(t)$, $\Delta d_{k-1}(t)$ are bounded, meanwhile $\tilde{\Delta}d_{k-1}(\mathrm{t})$, $\tilde{\psi_{k-1}}(\mathrm{t})$ and $\hat{\psi_{k-1}}(t)$ are bounded. In addition, when $1 < \mathrm{k} \le \mathrm{m}$, $\hat{\phi}_k(t)$ is bounded from (34), so $\left|\sum_{i=1}^{m}\alpha_i\hat{\phi}_{k-i}(t)\right|$ is bounded too. Therefore, if β_1, β_2, $\beta_3 \in (0.5, 1]$ and δ_1,δ_2,δ_3, $\Delta u_k(\mathrm{t})$, $\Delta\hat{d}_k(t)$, and $\hat{\psi}_k(\mathrm{t})$ satisfy (30), $\Delta\hat{d}_k(t)$, $\hat{\phi}_k(\mathrm{t})$ and $\hat{\psi}_k(\mathrm{t})$ are bounded.

Theorem 2: Consider systems (8) satisfying Assumptions 1-4 with uncertainty compensation controller (29). If the PPD parameters $\phi_k(t)$ and $\psi_k(t)$ and uncertainty $\Delta d_k(t)$ are estimated in (28) (16) (12), then there exists $\beta_4 \in (0, 1)$ and $\delta_4 > \frac{(\beta_4 b)^2}{4}$, such that: 1) when $k \to \infty$, the system output tracking error is convergent; 2) the control law $u_k(\mathrm{t})$ is bounded.

Proof: According to (9) and (21), we have

$\mathrm{e}_k(t+1) - \mathrm{e}_{k-1}(t+1)$

$$\hat{\phi}_k(\mathrm{t}) = \begin{cases}\hat{\phi_{k-1}}(\mathrm{t}) + \dfrac{\beta_2\Delta u_{k-1}(t)(\Delta y_{k-1}(t+1) - \omega_{k-1}(t))}{\delta_2 + |\Delta u_{k-1}(t)|^2}, & 1 < k \le m\\ \dfrac{\beta_2\Delta u_{k-1}(t)\left(\Delta y_{k-1}(t+1) - \hat{\psi_{k-1}}(t)\Delta\hat{d_{k-1}}(t)\right)}{\delta_2 + |\Delta u_{k-1}(t)|^2} + \dfrac{\delta_2\sum_{i=1}^{m}\alpha_i\hat{\phi}_{k-i}(t)}{\delta_2 + |\Delta u_{k-1}(t)|^2}, & k > m\end{cases} \tag{28}$$

$$u_k(\mathrm{t}) = \begin{cases}u_{k-1}(\mathrm{t}) + \dfrac{\beta_4\hat{\phi}_k(\mathrm{t})\mathrm{e}_{k-1}(t+1)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} - \dfrac{\beta_4\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2}, & 1 < k \le n\\ \dfrac{\left(\hat{\phi}_k(\mathrm{t})\right)^2 u_{k-1}(\mathrm{t}) + \delta_4\sum_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} + \dfrac{\beta_4\hat{\phi}_k(\mathrm{t})(\mathrm{e}_{k-1}(t+1) - \hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t))}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2}, & k > n\end{cases} \tag{29}$$

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

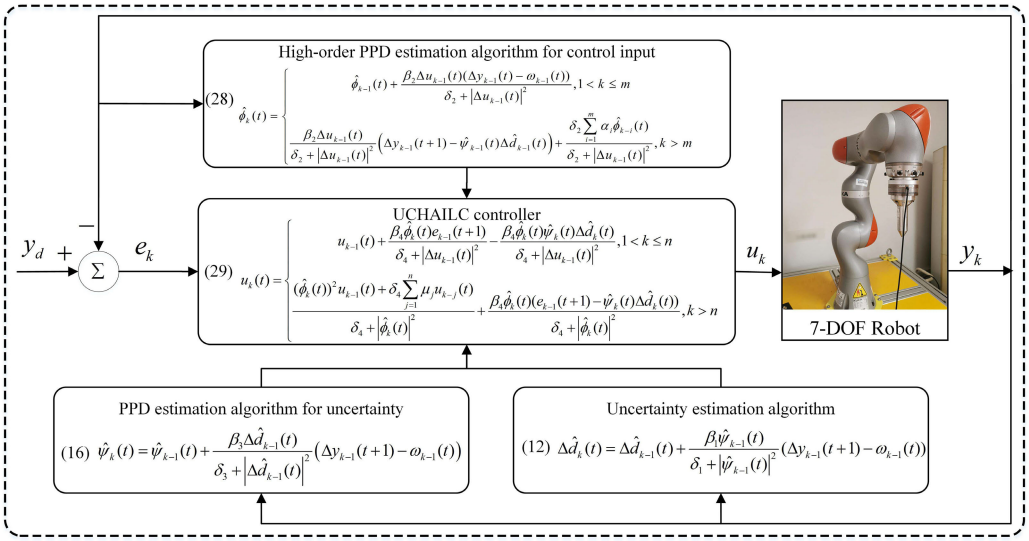


Fig. 4. Structure diagram of the proposed UCHAILC.

$$
\begin{aligned}
&= (y_d(t+1) - y_k(t+1)) \\
&\quad - (y_d(t+1) - y_{k-1}(t+1)) \\
&= -\Delta y_k(t+1) = -\phi_k(t)\Delta u_k(t) - \psi_k(t)\Delta d_k(t) \\
&= -\phi_k(t)\Big(\frac{\beta_4\hat{\phi}_k(\mathrm{t})\mathrm{e}_{k-1}(t+1)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} \\
&\quad + \frac{\left(\hat{\phi}_k(\mathrm{t})\right)^2 u_{k-1}(\mathrm{t}) + \delta_4\sum_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} \\
&\quad - \frac{\beta_4\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} - u_{k-1}(\mathrm{t})\Big) - \psi_k(t)\Delta d_k
\end{aligned} \tag{36}
$$

Take the absolute values of both sides, we can get

$$
\begin{aligned}
|\mathrm{e}_k(t+1)| &\le \left|1 - \frac{\beta_4\phi_k(t)\hat{\phi}_k(\mathrm{t})}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2}\right| |\mathrm{e}_{k-1}(t+1)| \\
&\quad + \left|\frac{\beta_4\phi_k(t)\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} - \psi_k(t)\Delta d_k(t)\right| \\
&\quad + \left|\frac{\delta_4\phi_k(t)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2}\right| \left|u_{k-1}(\mathrm{t}) - \sum\nolimits_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})\right| \\
&\le |\sigma||\mathrm{e}_{k-1}(t+1)| \\
&\quad + \left|\frac{\beta_4\phi_k(t)\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} - \psi_k(t)\Delta d_k(t)\right| \\
&\quad + \left|\frac{\delta_4\phi_k(t)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2}\right| \left|u_{k-1}(\mathrm{t}) - \sum\nolimits_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})\right|
\end{aligned} \tag{37}
$$

where $\sigma = 1 - \frac{\beta_4\phi_k(t)\hat{\phi}_k(\mathrm{t})}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2}$.

The convergence of the system output tracking error needs to satisfy the following condition

$$
0 < \frac{\beta_4\phi_k(t)\hat{\phi}_k(\mathrm{t})}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} \le \frac{\beta_4\phi_k(t)}{2\sqrt{\delta_4}} < 1 \tag{38}
$$

Since $\hat{\phi}_k(\mathrm{t})$ is bounded and $|\phi_k(t)| \le b$, thereby, $\delta_4 > \frac{(\beta_4 b)^2}{4}$, $\beta_4 \in (0, 1)$ need to be satisfied. From Assumption 4 and Theorem 1, when $k \to \infty$ there must exist a small constant η satisfying

$$
\left|\frac{\beta_4\phi_k(t)\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4 + \left|\hat{\phi}_k(\mathrm{t})\right|^2} - \psi_k(t)\Delta d_k(t)\right|
$$

$$
\begin{aligned}
\left|\hat{\phi}_k(\mathrm{t})\right| &= \left|\frac{\delta_2\sum_{i=1}^{m}\alpha_i\hat{\phi}_{k-i}(t)}{\delta_2 + |\Delta u_{k-1}(t)|^2}\right| \\
&\quad + \left|\frac{\beta_2\Delta u_{k-1}(t)\big(\phi_{k-1}(t)\Delta u_{k-1}(t) + \psi_{k-1}(t)\Delta d_{k-1}(t) - \hat{\psi_{k-1}}(t)\Delta\hat{d_{k-1}}(t)\big)}{\delta_2 + |\Delta u_{k-1}(t)|^2}\right| \\
&\le \left|\frac{\delta_2}{\delta_2 + |\Delta u_{k-1}(t)|^2}\right| \left|\sum\nolimits_{i=1}^{m}\alpha_i\hat{\phi}_{k-i}(t)\right| \\
&\quad + |\beta_2\phi_{k-1}(t)|\left|\frac{(\Delta u_{k-1}(t))^2}{\delta_2 + |\Delta u_{k-1}(t)|^2}\right| \\
&\quad + \left|\frac{\beta_2\Delta u_{k-1}(t)\big(\tilde{\psi_{k-1}}(\mathrm{t})\Delta d_{k-1}(t) + \hat{\psi_{k-1}}(t)\tilde{\Delta}d_{k-1}(\mathrm{t})\big)}{\delta_2 + |\Delta u_{k-1}(t)|^2}\right|
\end{aligned} \tag{35}
$$

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

$$+\left|\frac{\delta_4\phi_k(t)}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right|\left|u_{k-1}(\mathrm{t})-\sum_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})\right|\le\eta \quad (39)$$

Taking the norm to both sides of (37), we can get

$$\begin{aligned}|\mathrm{e}_k(t+1)| &\le |\sigma||\mathrm{e}_{k-1}(t+1)|+\eta\\ &\le\cdots\le\left|\sigma^{k-1}\right||\mathrm{e}_1(t+1)|+\frac{(1-\sigma^{k-1})}{1-\sigma}\eta \quad (40)\end{aligned}$$

Since $0<\sigma<1$, the boundedness of the tracking error $\mathrm{e}_k(t+1)$ is guaranteed.

when $1<k\le n$, we have the control input from (23)

$$\begin{aligned}|\Delta u_k(\mathrm{t})| &= \left|\frac{\beta_4\hat{\phi}_k(\mathrm{t})\mathrm{e}_{k-1}(t+1)}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}-\frac{\beta_4\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right|\\ &\le\left|\frac{\beta_4\hat{\phi}_k(\mathrm{t})}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right||\mathrm{e}_{k-1}(t+1)|\\ &\quad+\left|\frac{\beta_4\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right|\\ &\le|\varrho||\mathrm{e}_{k-1}(t+1)|+|\kappa| \quad (41)\end{aligned}$$

where $\varrho=\frac{\beta_4\hat{\phi}_k(\mathrm{t})}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}$, $\kappa=\frac{\beta_4\hat{\phi}_k(\mathrm{t})\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t)}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}$ are bounded, and $0<\varrho<1$.

According to (40) and (41), we have

$$\begin{aligned}|u_k(\mathrm{t})| &= |u_k(\mathrm{t})-u_{k-1}(\mathrm{t})+u_{k-1}(\mathrm{t})|\\ &\le|\Delta u_k(\mathrm{t})|+|u_{k-1}(\mathrm{t})|\\ &\le\cdots\le|\Delta u_k(\mathrm{t})|+|\Delta u_{k-1}(\mathrm{t})|+\cdots|\Delta u_1(\mathrm{t})|+|u_0(\mathrm{t})|\\ &\le|\varrho|(|\mathrm{e}_{k-1}(t+1)|+\cdots+|\mathrm{e}_1(t+1)|)+k|\kappa|\\ &\quad+|u_0(\mathrm{t})|\\ &\le|\varrho|(\frac{1-\sigma^{k-1}}{1-\sigma}|\mathrm{e}_1(t+1)|\\ &\quad+\frac{k(1-\sigma)-\left(1-\sigma^{k-1}\right)}{(1-\sigma)^2}\eta)+k|\kappa|+|u_0(\mathrm{t})| \quad (42)\end{aligned}$$

Because $\hat{\phi}_k(\mathrm{t})$, $\mathrm{e}_1(t+1)$ and $u_0(\mathrm{t})$ are bounded, $u_k(\mathrm{t})$ is also bounded.

When $k>n$, the control input is from (27)

$$\begin{aligned}|u_k(\mathrm{t})| &= \left|\frac{\left(\hat{\phi}_k(\mathrm{t})\right)^2u_{k-1}(\mathrm{t})}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right|\\ &\quad+\left|\frac{\delta_4\sum_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right|\\ &\quad+\left|\frac{\beta_4\hat{\phi}_k(\mathrm{t})(\mathrm{e}_{k-1}(t+1)-\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t))}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right|\\ &=\left|1-\frac{\delta_4}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right||u_{k-1}(\mathrm{t})|\\ &\quad+\left|\frac{\delta_4}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right|\left|\sum_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})\right|\\ &\quad+\left|\frac{\beta_4\hat{\phi}_k(\mathrm{t})(\mathrm{e}_{k-1}(t+1)-\hat{\psi}_k(\mathrm{t})\Delta\hat{d}_k(t))}{\delta_4+\left|\hat{\phi}_k(\mathrm{t})\right|^2}\right| \quad (43)\end{aligned}$$

Since the boundedness of $u_k(\mathrm{t})$ when $k<n$ has been proved, $\left|\sum_{j=1}^{n}\mu_j u_{k-j}(\mathrm{t})\right|$ is convergent, whilst $\mathrm{e}_k(t+1)$,$\hat{\phi}_k(\mathrm{t})$, $\hat{\psi}_k(\mathrm{t})$ and $\Delta\hat{d}_k(t)$ have been verified to be bounded, the boundedness of the $u_k(\mathrm{t})$ is guaranteed. This completes the proof.

IV. Simulation and Experiments

We conduct comparison simulation on MATLAB and experiments on a 7-DOF flexible rehabilitation robot to verify the effectiveness of the proposed UCHAILC method.

A. Simulation Validation

Firstly, we perform a simulation test on MATLAB to test the feasibility of the UCHAILC method. The proposed UCHAILC method and MFAILC [16] are applied to control the nonlinear system (44), respectively, which contains time-varying parameters and random disturbance. In MFAILC, the robot system is converted into the dynamic linearization model $\Delta\mathrm{y}_k(\mathrm{t}+1)=\phi_k(t)\Delta u_k(t)$, and the control knowledge of the previous only one run is used to update the current PPD parameter and control input. The nonlinear system is defined as follows:

$$y(t+1)=\begin{cases}\dfrac{y(t)}{1+y^2(t)}+(u(t)+d(t))^3, 0\le t<150\\ \dfrac{(y(t)y(t-1)y(t-2)u(t-1)(y(t-2)-1))}{1+y^2(t-1)+y^2(t-2)}\\ +\dfrac{a(t)(u(t)+d(t))}{1+y^2(t-1)+y^2(t-2)}, 150\le t\le 300\end{cases} \quad (44)$$

where $y(t),u(t),d(t)$ are defined in (8), and parameters are time-varying.

The desired trajectory is:

$$y_d(t+1)=\begin{cases}0.5*(-1)^{round(t/100)}, 0\le t\le 150\\ 0.5\sin\left(\dfrac{\pi t}{100}\right)+0.3\cos\left(\dfrac{\pi t}{50}\right), 150<t\le 200\\ 0.5*(-1)^{round(t/100)}, 200<t\le 300\end{cases} \quad (45)$$

During simulation, we assume that the disturbance function with the non-repetitive time-varying parameter is $d(k,t)=0.1*rand(1,1)$. The value ranges of iteration parameters and weighting coefficients are determined by Theorems 1 and 2, and different parameter values within the range are tested in simulation. Since the purpose of the simulation is to verify the effectiveness of the algorithm, instead of selecting the optimal parameters, we only display the set of iteration parameters with the best performance on the robot platform (See IV.B). Other parameters are determined according to the system, we choose the initial values of output and input $y(1)=0.5$, $u(1)=0$ and disturbance $\hat{d}(1)=0$, respectively. The initial values of the PPD parameters are selected as $\hat{\phi}_1=10$, $\hat{\psi}_1=0.75$.

The tracking results of the MFAILC and the UCHAILC are shown in Fig. 5. The black solid line denotes the desired trajectory and the rest five lines are the trajectories of the 4^{th}, 10^{th}, 20^{th}, 50^{th} and 100^{th} iterations. As the number of iteration increases, the actual trajectory gradually approaches

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

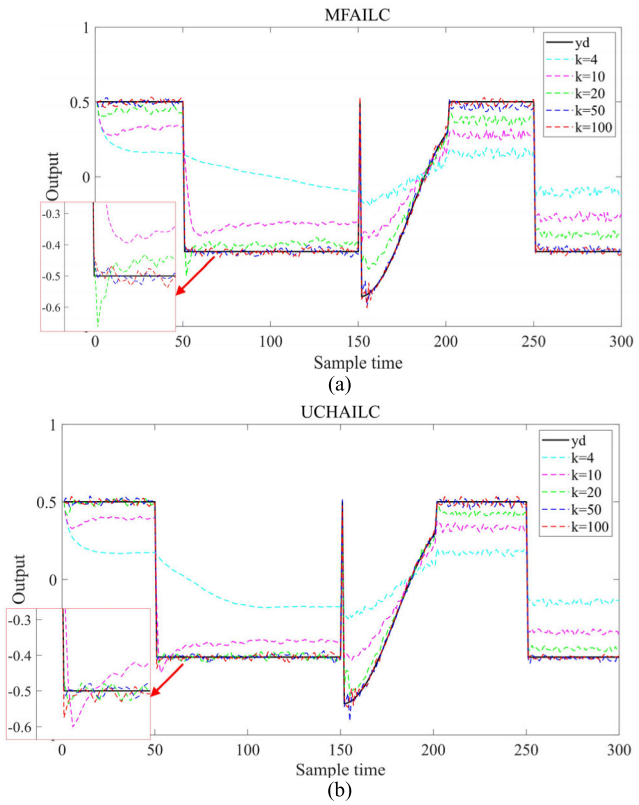


Fig. 5. Trajectory tracking results of MFAILC and UCHAILC.

the desired trajectory. Due to the superimposed random noise, the actual trajectory fluctuates around the desired trajectory, even the last 100th trajectory still cannot coincide with y_d. From the enlarged image in the corner, we can see that the magenta 10th iteration trajectory of MFAILC is completely deviated from the desired black line, the magenta 10th line in UCHAILC is close to the desired trajectory; and the green 20th iteration trajectory of MFAILC still has a large deviation, while the 20th's deviation of UCHAILC is much smaller. Meanwhile, the error of the 50th and 100th of MFAILC and UCHAILC are close to the desired trajectory. To illustrate the error clearly, Fig. 6 shows the curves of the maximum tracking error of each iteration, it is obvious the curve of UCHAILC is almost all below the MFAILC and it is also smoother than MFAILC. Therefore, the UCHAILC can achieve a fast convergence of the maximum tracking error. Compared with MFAILC, the UCHAILC takes robot system model uncertainties and external human-robot uncertain disturbances into consideration. Thus the UCHAILC can obtain better tracking performance when the system contains uncertainty in simulation.

B. Robot-Assisted Circular Trajectory Tracking Control

To ensure the safety and accuracy of the rehabilitation training, the proposed UCHAILC method is applied to 7-DOF upper limb rehabilitation robot (Fig. 1) to improve the tracking accuracy of the robot. The control input is the set value of the seven joints of the robot. The main task is to make each of the seven joints keep track of their respective desired trajectory,

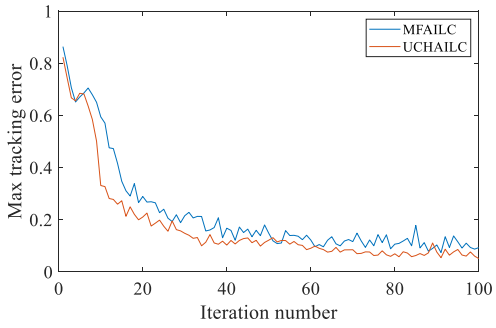


Fig. 6. Maximum tracking error of MFAILC and UCHAILC.

to make the manipulator overcome system model uncertainties and external human-robot uncertain disturbances.

The target trajectory is designed to follow a circular trajectory 3 times in Cartesian space, then the Cartesian trajectory is transformed into joint space trajectories through inverse kinematics, which is implemented with the damped least squares method in KUKA Sunrise Toolbox. The black solid lines in Fig. 7are the desired trajectories. MFAILC and UCHAILC are used to control the robot to move along a circular trajectory with the participant. After trying different iteration parameters and weighting coefficients, the set of parameters with the best performance on the robot platform are chosen as $\beta_1 = 0.9$, $\beta_2 = 0.6$, $\beta_3 = 0.6$, $\beta_4 = 0.9$, $\delta_1 = 0.01$,$\delta_2 = 0.02$, $\delta_3 = 0.02$, $\delta_4 = 0.01$. The weighting coefficients of high-order learning law are $\alpha_1 = 0.5$, $\alpha_2 = 0.3$, $\alpha_3 = 0.2$, $\mu_1 = 0.5$, $\mu_2 = 0.3$, $\mu_3 = 0.2$, and the high-order estimation algorithm starts from the 4th iteration. Considering the relative positions between the manipulator and the human, we set the initial values of control input as $u_1 = [0.00, 0.64, 0.00, -1.04, 0.00, 1.46, 0.00]$ and uncertainty $\hat{d}_1 = [0, 0, 0, 0, 0, 0, 0]$. The initial values of PPD parameters are selected as $\hat{\phi}_1 = 1$,$\hat{\psi}_1 = 0.75$.

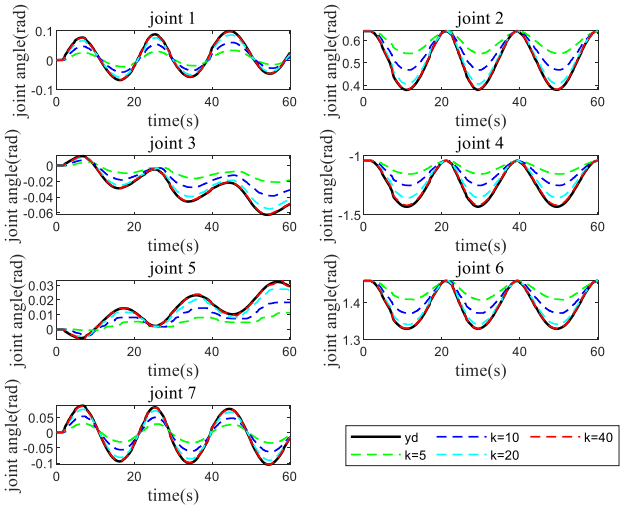
The tracking results of MFAILC and UCHAILC are shown in Fig. 7. The black solid lines represent the desired joint angles, and the dotted lines are the actual joint angles in different iterations. It can be seen that seven joints of the upper limb rehabilitation robot can gradually get close to the desired trajectory with the number of iteration increases. In addition, the red last iteration trajectories almost coincide with the desired trajectories, and the tracking error of the last iteration is acceptable in practice. Table I and II list the max tracking error rate of a trajectory for the two methods, which can be obtained through the following equation:

$$r = \max_{0 \le t \le 60} \left(\frac{|y_k(t) - y_d(t)|}{|y_d(t)|} \right) \tag{46}$$

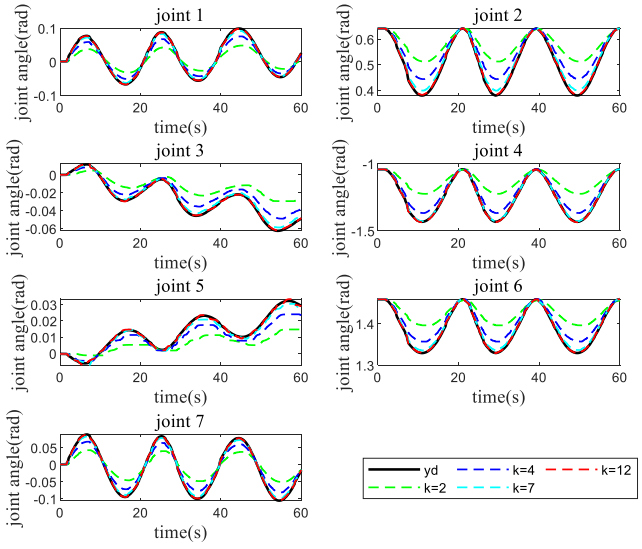
We can find that the tracking error of UCHAILC can reach a satisfactory accuracy after about 10 iterations, while MFAILC needs almost 30 times to achieve a similar accuracy. Therefore, the efficiency of UCHAILC is much higher than MFAILC. Fig. 8 presents the control input of joint 4 as an example. The curves of control input in UCHAILC fluctuated more obviously than that in MFAILC, which indicates that UCHAILC performs better robustness towards the system model uncertainties and external human-robot uncertain disturbances than MFAILC.

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

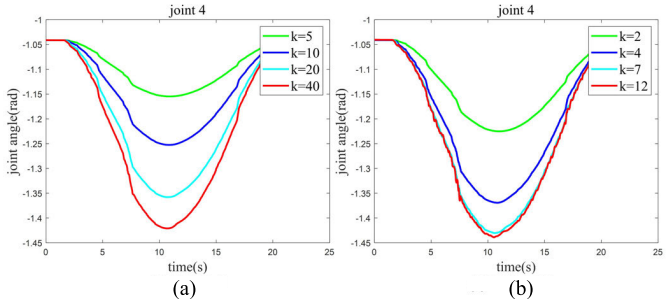


(a)



(b)

Fig. 7. Joint angles tracking results for circular trajectory.



(a) (b)

Fig. 8. Control input of joint 4 for circular trajectory tracking.

Fig. 9 is the tracking results of the end-effecter in Cartesian space. The black curve is the desired trajectory, and 3 circles of the trajectories are completely in consistent; the blue and red curves represent the actual Cartesian trajectory of the last iteration of MFAILC and UCHAILC. In the middle enlarged view, we can see that the actual 3 circle trajectories of the two methods are not the same as desired trajectory because of unavoidable error, but the trajectory of the proposed UCHAILC method is closer to the desired trajectory than MFAILC.

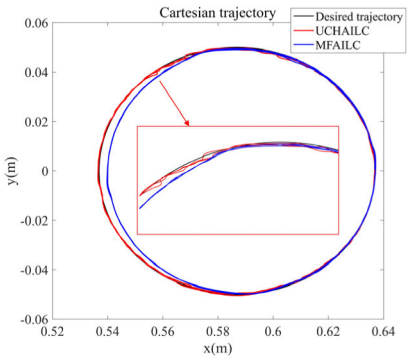


Fig. 9. Cartesian trajectory of MFAILC and UCHAILC.

TABLE I

MAX TRACKING ERROR RATE OF MFAILC IN TWO TRAJECTORY TRACKING EXPERIMENTS

Iteration		k = 5	k = 10	k = 20	k = 30	k = 40
Circular trajectory	j1	0.0663	0.0396	0.0139	**0.0060**	0.0037
	j2	0.1625	0.0907	0.0286	**0.0091**	0.0071
	j3	0.0427	0.0252	0.0088	**0.0033**	0.0013
	j4	0.2795	0.1834	0.0769	**0.0328**	0.0158
	j5	0.0229	0.0140	0.0049	**0.0019**	0.0014
	j6	0.0799	0.0435	0.0136	**0.0042**	0.0040
	j7	0.0719	0.0434	0.0155	**0.0062**	0.0040
Drinking trajectory	j1	0.4228	0.2359	0.0731	0.0230	**0.0073**
	j2	0.1814	0.1016	0.0320	0.0119	**0.0059**
	j3	0.0997	0.0598	0.0215	0.0078	**0.0034**
	j4	0.1223	0.0809	0.0352	0.0169	**0.0088**
	j5	0.8295	0.4838	0.1637	0.0562	**0.0195**
	j6	0.4375	0.2324	0.0659	0.0189	**0.0076**
	j7	0.3915	0.2343	0.0834	0.0320	**0.0134**

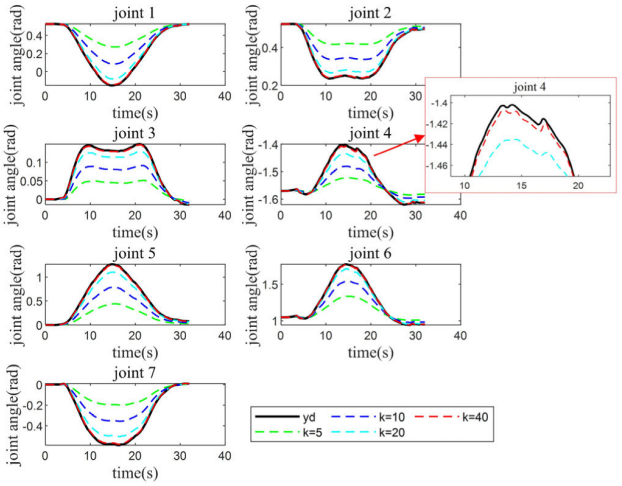
TABLE II

MAX TRACKING ERROR RATE OF UCHAILC IN TWO TRAJECTORY TRACKING EXPERIMENTS

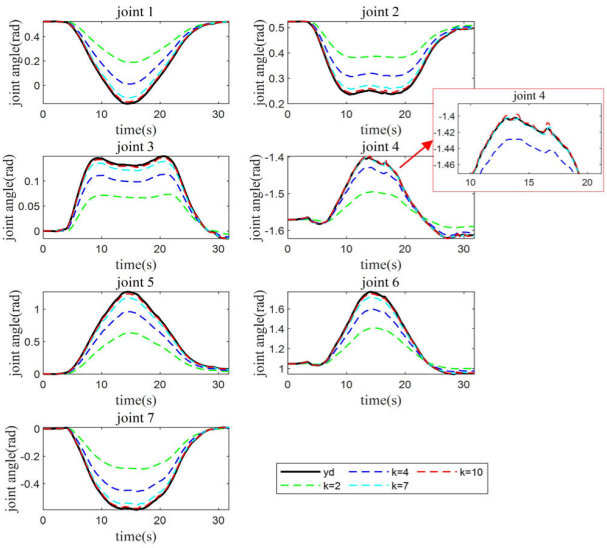
Iteration		k = 5	k = 10	k = 20	k = 30	k = 40
Circular trajectory	j1	0.0663	0.0396	0.0139	**0.0060**	0.0037
	j2	0.1625	0.0907	0.0286	**0.0091**	0.0071
	j3	0.0427	0.0252	0.0088	**0.0033**	0.0013
	j4	0.2795	0.1834	0.0769	**0.0328**	0.0158
	j5	0.0229	0.0140	0.0049	**0.0019**	0.0014
	j6	0.0799	0.0435	0.0136	**0.0042**	0.0040
	j7	0.0719	0.0434	0.0155	**0.0062**	0.0040
Drinking trajectory	j1	0.4228	0.2359	0.0731	0.0230	**0.0073**
	j2	0.1814	0.1016	0.0320	0.0119	**0.0059**
	j3	0.0997	0.0598	0.0215	0.0078	**0.0034**
	j4	0.1223	0.0809	0.0352	0.0169	**0.0088**
	j5	0.8295	0.4838	0.1637	0.0562	**0.0195**
	j6	0.4375	0.2324	0.0659	0.0189	**0.0076**
	j7	0.3915	0.2343	0.0834	0.0320	**0.0134**

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a)



(b)

Fig. 10. Joint angles tracking results for human-like trajectory.

C. Human-Like Trajectory Tracking Control

Finally, the proposed UCHAILC is applied to the human-like trajectory tracking control experiments on rehabilitation robots. In Section II, we have acquired the drinking trajectory in Cartesian space in Fig. 3. Considering the reachable space of the manipulator, we choose an available initial position for the upper limb rehabilitation robot to design the drinking trajectory and then convert it into joint space through inverse kinematics. We adopt the controller parameters and weight factors with the best performance on the robot platform, only the initial control input is changed into $u_1 = [\pi/6, \pi/6, 0, -\pi/2, 0, \pi/3, 0]$ to match the reach of the manipulator.

Fig. 10 exhibits the tracking results of MFAILC and UCHAILC, the black solid lines represent the desired joint angles, which are irregular and contain small perturbations caused by some unconscious small movements of people while acquiring data using a motion capture system. It can be seen from the enlarger image of joint 4 that the black desired trajectory has several irregular bumps, which will increase the difficulty of the tracking control algorithm. Both controllers can obtain satisfying performance as long as the iteration number is sufficient. In the enlarger image of Fig.10(a), the 40th trajectory still doesn't reach the desired trajectory though the error is acceptable, while the 10th trajectory of Fig.10(b) can reach the desired trajectory. The max tracking error rate of the two methods are shown in Table I and II. The UCHAILC can achieve a fast convergence of the maximum tracking error. In addition, it can be seen from the results of two sets of experiments that the error rate of the drinking trajectory tracking is larger than that of the circular trajectory tracking with the same number of iterations. Therefore, irregular trajectory tracking is more difficult than circle trajectory tracking. In conclusion, the proposed UCHAILC can greatly improve the efficiency of iteration and decrease the consumption of time and memory.

V. Discussion and Conclusion

Position tracking control is one of the most important prerequisites for robot-assisted rehabilitation, which can ensure the safety and effectiveness of the rehabilitation process, and avoid secondary injuries to patients caused by excessive tracking errors. In this paper, an uncertainty compensated high-order adaptive iterative learning controller is proposed for the precise tracking control of a 7-DOF upper limb rehabilitation robot. Uncertainty compensation enhances the adaptability of the robot system towards model uncertainty and external disturbances to ensure the system's robustness. Usually, ILC needs numerous runs for accurate trajectory tracking, while the proposed UCHAILC utilizes the high-order learning law to make it converge faster using the data of previous several iterations. Tracking control experiment results of the upper limb rehabilitation robot indicate that the proposed UCHAILC method has a faster convergence rate and can achieve better tracking performance.

The proposed UCHAILC method is a model-free controller, which is convenient to transplant and not restricted by the hardware system. Therefore, this method will have a wide range of application prospects in the nonlinear system control area. Besides, the automatic rehabilitation evaluation based on the patient's training data could be combined into rehabilitation training to formulate a personalized rehabilitation strategy, in which the rehabilitation trajectory could be adjusted according to the patient's recovery conditions, e.g. the appropriate range of motion and training speed. In addition, the surface electromyography (sEMG) signal could be introduced to identify the patient's muscle fatigue state to reasonably adjust the training cycle and improve the recovery effect.

References

[1] Y. Liu, Q. Song, C. Li, X. Guan, and L. Ji, "Quantitative assessment of motor function for patients with a stroke by an end-effector upper limb rehabilitation robot," *BioMed Res. Int.*, vol. 2020, Apr. 2020, Art. no. 5425741.

[2] H. I. Krebs, N. Hogan, M. L. Aisen, and B. T. Volpe, "Robot-aided neurorehabilitation," *IEEE Trans. Rehabil. Eng.*, vol. 6, no. 1, pp. 75–87, Mar. 1998.

[3] R. Bertani, C. Melegari, M. C. De Cola, A. Bramanti, P. Bramanti, and R. S. Calabro, "Effects of robot-assisted upper limb rehabilitation in stroke patients: A systematic review with meta-analysis," *Neurological Sci.*, vol. 38, no. 9, pp. 1561–1569, Sep. 2017.

[4] M. Nann et al., "Restoring activities of daily living using an EEG/EOG-controlled semiautonomous and mobile whole-arm exoskeleton in chronic stroke," *IEEE Syst. J.*, vol. 15, no. 2, pp. 2314–2321, Jun. 2021.

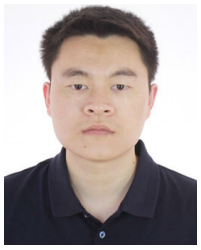
Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[5] Y. Wu, T. J. Tarn, and N. Xi, "Force and transition control with environmental uncertainties," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, May 1995, pp. 899–904.
[6] S. Dong and J. K. Mills, "Adaptive learning control of robotic systems with model uncertainties," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, May 1998, pp. 1847–1852.
[7] M. Sharifi, S. Behzadipour, and G. R. Vossoughi, "Model reference adaptive impedance control of rehabilitation robots in operational space," in *Proc. 4th IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics (BioRob)*, Jun. 2012, pp. 1698–1703.
[8] D. Pont-Esteban, M. Á. Sánchez-Urán, and M. Ferre, "Robust motion control architecture for an upper-limb rehabilitation exosuit," *IEEE Access*, vol. 10, pp. 113631–113648, 2022.
[9] Q. Wu, X. Wang, B. Chen, and H. Wu, "Development of an RBFN-based neural-fuzzy adaptive control strategy for an upper limb rehabilitation exoskeleton," *Mechatronics*, vol. 53, pp. 85–94, Aug. 2018.
[10] F. Afsharnia, A. Madady, and M. B. Menhaj, "Adaptive iterative learning control for unknown linear time-varying continuous systems," *Trans. Inst. Meas. Control*, vol. 42, no. 5, pp. 981–996, Mar. 2020.
[11] H. Zhang, Z. Li, and C.-Y. Su, "Iterative learning control of an exoskeleton for human upper limbs," in *Proc. 11th World Congr. Intell. Control Autom.*, Jun. 2014, pp. 1917–1922.
[12] X. Zhu and J. Wang, "Double iterative compensation learning control for active training of upper limb rehabilitation robot," *Int. J. Control, Autom. Syst.*, vol. 16, no. 3, pp. 1312–1322, Jun. 2018.
[13] K. Maqsood, J. Luo, C. Yang, Q. Ren, and Y. Li, "Iterative learning-based path control for robot-assisted upper-limb rehabilitation," *Neural Comput. Appl.*, vol. 35, no. 32, pp. 23329–23341, Nov. 2023.
[14] Z. Hou, R. Chi, and H. Gao, "An overview of dynamic-linearization-based data-driven control and applications," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4076–4090, May 2017.
[15] Z. Yunjie, C. Rongmin, and Z. Huixing, "Model-free adaptive iterative learning control based on data-driven for noncircular turning tool feed system," in *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems*. Berlin, Germany: Springer, vol. 2016, pp. 3–10.
[16] X. Bu, Q. Yu, Z. Hou, and W. Qian, "Model free adaptive iterative learning consensus tracking control for a class of nonlinear multiagent systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, pp. 677–686, Apr. 2019.
[17] Y. Wang, H. Li, X. Qiu, and X. Xie, "Consensus tracking for nonlinear multi-agent systems with unknown disturbance by using model free adaptive iterative learning control," *Appl. Math. Comput.*, vol. 365, Jan. 2020, Art. no. 124701.
[18] S. Sang, R. Zhang, and X. Lin, "Model-free adaptive iterative learning bipartite containment control for multi-agent systems," *Sensors*, vol. 22, no. 19, p. 7115, Sep. 2022.
[19] W. Meng, S. Q. Xie, Q. Liu, C. Z. Lu, and Q. Ai, "Robust iterative feedback tuning control of a compliant rehabilitation robot for repetitive ankle training," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 173–184, Feb. 2017.
[20] Q. Ai et al., "High-order model-free adaptive iterative learning control of pneumatic artificial muscle with enhanced convergence," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9548–9559, Nov. 2020.
[21] R. Chi, Y. Liu, Z. Hou, and S. Jin, "Data-driven terminal iterative learning control with high-order learning law for a class of non-linear discrete-time multiple-input–multiple output systems," *IET Control Theory Appl.*, vol. 9, no. 7, pp. 1075–1082, Apr. 2015.
[22] Z. Liu et al., "An optimal motion planning method of 7-DOF robotic arm for upper limb movement assistance," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2019, pp. 277–282.
[23] M. Safeea and P. Neto, "KUKA sunrise toolbox: Interfacing collaborative robots with MATLAB," *IEEE Robot. Autom. Mag.*, vol. 26, no. 1, pp. 91–96, Mar. 2019.
[24] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. London, U.K.: Pearson, 2009.
[25] M. Deng, Z. Li, Y. Kang, C. L. P. Chen, and X. Chu, "A learning-based hierarchical control scheme for an exoskeleton robot in human–robot cooperative manipulation," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 112–125, Jan. 2020.
[26] D. Xu, B. Jiang, and F. Liu, "Improved data driven model free adaptive constrained control for a solid oxide fuel cell," *IET Control Theory Appl.*, vol. 10, no. 12, pp. 1412–1419, Aug. 2016.
[27] Z. Hou and S. Jin, "A novel data-driven control approach for a class of discrete-time nonlinear systems," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 6, pp. 1549–1558, Nov. 2011.
[28] Z. Hou and Y. Zhu, "Controller-dynamic-linearization-based model free adaptive control for discrete-time nonlinear systems," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2301–2309, Nov. 2013.
[29] R. Chi, B. Huang, Z. Hou, and S. Jin, "Data-driven high-order terminal iterative learning control with a faster convergence speed," *Int. J. Robust Nonlinear Control*, vol. 28, no. 1, pp. 103–119, Jan. 2018.

Qingsong Ai (Member, IEEE) received the Ph.D. degree in information engineering from the Wuhan University of Technology, Wuhan, China, in 2008. He was a Visiting Researcher with The University of Auckland, New Zealand, from 2006 to 2007, and University of Leeds, U.K. from 2017 to 2018. He is currently a Professor in information engineering with the Wuhan University of Technology and Hubei University, China. He is also the Project Leader of 12 national, ministerial or provincial projects. He has authored more than 70 international journal articles, book chapters, and conference papers.

Zemin Liu received the bachelor's degree in communication engineering from the Wuhan University of Technology, Wuhan, China, in 2018, where he is currently pursuing the Ph.D. degree in information engineering. His current research interests include upper limb rehabilitation robotics and robot motion control.



Wei Meng (Member, IEEE) received the joint Ph.D. degree in information and mechatronics engineering from the Wuhan University of Technology, Wuhan, China, and The University of Auckland, Auckland, New Zealand, in 2016. He was a Research Fellow in robotics with the School of Electronic and Electrical Engineering, University of Leeds, Leeds, U.K. from 2018 to 2020. He is currently an Associate Professor with the School of Information Engineering, Wuhan University of Technology. He has authored four monographs and over 60 peer-reviewed papers and ten patents in rehabilitation robotics and human–robot interaction control.

Quan Liu (Member, IEEE) received the Ph.D. degree in mechanical engineering from the Wuhan University of Technology, Wuhan, China, in 2003. She is currently a Chair Professor in information science with the Wuhan University of Technology. She has authored more than 100 academic papers and books. Her research interests include signal processing, embedded systems, robots, and electronics. She was a recipient of two national awards and three provincial and ministerial awards.

Sheng Q. Xie (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from the University of Canterbury, New Zealand, in 2002. He joined The University of Auckland, New Zealand, in 2003, and became a Chair Professor in (bio)mechatronics in 2011. Since 2017, he has been the Chair of Robotics and Autonomous Systems with the University of Leeds, Leeds, U.K. He has authored or coauthored eight books, 15 book chapters, and over 400 international journals and conference papers. His current research interests include medical and rehabilitation robots, and advanced robot control. He is an elected fellow of the Institution of Professional Engineers New Zealand, a fellow of American Society of Mechanical Engineers (ASME), and a fellow of the Institution of Mechanical Engineers (IMechE).

Authorized licensed use limited to: University of Leeds. Downloaded on February 07,2024 at 20:09:06 UTC from IEEE Xplore. Restrictions apply.