2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Acceleration-Free Recursive Composite Learning Control of High-DoF Robot Manipulators

Yuejiang Zhu[2], Tian Shi[2], Weibing Li[2], and Yongping Pan[1]

***Abstract*— Composite learning robot control (CLRC) is an adaptive control approach that achieves exponential parameter convergence without using a stringent condition termed persistent excitation (PE). For robots with low degrees of freedom (DoFs), a filtered regressor of the robot dynamics needed in CLRC can be calculated analytically without joint accelerations, but this is difficult for high-DoF robots. Under the linear parameterization by the recursive Newton-Euler algorithm, this paper proposes an acceleration-free recursive CLRC (RCLRC) method for high-DoF robots to achieve exponential parameter convergence under a weakened condition termed interval excitation (IE). The proposed method has a low computational cost and avoids undesirable acceleration estimation that seriously affects performance. Simulations and experiments on a 7-DoF robot manipulator have verified the superiority of the proposed RCLRC, where it outperforms its analytical version in both estimation and tracking.**

I. INTRODUCTION

Adaptive control is appealing for realizing superior robot tracking control under parameter uncertainties and external disturbances, and linear parameterization is a crucial assumption for adaptive robot control [1]. Two basic methods for adaptive control are direct and indirect adaptive control [2]. Composite adaptive robot control (CARC) combines joint tracking and torque prediction errors to drive parameter estimation, resulting in faster convergence of both tracking and parameter estimation errors [3]. Joint accelerations are usually inaccessible in real-world robots. To calculate the prediction error, a stable low-pass filter can be applied to the linearly parameterized model to generate a filtered regressor of the robot dynamics. For robots with low degrees of freedom (DoFs), the filtered regressor can be easily obtained by analytically calculating the filtered expression of each element in the regression matrix without joint accelerations. However, this is challenging for high-DoF robots because the term-by-term treatment of all elements in the large-size regressor is tedious and inflexible.

The classical recursive Newton-Euler (RNE) algorithm is appealing to derive the linear parameterization of the robot dynamics due to the convenience of implementation and low computational cost [4]. Recursive CARC (RCARC) using the RNE algorithm provides a computationally efficient way for high-DoF robots without acceleration signals in the filtered regressor [5]–[7]. However, the base parameter model, which is required to obtain the identifiable linear combination of unknown parameters to handle the lack of structural identifiability [8], has not been used in existing RCARC methods. Besides, a stringent condition of persistent excitation (PE) is required in RCARC to achieve parameter convergence.

Composite learning robot control (CLRC) makes use of instantaneous data together with online historical data to drive parameter estimation, which guarantees exponential parameter convergence under a relaxed condition of interval excitation (IE) [9]. CLRC has been extensively studied based on some real-world robots [10]–[18], but most studies consider only the low DoF cases. A preliminary evaluation of CLRC for high-DoF robot manipulators in [13] employs the analytical method for the linear parameterization of the dynamic model, which requires observation methods, such as disturbance observers [19], extended high-gain observers [20], and dirty derivative filters (DDFs) [21], to estimate acceleration signals before regressor filtering, but the estimation inaccuracy under noisy measurement may seriously influence parameter convergence, resulting in degraded tracking performance.

This paper proposes an acceleration-free recursive CLRC (RCLRC) strategy for robotic manipulators with high DoFs. First, the linear parameterization of a generalized dynamic model with an auxiliary variable is implemented via spatial vector algebra [22] to calculate the filtered dynamic model and alleviate the regressor's computational complexity from the orders $O(n^4)$ to $O(n^2)$, where n is the number of DoFs [23]. Second, an automatic recursive method named the recursive parameter nullspace (RPN) algorithm [24] is applied to derive a base parameter model to overcome the lack of structural identifiability. Third, the filtered regressor is recursively calculated without acceleration signals. Finally, an acceleration-free composite learning law is formulated to overcome the negative effect of inaccurate acceleration estimation.

Throughout this paper, $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{R}^n$ and $\mathbb{R}^{N\times N}$ represent the spaces of real numbers, positive real numbers, real n-vectors, and real $N\times N$-matrices, respectively, $\sigma_{\min}(\Theta)\in\mathbb{R}^+$ denotes the minimum singular value (MSV) of $\Theta\in\mathbb{R}^{N\times N}$, $\|\boldsymbol{x}\|$ is the Euclidean norm of $\boldsymbol{x}\in\mathbb{R}^n$, I is an identity matrix with a proper dimension, diag$(x_1, x_2, \cdots, x_n)$ is a diagonal matrix with diagonal elements x_1 to x_n, sign(x_i) is a signum function, $O(n)$ signifies that the upper bound on the growth rate is n, and $\arg\max_{x\in S} f(x) := \{x\in S | f(y)\le f(x), \forall y\in S\}$ with $f:\mathbb{R}\mapsto\mathbb{R}$ and $S\subset\mathbb{R}$, where $\boldsymbol{x}=[x_1, ..., x_n]^{\mathrm{T}}\in\mathbb{R}^n$, $x_i\in\mathbb{R}$ is the ith element of $\boldsymbol{x}$, and n and m are positive integers.

*This work was supported in part by the Guangdong Provincial Pearl River Talents Program, China, under Grant 2019QN01X154, in part by the Fundamental Research Funds for the Central Universities, Sun Yat-sen University, China, under Grant 23lgzy004, in part by the National Natural Science Foundation of China under Grant 62206317, and in part by the Young Talent Support Project of Guangzhou Association for Science and Technology under Grant QT-2023-018 (*Corresponding author: Yongping Pan*).

[1]Y. Pan is with School of Advanced Manufacturing, Sun Yat-sen University, Shenzhen 518100, China panyongp@mail.sysu.edu.cn

[2]Y. Zhu, T. Shi, and W. Li are with School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China zhuyj69@mail2.sysu.edu.cn, shit23@mail2.sysu.edu.cn, liwb53@mail.sysu.edu.cn

II. PRELIMINARIES

In this paper, spatial vectors that combine the angular and linear quantities of link motions and forces [22] are applied to simplify the derivation and programming of the RNE algorithm. Several important definitions are introduced below. A spatial velocity vector of Link i is defined by

$$\boldsymbol{v}_i := \begin{bmatrix} \boldsymbol{\omega}_i \\ \boldsymbol{\nu}_i \end{bmatrix} \in \mathbb{R}^6$$

where $\boldsymbol{\omega}_i \in \mathbb{R}^3$ and $\boldsymbol{\nu}_i \in \mathbb{R}^3$ are angular and linear velocities of Frame i related to Frame $i-1$, represented in Frame i, respectively. A spatial force vector of Link i is defined by

$$\boldsymbol{f}_i := \begin{bmatrix} \boldsymbol{\tau}_{\mathrm{s}i} \\ \boldsymbol{f}_{\mathrm{s}i} \end{bmatrix} \in \mathbb{R}^6$$

where $\boldsymbol{\tau}_{\mathrm{s}i} \in \mathbb{R}^3$ denotes a torque applied to Link i, and $\boldsymbol{f}_{\mathrm{s}i} \in \mathbb{R}^3$ is an applied force. For any vector $\boldsymbol{a} = [a_1, a_2, a_3]^{\mathrm{T}} \in \mathbb{R}^3$, its cross-product operator $\boldsymbol{a}\times$ is defined by

$$\boldsymbol{a}\times := \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

A spatial transformation matrix for transforming the spatial motion vector from Frame $i-1$ to Frame i is defined by

$$X_{i-1}^{i} := \begin{bmatrix} R_{i-1}^{i} & 0 \\ -R_{i-1}^{i}\boldsymbol{r}_{i-1,i}\times & R_{i-1}^{i} \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

where R_{i-1}^{i} denotes the rotation matrix from Frame $i-1$ to Frame i, $\boldsymbol{r}_{i-1,i}$ is the origin position of Frame i relative to that of Frame $i-1$, expressed in Frame $i-1$. The spatial inertia that describes the inertial property of Link i related to the origin of Frame i is defined as follows:

$$J_i = \begin{bmatrix} J_{\mathrm{o}i} & m_i\boldsymbol{c}_i\times \\ -m_i\boldsymbol{c}_i\times & m_i I \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{1}$$

where $m_i \in \mathbb{R}$ is the mass of Link i, $J_{\mathrm{c}i} \in \mathbb{R}^{3\times 3}$ is the inertia tensor related to the center of the mass of Link i, $\boldsymbol{c}_i = [c_{\mathrm{x}i}, c_{\mathrm{y}i}, c_{\mathrm{z}i}]^{\mathrm{T}} \in \mathbb{R}^3$ is a vector pointing from the origin of Frame i to the Link i's center of mass, expressed in Frame i and the inertia tensor related to the local frame, $J_{\mathrm{o}i}$, satisfies $J_{\mathrm{o}i} = J_{\mathrm{c}i} + m_i(\boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{c}_i I - \boldsymbol{c}_i\boldsymbol{c}_i^{\mathrm{T}})$, and its expanded form is defined by

$$J_{\mathrm{o}i} = \begin{bmatrix} J_{\mathrm{xx}i} & J_{\mathrm{xy}i} & J_{\mathrm{xz}i} \\ J_{\mathrm{xy}i} & J_{\mathrm{yy}i} & J_{\mathrm{yz}i} \\ J_{\mathrm{xz}i} & J_{\mathrm{yz}i} & J_{\mathrm{zz}i} \end{bmatrix}.$$

The RNE algorithm in a spatial vector form is given by [22]

$$\text{Forward:} \begin{cases} \boldsymbol{v}_0 = 0, \boldsymbol{a}_0 = -\boldsymbol{g}_0 \\ \boldsymbol{v}_i = X_{i-1}^{i}\boldsymbol{v}_{i-1} + \dot{q}_i\boldsymbol{s}_i \\ \boldsymbol{a}_i = X_{i-1}^{i}\boldsymbol{a}_{i-1} + \ddot{q}_i\boldsymbol{s}_i + \boldsymbol{v}_i \times \dot{q}_i\boldsymbol{s}_i \end{cases}$$

with $i = 1, 2, ..., n$, $\boldsymbol{g}_0 = [0, 0, 0, 0, 0, 9.81]^{\mathrm{T}}$ and

$$\text{Backward:} \begin{cases} \boldsymbol{f}_i^{B} = J_i\boldsymbol{a}_i + [\boldsymbol{v}_i\times]^{*}J_i\boldsymbol{v}_i \\ \boldsymbol{f}_i = \boldsymbol{f}_i^{B} + [X_{i+1}^{i}]^{*}\boldsymbol{f}_{i+1} \\ \tau_i = \boldsymbol{s}_i^{\mathrm{T}}\boldsymbol{f}_i \end{cases}$$

with $i = n, n-1, ..., 1$, $\boldsymbol{f}_{n+1} = \boldsymbol{0}$, where $\boldsymbol{s}_i \in \mathbb{R}^6$ is an axis vector determined by the joint type, $\tau_i \in \mathbb{R}$ is the ith joint torque, and $\boldsymbol{a}_i \in \mathbb{R}^6$ is an spatial acceleration vector of Link i, defined by $\boldsymbol{a}_i := \dot{\boldsymbol{v}}_i$, and $\boldsymbol{v}_i\times$ is a cross-product operator for the spatial velocity $\boldsymbol{v}_i$ defined by

$$\boldsymbol{v}_i\times := \begin{bmatrix} \boldsymbol{\omega}_i\times & 0 \\ \boldsymbol{\nu}_i\times & \boldsymbol{\omega}_i\times \end{bmatrix}.$$

Note that $[\boldsymbol{v}_i\times]^{*} = -[\boldsymbol{v}_i\times]^{\mathrm{T}}$ and $[X_{i+1}^{i}]^{*} = [X_{i}^{i+1}]^{\mathrm{T}}$.

The following definitions are also introduced for the convenience of control analysis [9].

Definition 1: A bounded signal $\Phi(t) \in \mathbb{R}^{n\times N}$ is of IE if $\exists T_{\mathrm{e}}, \tau_{\mathrm{d}}, \sigma \in \mathbb{R}^{+}$ such that $\int_{T_{\mathrm{e}}-\tau_{\mathrm{d}}}^{T_{\mathrm{e}}} \Phi^{\mathrm{T}}(\tau)\Phi(\tau)d\tau \geq \sigma I$.

Definition 2: A bounded signal $\Phi(t) \in \mathbb{R}^{n\times N}$ is of PE if $\exists \tau_{\mathrm{d}}, \sigma \in \mathbb{R}^{+}$ such that $\int_{t-\tau_{\mathrm{d}}}^{t} \Phi^{\mathrm{T}}(\tau)\Phi(\tau)d\tau \geq \sigma I, \forall t \geq 0$.

III. PROBLEM FORMULATION

Consider an n-DoF serial manipulator with revolute joints. In this study, the kinematics is derived by the modified Denavit-Hartenberg (DH) convention, and the dynamics is given by an Euler-Lagrange formulation as follows:

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) + F(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{2}$$

in which $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ denote the joint angular position, velocity, and acceleration, respectively, $M \in \mathbb{R}^{n\times n}$ denotes an inertia matrix, $C \in \mathbb{R}^{n\times n}$ is a centripetal-Coriolis matrix, $G \in \mathbb{R}^n$ is a gravity torque, $\boldsymbol{\tau} \in \mathbb{R}^n$ is an input torque, and $F \in \mathbb{R}^n$ is a friction torque given by [25]

$$F_i(\dot{q}_i) = f_{\mathrm{v}i}\dot{q}_i + f_{\mathrm{c}i}\mathrm{sign}(\dot{q}_i) + f_{\mathrm{o}i}$$

where $f_{\mathrm{v}i}, f_{\mathrm{c}i}, f_{\mathrm{o}i} \in \mathbb{R}^{+}$ denote coefficients of viscous friction, Coulomb friction, and Coulomb friction offset of the ith joint for $i = 1$ to n, respectively. Then, the linearly parameterized model of the friction torque can be written as follows:

$$F(\dot{\boldsymbol{q}}) = \Phi_{\mathrm{fric}}(\dot{\boldsymbol{q}})W_{\mathrm{fric}}$$

with a friction regressor $\Phi_{\mathrm{fric}} := [\mathrm{diag}(\dot{\boldsymbol{q}}), \mathrm{diag}(\mathrm{sign}(\dot{\boldsymbol{q}})), I] \in \mathbb{R}^{n\times 3n}$ and a friction parameter vector $W_{\mathrm{fric}} := [\boldsymbol{f}_{\mathrm{v}}^{\mathrm{T}}, \boldsymbol{f}_{\mathrm{c}}^{\mathrm{T}}, \boldsymbol{f}_{\mathrm{o}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3n}$, in which $\boldsymbol{f}_{\mathrm{v}} = [f_{\mathrm{v}1}, f_{\mathrm{v}2}, \cdots, f_{\mathrm{v}n}]^{\mathrm{T}}$, $\boldsymbol{f}_{\mathrm{c}} = [f_{\mathrm{c}1}, f_{\mathrm{c}2}, \cdots, f_{\mathrm{c}n}]^{\mathrm{T}}$, and $\boldsymbol{f}_{\mathrm{o}} = [f_{\mathrm{o}1}, f_{\mathrm{o}2}, \cdots, f_{\mathrm{o}n}]^{\mathrm{T}}$. Then, define the left-hand side of (2) in a generalized form

$$H(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) = M(\boldsymbol{q})\dot{\boldsymbol{\zeta}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\zeta} + G(\boldsymbol{q}) + F(\dot{\boldsymbol{q}}) \tag{3}$$

with $\boldsymbol{\zeta} \in \mathbb{R}^n$ being an auxiliary variable.

Property 1: $M(\boldsymbol{q})$ is symmetric positive-definite and satisfies $m_0 I_n \leq M(\boldsymbol{q}) \leq m_1 I_n$ with constants $m_0, m_1 \in \mathbb{R}^{+}$.

Property 2: $\dot{M}(\boldsymbol{q}) - 2C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is skew-symmetric such that $\boldsymbol{x}^{\mathrm{T}}(\dot{M}(\boldsymbol{q}) - 2C(\boldsymbol{q}, \dot{\boldsymbol{q}}))\boldsymbol{x} = 0, \forall \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{x} \in \mathbb{R}^n$.

Property 3: $H(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})$ can be linearly parameterized by

$$H(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) = \Phi_{\mathrm{full}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})W_{\mathrm{full}} \tag{4}$$

with a full regressor $\Phi_{\mathrm{full}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) := [\Phi_{\mathrm{link,full}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}), \Phi_{\mathrm{fric}}(\dot{\boldsymbol{q}})] \in \mathbb{R}^{n\times 13n}$ and a full parameter vector $W_{\mathrm{full}} := [W_{\mathrm{link,full}}^{\mathrm{T}}, W_{\mathrm{fric}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{13n}$, in which $W_{\mathrm{link,full}} \in \mathbb{R}^{10n}$ is a link parameter vector stacked downward by the n columns of the link's dynamic parameters in order, namely, $W_{\rm link,full} = [W_1^{\rm T}, W_2^{\rm T}, \cdots, W_n^{\rm T}]^{\rm T}$, where for $i = 1$ to n, one has

$$W_i := [m_i, m_i c_{{\rm x}i}, m_i c_{{\rm y}i}, m_i c_{{\rm z}i}, J_{{\rm xx}i}, J_{{\rm yy}i}, J_{{\rm zz}i}, J_{{\rm xy}i}, J_{{\rm xz}i}, J_{{\rm yz}i}]^{\rm T}.$$

The terms $\Phi_{\rm link,full}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})$ and $W_{\rm link,full}$ in (4) can be transformed into compact forms

$$\Phi_{\rm link,base}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) = \Phi_{\rm link,full}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) P_{\rm m},$$
$$W_{\rm link,base} = P_{\rm p} W_{\rm link,full}$$

in which $W_{\rm link,base} \in \mathbb{R}^m$ and $\Phi_{\rm link,base}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) \in \mathbb{R}^{n\times m}$ are a link base parameter vector and its regressor, respectively, $P_{\rm m} \in \mathbb{R}^{10n\times m}$ and $P_{\rm p} \in \mathbb{R}^{m\times 10n}$ can be determined by the robot's modified DH parameters, and m is the number of identifiable combinations. Then, the parameterized form (4) can be transformed into a compact form

$$H(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) = \Phi(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) W \tag{5}$$

in which $W := [W_{\rm link,base}^{\rm T}, W_{\rm fric}^{\rm T}]^{\rm T} \in \mathbb{R}^N$ is an identifiable parameter vector, and $\Phi(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) := [\Phi_{\rm link,base}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}), \Phi_{\rm fric}(\dot{\boldsymbol{q}})] \in \mathbb{R}^{n\times N}$ is its regressor with $N = m + 3n$.

The problem of RCLRC design applied for high-DoF robot manipulators is as follows: Given the desired trajectories $\boldsymbol{q}_{\rm d}(t)$, $\dot{\boldsymbol{q}}_{\rm d}(t)$, $\ddot{\boldsymbol{q}}_{\rm d}(t) \in \mathbb{R}^n$, the measurement of the joint position $\boldsymbol{q}$, and the estimation of the joint velocity $\dot{\boldsymbol{q}}$ under known kinematic parameters and unknown dynamic parameters, the CLRC control law $\boldsymbol{\tau}$ is implemented recursively, and a composite learning law for the parameter estimate $\hat{W} \in \mathbb{R}^N$ is derived recursively without resorting to the acceleration signal $\ddot{\boldsymbol{q}}$, such that the position tracking error $\boldsymbol{e}(t) := \boldsymbol{q}_{\rm d}(t) - \boldsymbol{q}(t)$ and the parameter estimation error $\tilde{W}(t) := W - \hat{W}(t)$ exponentially converge to zero under the relaxed IE condition.

IV. Closed-Form Composite Learning Control

Consider the robot parameterized model (2) with (5) under Properties 1-3 ($\boldsymbol{\zeta} = \dot{\boldsymbol{q}}$). Using a stable filter $L(s) := \frac{\alpha}{s+\alpha}$, one gets a filtered parameterized model

$$\boldsymbol{\tau}_{\rm f}(t) = \Phi_{\rm f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) W \tag{6}$$

with $\boldsymbol{\tau}_{\rm f} := L(s)[\boldsymbol{\tau}]$ and $\Phi_{\rm f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) := L(s)[\Phi(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})]$, in which $\alpha \in \mathbb{R}^+$ is a filtering constant, and s is the complex Laplace operator. Define an excitation matrix

$$\Theta(t) := \int_{t-\tau_d}^{t} \Phi_{\rm f}^{\rm T}(\tau) \Phi_{\rm f}(\tau) d\tau \tag{7}$$

where $\tau_{\rm d} \in \mathbb{R}^+$ is the length of integration duration. In CARC, a torque prediction error is defined by

$$\boldsymbol{\varepsilon}(t) := \boldsymbol{\tau}_{\rm f}(t) - \Phi_{\rm f}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \hat{W}(t) \tag{8}$$

and a generalized prediction error is defined by

$$\boldsymbol{\xi}(t) = \begin{cases} \Theta(t_{\rm e}) W - \Theta(t_{\rm e}) \hat{W}(t), & t \geq T_e \\ \Theta(t) W - \Theta(t) \hat{W}(t), & \text{otherwise} \end{cases} \tag{9}$$

with $t_{\rm e} := \arg\max_{\zeta\in[T_{\rm e},t]} \sigma_{\min}(\Theta(\zeta))$ and $\Theta(T_{\rm e}) \geq \sigma I$.

Let $\boldsymbol{e}_{\rm s}(t) := \dot{\boldsymbol{e}}(t) + \Lambda \boldsymbol{e}(t)$ denote a sliding tracking error, where $\Lambda \in \mathbb{R}^{n\times n}$ is a positive-definite diagonal matrix. The composite adaptive control law is given by [9]

$$\boldsymbol{\tau} = K_{\rm c} \boldsymbol{e}_{\rm s} + \Phi(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_{\rm r}, \ddot{\boldsymbol{q}}_{\rm r}) \hat{W} \tag{10}$$

where the parameter estimate $\hat{W}$ is updated by

$$\dot{\hat{W}} = \Gamma(\Phi^{\rm T}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_{\rm r}, \ddot{\boldsymbol{q}}_{\rm r}) \boldsymbol{e}_{\rm s} + \kappa \boldsymbol{\xi}) \tag{11}$$

in which $\dot{\boldsymbol{q}}_{\rm r}(t) := \dot{\boldsymbol{q}}_{\rm d}(t) + \Lambda \boldsymbol{e}(t)$ is a joint "reference velocity", $\Gamma \in \mathbb{R}^{N\times N}$ is a positive-definite diagonal matrix of learning rates, and $\kappa \in \mathbb{R}$ is a weighting factor.

For low-DoF robot manipulators, the filtered regressor $\Phi_{\rm f}$ in (6) can be easily obtained by using analytical expressions such that the usage of the acceleration signal $\ddot{\boldsymbol{q}}$ can be avoided [2]. However, for high-DoF robot manipulators, it is difficult to calculate $\Phi_{\rm f}$ without resorting to $\ddot{\boldsymbol{q}}$ because the analytical formula of the regressor Φ in (5) may be not available. Even when the analytical formula of Φ is available, it may contain hundreds of elements such that analytically calculating their filtered expressions is tedious and inflexible.

V. Recursive-Form Composite Learning Control

The central task to derive RCLRC is the recursive implementation of two regressors: 1) $\Phi(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})$ that appears in both the control law (10) and the parameter estimation law (11); 2) $\Phi_{\rm f}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ that only appears in the parameter estimation law (11). The recursive implementation of $\Phi(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})$ involves recursive linear parameterization which lays the foundation to the recursive implementation of $\Phi_{\rm f}(\boldsymbol{q}, \dot{\boldsymbol{q}})$.

Define W_i^l as the lth parameter in the parameter vector W_i of Link i. The spatial inertia matrix J_i of Link i can be written as a linear combination of constant displacement matrices $R_l \in \mathbb{R}^6$ by taking parameters as coefficients:

$$J_i = \sum\nolimits_{l=1}^{10} W_i^l R_l \tag{12}$$

where R_l is constituted by Boolean bits with a sign, denoting the placement and sign of the lth parameter in the expression of J_i in (1). The dynamic model (3) formulated by the spatial vector-based RNE algorithm is given as follows [5]:

$$\text{Forward:} \begin{cases} \boldsymbol{v}_0 = \boldsymbol{0}, \boldsymbol{u}_0 = \boldsymbol{0}, \dot{\boldsymbol{u}}_0 = -\boldsymbol{g}_0 \\ \boldsymbol{v}_i = X_{i-1}^i \boldsymbol{v}_{i-1} + \boldsymbol{\zeta} \boldsymbol{s}_i \\ \boldsymbol{u}_i = X_{i-1}^i \boldsymbol{u}_{i-1} + \boldsymbol{\zeta}_i \boldsymbol{s}_i \\ \dot{\boldsymbol{u}}_i = X_{i-1}^i \dot{\boldsymbol{u}}_{i-1} + \dot{\boldsymbol{\zeta}}_i \boldsymbol{s}_i + \boldsymbol{v}_i \times \boldsymbol{\zeta}_i \boldsymbol{s}_i \end{cases} \tag{13}$$

with $i = 1, 2, ..., n$, $\boldsymbol{g}_0 = [0, 0, 0, 0, 0, 9.81]^{\rm T}$ and

$$\text{Backward:} \begin{cases} \boldsymbol{f}_i^B = J_i \dot{\boldsymbol{u}}_i + [\boldsymbol{v}_i \times]^* J_i \boldsymbol{u}_i \\ \boldsymbol{f}_i = \boldsymbol{f}_i^B + [X_{i+1}^i]^* \boldsymbol{f}_{i+1} \\ F_i(\dot{\boldsymbol{q}}) = f_{{\rm v}i} \dot{q}_i + f_{{\rm c}i} \text{sign}(\dot{q}_i) + f_{{\rm o}i} \\ \tau_i = \boldsymbol{s}_i^{\rm T} \boldsymbol{f}_i + F_i(\dot{\boldsymbol{q}}) \end{cases} \tag{14}$$

with $i = n, n-1, ..., 1$, $\boldsymbol{f}_{n+1} = \boldsymbol{0}$, where $\boldsymbol{u}_i \in \mathbb{R}^6$ denotes a generalized spatial velocity for the auxiliary variable $\boldsymbol{\zeta}$, and $\dot{q}_i$ and $F_i(\dot{\boldsymbol{q}})$ are the ith elements of $\dot{\boldsymbol{q}}$ and $F(\dot{\boldsymbol{q}})$, respectively. With $\boldsymbol{\zeta} = \dot{\boldsymbol{q}}$, the control torque τ_i of Joint i satisfies

$$\tau_i - F_i(\dot{\boldsymbol{q}}) = \sum\nolimits_{j=i}^{n} \boldsymbol{s}_i^{\rm T} [X_j^i]^* \boldsymbol{f}_j^B. \tag{15}$$

For the convenience of analysis, the robot dynamics can be rewritten into a matrix form

$$\begin{bmatrix}\tau_1 - F_1(\dot{\boldsymbol{q}})\\ \tau_2 - F_2(\dot{\boldsymbol{q}})\\ \vdots \\ \tau_n - F_n(\dot{\boldsymbol{q}})\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{s}_1^{\mathrm{T}} & \boldsymbol{s}_1^{\mathrm{T}}[X_2^1]^* & \cdots & \boldsymbol{s}_1^{\mathrm{T}}[X_n^1]^*\\ 0 & \boldsymbol{s}_2^{\mathrm{T}} & \cdots & \boldsymbol{s}_2^{\mathrm{T}}[X_n^2]^*\\ \vdots & \vdots & \cdots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{s}_n^{\mathrm{T}}\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}\boldsymbol{f}_1^B\\ \boldsymbol{f}_2^B\\ \vdots\\ \boldsymbol{f}_n^B\end{bmatrix}}_{B}$$

with $A \in \mathbb{R}^{n\times 6n}$ and $B \in \mathbb{R}^{6n}$. Substituting (12) into the expression of $\boldsymbol{f}_i^B$ in (14), one obtains

$$\boldsymbol{f}_i^B = \sum\nolimits_{l=1}^{10} W_i^l \underbrace{(R_l\dot{\boldsymbol{u}}_i + [\boldsymbol{v}_i\times]^* R_l\boldsymbol{u}_i)}_{\boldsymbol{f}_{i,l}} \tag{16}$$

where $\boldsymbol{f}_{i,l} \in \mathbb{R}^6$ is a force wrench related to l-th parameter. Then, $\boldsymbol{f}_i^B$ can be rewritten into a matrix form

$$\boldsymbol{f}_i^B = \underbrace{[\boldsymbol{f}_{i,1}, \boldsymbol{f}_{i,2}, \boldsymbol{f}_{i,3}, \cdots, \boldsymbol{f}_{i,10}]}_{F_i}\underbrace{\begin{bmatrix}W_i^1\\ W_i^2\\ \vdots\\ W_i^{10}\end{bmatrix}}_{W_i} \tag{17}$$

with $F_i \in \mathbb{R}^{6\times 10}$. Then, B can be rewritten into

$$B = \underbrace{\begin{bmatrix}F_1 & & & \\ & F_2 & & \\ & & \ddots & \\ & & & F_n\end{bmatrix}}_{H}\underbrace{\begin{bmatrix}W_1\\ W_2\\ \vdots\\ W_n\end{bmatrix}}_{W_{\mathrm{link,full}}} \tag{18}$$

with $H \in \mathbb{R}^{6n\times 10n}$. The parameterized model is obtained by

$$\begin{aligned}\boldsymbol{\tau} &= AHW_{\mathrm{link,full}} + \Phi_{\mathrm{fric}}(\dot{\boldsymbol{q}})W_{\mathrm{fric}}\\ &= AHP_{\mathrm{m}}W_{\mathrm{link,base}} + \Phi_{\mathrm{fric}}(\dot{\boldsymbol{q}})W_{\mathrm{fric}}.\end{aligned} \tag{19}$$

The RCLRC employs the RPN algorithm proposed in [24] to calculate the projector matrices P_{m} and P_{p} for transforming to the base parameter form. After the formulation of the recursive linear parameterization for the robot dynamic model, it is possible to calculate the filtered robot dynamics recursively to avoid using $\ddot{\boldsymbol{q}}$. The derivation process is restated here for self-consistency with additional projection to the forms of the base parameters and the consideration of the friction torque. It is worth noting that linear filtering to an input is essentially the convolution of the input with the impulse response of the linear filter in the time domain. With Property 2, the rewritten filtered dynamic model (2) is given by [5]

$$\begin{aligned}\boldsymbol{\tau}_{\mathrm{f}}(t) &= w(0)M(\boldsymbol{q})\dot{\boldsymbol{q}} - w(t)M(\boldsymbol{q}(0))\dot{\boldsymbol{q}}(0)\\ &+L(s)[-\alpha M(\boldsymbol{q})\dot{\boldsymbol{q}} - C^{\mathrm{T}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + F(\boldsymbol{q})].\end{aligned} \tag{20}$$

Linear parameterization results of $M(\boldsymbol{q})\dot{\boldsymbol{q}}$ and $C^{\mathrm{T}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ enable the calculation of Φ_{f} recursively by using their spatial vector-based expressions derived in [5] as follows:

$$[M(\boldsymbol{q})\dot{\boldsymbol{q}}]_i = \sum\nolimits_{j=i}^{n} \boldsymbol{s}_i^{\mathrm{T}}[X_j^i]^* J_j \boldsymbol{v}_j, \tag{21}$$

$$[C^{\mathrm{T}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}]_i = \sum\nolimits_{j=i}^{n} (\boldsymbol{s}_i\boldsymbol{v}_i\times)^{\mathrm{T}}[X_j^i]^* J_j \boldsymbol{v}_j. \tag{22}$$

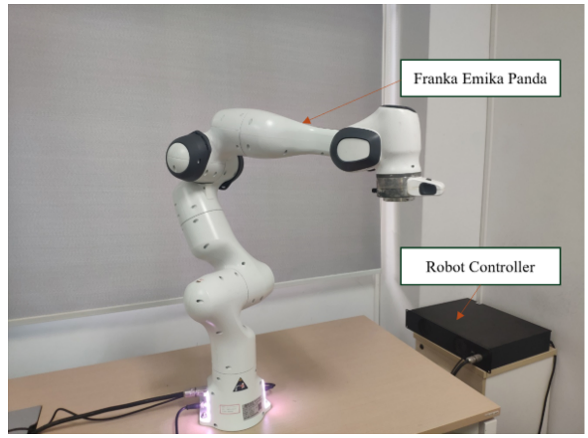

Fig. 1. A 7-DoF collaborative robot named Panda from Framka Emika Inc. for experimental studies.

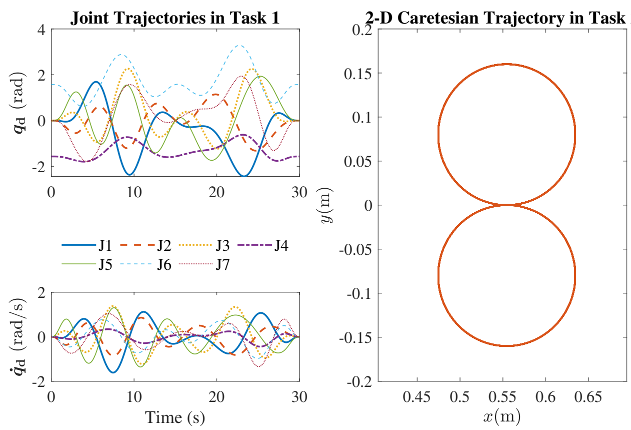

Fig. 2. The desired trajectories for two control tasks. Note that Task 1 is used for evaluation in the transient stage at $t \in [0, 30]$, and Task 2 is a trajectory consisting of five repeating ∞-like trajectories on the X-Y plane of Frame 0 at $t \in [100, 150]$ for evaluation in the steady stage. The first three repetitions of the ∞-like trajectories are at low speeds, while the last two are at high speeds. The period between the two tasks is used to wait for parameter convergence.

In a manner similar to derivating the recursive linear parameterization for (3) [as in (15)-(19)], one can get $\Phi_1(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times 10n}$ and $\Phi_2(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times 10n}$, the regressors of $M(\boldsymbol{q})\dot{\boldsymbol{q}}$ and $C^{\mathrm{T}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$, respectively, which satisfy

$$M(\boldsymbol{q})\dot{\boldsymbol{q}} = \Phi_1(\boldsymbol{q},\dot{\boldsymbol{q}})P_{\mathrm{m}}W, \tag{23}$$

$$C^{\mathrm{T}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \Phi_2(\boldsymbol{q},\dot{\boldsymbol{q}})P_{\mathrm{m}}W. \tag{24}$$

Then, the recursive form of (20) is given by

$$\begin{aligned}\boldsymbol{\tau}_{\mathrm{f}}(t) &= [w(0)\Phi_1(\boldsymbol{q},\dot{\boldsymbol{q}}) - w(t)\Phi_1(\boldsymbol{q}(0),\dot{\boldsymbol{q}}(0))]P_{\mathrm{m}}W_{\mathrm{link,base}}\\ &+ L(s)[-\alpha\Phi_1(\boldsymbol{q},\dot{\boldsymbol{q}}) - \Phi_2(\boldsymbol{q},\dot{\boldsymbol{q}})]P_{\mathrm{m}}W_{\mathrm{link,base}}\\ &+ L(s)[\Phi_{\mathrm{fric}}(\dot{\boldsymbol{q}})]W_{\mathrm{fric}}.\end{aligned}$$

In summary, the filtered regressor Φ_f with the consideration of friction can be recursively computed by

$$\Phi_{\mathrm{f}}(\boldsymbol{q},\dot{\boldsymbol{q}}) = [\Phi_{\mathrm{f,link}}, L(s)[\Phi_{\mathrm{fric}}(\dot{\boldsymbol{q}})]], \tag{25}$$

$$\begin{aligned}\Phi_{\mathrm{f,link}} &= w(t-r)\Phi_1(\boldsymbol{q}(r),\dot{\boldsymbol{q}}(r))|_0^t P_{\mathrm{m}}\\ &+ L(s)[-\alpha\Phi_1(\boldsymbol{q},\dot{\boldsymbol{q}}) - \Phi_2(\boldsymbol{q},\dot{\boldsymbol{q}})]P_{\mathrm{m}}.\end{aligned} \tag{26}$$

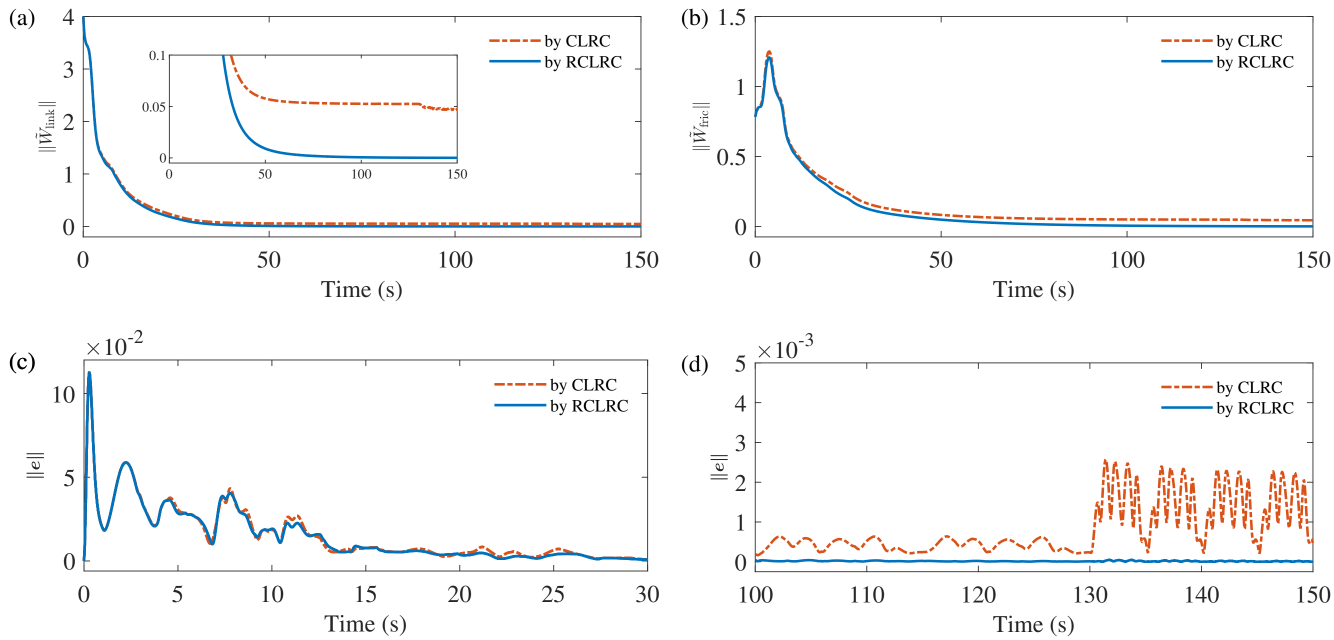

Fig. 3. Performance comparisons of two controllers in simulations. (a) The norms of the link parameter estimation error $\tilde{W}_{\text{link}}$. (b) The norms of the friction parameter estimation error $\tilde{W}_{\text{fric}}$. (c) The norms of the tracking error $\boldsymbol{e}$ in Task 1. (d) The norms of the tracking error $\boldsymbol{e}$ in Task 2.

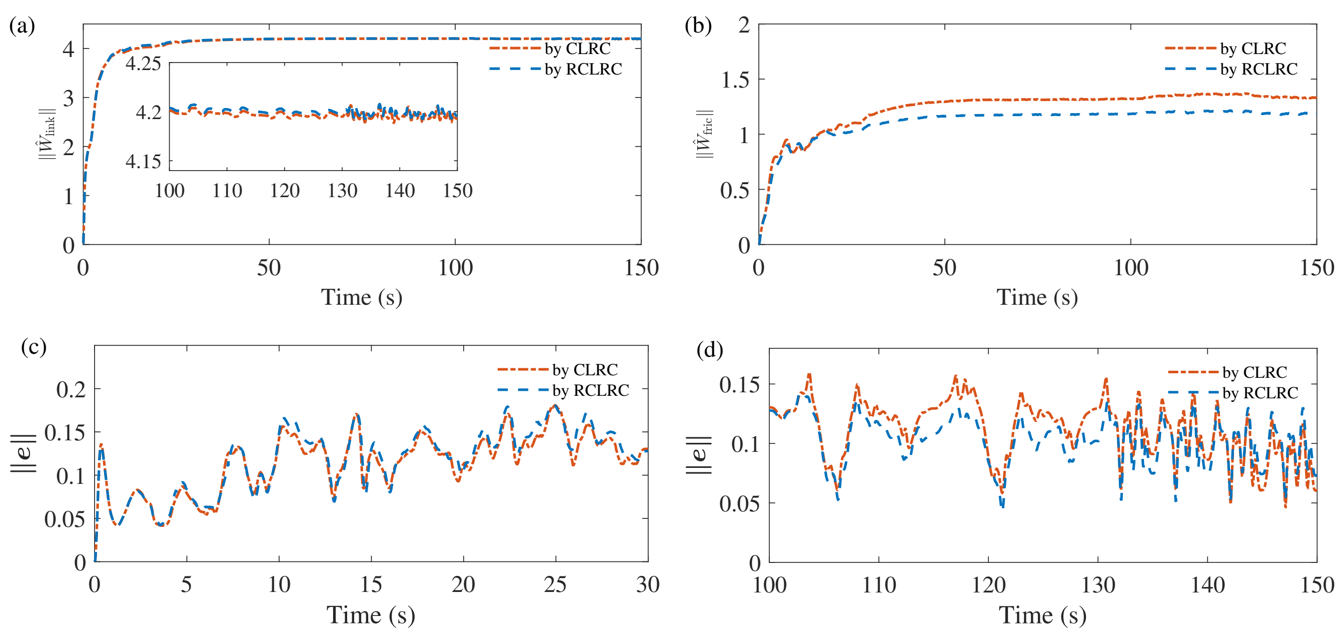

Fig. 4. Performance comparisons of two controllers in experiments. (a) The norms of the link parameter estimation $\hat{W}_{\text{link}}$. (b) The norms of the friction parameter estimation $\hat{W}_{\text{fric}}$. (c) The norms of the tracking error $\boldsymbol{e}$ in Task 1. (d) The norms of the tracking error $\boldsymbol{e}$ in Task 2

VI. Simulation Results

A collaborative robot with 7-DoFs named Franka Emika Panda in Fig. 1 is simulated in MATLAB software to verify the proposed RCLRC, where the solver is set as fixed-step ode2 with a step size of 0.2 ms. The parameter vector W_{full} of the dynamics (2) is from the identified result in [25]. Set the control parameters α =5, τ_{d} =30 s, $\Lambda = 10I$, Γ =0.2I, $\kappa =$ 1, and $\hat{W}(0) = \mathbf{0} \in \mathbb{R}^{64}$. The control gain K_{c} is set to be

$$K_{\text{c},i} = \begin{cases} k_i^* e^{0.1\ln(0.1)t}, & t \le 10 \\ 0.1 k_i^*, & t > 10 \end{cases}$$

with $i = 1, 2, \cdots, 7$ and $k^* = [20, 20, 20, 20, 8, 8, 5]$, where $K_{\text{c},i}$ is the ith diagonal element of K_{c}. The CLRC in [9] is selected as a baseline, where the shared parameters of the two controllers are set to be the same values for fair comparisons.

It is assumed that $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are measurable, but $\ddot{\boldsymbol{q}}$ in CLRC is estimated by a DDF: $\ddot{\boldsymbol{q}} \approx \frac{\lambda_{\rm d} s}{s+\lambda_{\rm d}}[\dot{\boldsymbol{q}}]$ with $\lambda_{\rm d} = 50$. The desired output $\boldsymbol{q}_{\rm d}$ with two tasks for simulations are shown in Fig. 2. Task 1 is a tracking task for achieving parameter convergence, and Task 2 is a period movement with low and high speeds for verifying the accuracy of parameter estimation.

Simulation results are shown in Fig. 3, where the CLRC does not achieve the convergence of the estimation errors $\tilde{W}_{\rm link}$ and $\tilde{W}_{\rm fric}$ to $\mathbf{0}$ [see Fig. 3(a)-(b)] as it utilizes the estimated acceleration $\ddot{\boldsymbol{q}}$. In contrast, the proposed RCLRC exhibits the rapid convergence of $\tilde{W}_{\rm link}$ and $\tilde{W}_{\rm fric}$ to $\mathbf{0}$ after 30 s [see Fig. 3(a)-(b)], and its tracking performance is better in Task 2 [see Fig. 3(d)], which implies that: 1) The exponential stability and parameter convergence by the RCLRC are guaranteed under IE, significantly relaxing PE; 2) the acceleration-free filtered regressor $\Phi_{\rm f}$ is beneficial for parameter estimation.

VII. Experimental Results

An experimental platform of the Panda robot is shown in Fig. 1, where control algorithms run on a personal computer with AMD Ryzen 7 3700X CPU and Ubuntu 20.04 operating system. Both the controllers have the same parameter values as those in the simulations except $\Gamma = \text{diag}(0.1, 0.1, \cdots, 0.1)$ and $\lambda_{\rm d} = 8$. The sampling time is set as 1 ms, and the control system works smoothly since the computational times for both the controllers are within 1 ms. Experimental results are exhibited in Fig. 4, where the norms of the parameter estimates $\hat{W}_{\rm link}$ and $\hat{W}_{\rm fric}$ converge to certain constants after 30 s for both the controllers [see Figs. 4(a)-(b)]. However, the proposed RCLRC performs superior trajectory tracking compared with the CLRC in Task 2 [see Fig. 4(d)], which verifies that the RCLRC achieves the better estimation of $W_{\rm link}$ and $W_{\rm fric}$. The little difference in tracking accuracy between the two controllers in Task 1 [see Fig. 4(c)] is due to the balance between tracking and estimation speeds during the process of parameter convergence.

VIII. Conclusion

In this paper, we have presented an acceleration-free RCLRC method with guaranteed parameter convergence under the IE condition. The novelty of the proposed method lies in that the spatial vector-based RNE algorithm is utilized for robot dynamics modeling, resulting in the acceleration-free calculation of the filtered regressor. Simulations and experiments on a 7-DoF robot have validated that the proposed method outperforms the acceleration-based CLRC method in parameter estimation and trajectory tracking. We would enhance the proposed algorithm with more technical analysis and provide more comprehensive simulations and experiments in further studies.

References

[1] M. Benosman, "Model-based vs data-driven adaptive control: An overview," *Int J. Adapt. Control Signal Process.*, vol. 32, no. 5, pp. 753–776, Mar. 2018.

[2] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice hall, 1991.

[3] J. J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–19, Jul. 1989.

[4] W. Khalil, "Dynamic modeling of robots using Newton-Euler formulation," in *Informatics in Control, Automation and Robotics*. Berlin, Heidelberg, Germany: Springer, 2011, pp. 3–20.

[5] H. Wang, "Recursive composite adaptation for robot manipulators," *J. Dyn. Sys., Meas., Control*, vol. 135, no. 2, Mar. 2013.

[6] Y. Wang and Y. Xie, "Recursive adaptive control of Y-shape space robot," in *Proc. Chinese Control Conf.*, Guangzhou, China, 2019, pp. 2570–2575.

[7] H. Wang and Y. Xie, "On the recursive adaptive control for free-floating space manipulators," *J. Intell. Robot. Syst.*, vol. 66, no. 4, pp. 443–461, Mar. 2012.

[8] R. Bellman and K. J. Strm, "On structural identifiability," *Math. Biosci.*, vol. 7, no. 3–4, pp. 329–339, Apr. 1970.

[9] Y. Pan and H. Yu, "Composite learning robot control with guaranteed parameter convergence," *Automatica*, vol. 89, pp. 398–406, Mar. 2018.

[10] K. Guo, Y. Pan, and H. Yu, "Composite learning robot control with friction compensation: A neural network-based approach," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7841–7851, Oct. 2019.

[11] K. Guo, Y. Pan, D. Zheng, and H. Yu, "Composite learning control of robotic systems: A least squares modulated approach," *Automatica*, vol. 111, p. 108612, Jan. 2020.

[12] D. Huang, C. Yang, Y. Pan, and L. Cheng, "Composite learning enhanced neural control for robot manipulator with output error constraints," *IEEE Trans. Ind. Inform.*, vol. 17, no. 1, pp. 209–218, Dec. 2021.

[13] X. Liu, Z. Li, and Y. Pan, "Preliminary evaluation of composite learning tracking control on 7-DoF collaborative robots," *IFAC-PapersOnLine*, vol. 54, no. 14, pp. 470–475, Nov. 2021.

[14] X. Liu, Z. Li, and Y. Pan, "Experiments of composite learning admittance control on 7-DoF collaborative robots," in *Proc. Int. Conf. Intell. Robot. Appl.*, Yantai, China, 2021, pp. 532–541.

[15] Y. Pan, K. Guo, T. Sun, and M. Darouach, "Bioinspired composite learning control under discontinuous friction for industrial robots," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 85–90, Jul. 2022.

[16] K. Guo, Y. Liu, B. Xu, Y. Xu, and Y. Pan, "Locally weighted learning robot control with improved parameter convergence," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13 236–13 244, Jan. 2022.

[17] B. Lai, Z. Li, W. Li, C. Yang, and Y. Pan, "Homography-based visual servoing of eye-in-hand robots with exact depth estimation," *IEEE Trans. Ind. Electron.*, to be published, 10.1109/TIE.2023.3277072.

[18] K. Guo and Y. Pan, "Composite adaptation and learning for robot control: A survey," *Annu. Rev. Control*, vol. 55, pp. 279–290, 2023.

[19] A. Mohammadi, H. J. Marquez, and M. Tavakoli, "Nonlinear disturbance observers: Design and applications to Euler-Lagrange systems," *IEEE Control Syst. Mag.*, vol. 37, no. 4, pp. 50–72, Aug. 2017.

[20] H. K. Khalil, "High-gain observers in feedback control: Application to permanent magnet synchronous motors," *IEEE Control Syst. Mag.*, vol. 37, no. 3, pp. 25–41, Jun. 2017.

[21] M. Marchi, L. Fraile, and P. Tabuada, "Dirty derivatives for output feedback stabilization," *arXiv preprint arXiv:2202.01941*, 2022.

[22] R. Featherstone, *Rigid Body Dynamics Algorithms*. Berlin, Heidelberg, Germany: Springer, 2007.

[23] M. Walker, "Adaptive control of manipulators containing closed kinematic loops," *IEEE Trans. Robot. Autom.*, vol. 6, no. 1, pp. 10–19, Feb. 1990.

[24] P. M. Wensing, G. Niemeyer, and J. E. Slotine, "Observability in inertial parameter identification," *arXiv preprint arXiv:1711.03896*, 2017.

[25] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. De Luca, "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4147–4154, Oct. 2019.