



## OPEN ACCESS

## EDITED BY

Gustavo J. G. Lahr,  
Hospital Israelita Albert Einstein, Brazil

## REVIEWED BY

Shuchen Ding,  
Suzhou University of Science and  
Technology, China  
Xiaolong Chen,  
Chongqing University, China

## \*CORRESPONDENCE

Tian Wang,  
✉ wangtian@cmu.edu.cn  
Liang Zhang,  
✉ d1949@lnph.com  
Yu Lu,  
✉ d1848@lnph.com

RECEIVED 25 November 2024

ACCEPTED 06 February 2025

PUBLISHED 04 March 2025

## CITATION

Zhang F, Wang T, Zhang L, Shi E, Wang C, Li N,  
Lu Y and Zhang B (2025) Sliding-mode  
control based on prescribed performance  
function and its application to a SEA-Based  
lower limb exoskeleton.  
*Front. Robot. AI* 12:1534040.  
doi: 10.3389/frobt.2025.1534040

## COPYRIGHT

© 2025 Zhang, Wang, Zhang, Shi, Wang, Li, Lu  
and Zhang. This is an open-access article  
distributed under the terms of the [Creative  
Commons Attribution License \(CC BY\)](#). The  
use, distribution or reproduction in other  
forums is permitted, provided the original  
author(s) and the copyright owner(s) are  
credited and that the original publication in  
this journal is cited, in accordance with  
accepted academic practice. No use,  
distribution or reproduction is permitted  
which does not comply with these terms.

# Sliding-mode control based on prescribed performance function and its application to a SEA-Based lower limb exoskeleton

Feilong Zhang<sup>1,2</sup>, Tian Wang<sup>3\*</sup>, Liang Zhang<sup>4\*</sup>, Enming Shi<sup>1,2</sup>,  
Chengchao Wang<sup>1,2</sup>, Ning Li<sup>1</sup>, Yu Lu<sup>4\*</sup> and Bi Zhang<sup>1</sup>

<sup>1</sup>State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, <sup>2</sup>University of Chinese Academy of Sciences, Beijing, China, <sup>3</sup>The Fourth Affiliated Hospital of China Medical University, Shenyang, China, <sup>4</sup>Department of Rehabilitation Medicine, The People's Hospital of Liaoning Province, Shenyang, China

A sliding-mode control based on a prescribed performance function is proposed for discrete-time single-input single-output systems. The controller design aims to maintain the tracking error in a predefined convergence zone described by a performance function. However, due to the fixed structure of the controller, the applicability and universality of this method are limited. To address this issue, we separate the controller into two parts and analyze the principle of the prescribed performance control (PPC) method. Then we can replace the linear part of the controller with model-based control methods to adapt to the specific characteristics of the controlled system. Compared with current works, when the established system model is inaccurate, we can enhance the smoothness or response speed of the system by introducing a penalty constant to alter the system's transient characteristics while the tracking error is within the prescribed domain. Finally, numerical comparison simulations and a lower limb exoskeleton experiment illustrate the established results and the effectiveness of the proposed method.

## KEYWORDS

prescribed performance, sliding-mode control, predefined convergence zone, lower limb exoskeleton, transient characteristics

## 1 Introduction

Most control methods are performance-based methods. For example, the parameters of the PID are designed either to obtain the desired performance of rise time, maximum overshoot and steady-state error of the unit-step response of the system, or to achieve the desired sheared frequency and phase margin through Nichols plot or Bode diagram. If the performance of higher-order system cannot be guaranteed by PID controller, we can not only introduce the state/output feedback control and self-tuning method to achieve the desired locations of the poles which determine the speed and damping of the system response, but also apply the LQR, minimum variance control and predictive control methods to acquire the optimal control performance (Mir and Senroy, 2020; Kim and Ahn, 2018; Tajaddodianfar et al., 2019; Liang et al., 2019; Ahmed et al., 2017; Rosolia and Borrelli, 2017).

Additionally, in consideration of some systems with the slight nonlinearity, parameter uncertainty and even unknown structures, the performance of adaptability is made possible through combining the online estimate algorithm with the above controller according to the certainty equivalence principle (Åström and Wittenmark, 2013; Goodwin and Sin, 2009; Zhou et al., 2018; Zheng et al., 2016; Xi et al., 2024; Chen et al., 2021).

Considerable efforts have been made to deal with the design of the prescribed performance controller (PPC) for a kind of output constrained control problem. This methodology originates from (Bechlioulis and Rovithakis, 2008) and means that the tracking error should converge to an arbitrarily small residual set. Its advantage is that both transient and steady-state performance of the system can be assured, i.e., the convergence rate is no less than a prespecified value and the maximum overshoot is less than a small prespecified domain (Bechlioulis and Rovithakis, 2009; Bechlioulis and Rovithakis, 2011; Bechlioulis and Rovithakis, 2014). In addition, a series of state feedback control schemes are proposed in combination with fault tolerance (Gao et al., 2022), fuzzy adaptive (Chen et al., 2021; Sui et al., 2021), finite-time index (Dong and Yang, 2022; Gao et al., 2021; Liang et al., 2022), and neural network finite-time index (Shi et al., 2024; Deng et al., 2022) for realization of the PPC.

The aforementioned PPC methods are based on continuous state space model, however, sometimes discrete-time control methods are more difficult to analyze and are more suitable for computer control in industrial setting. A kind of sliding mode control based on ARX model was proposed by Nguyen et al. (2017) to maintain the tracking error trajectory in a predefined convergence zone described by the performance function. Similarly, (Liu and Yang, 2018; Liu and Yang, 2019a; Liu and Yang, 2019b), propose one class of sliding mode control based on the equivalent dynamic linearization model and introduce the online identification for the adaptability of this class of method in nonlinear systems (Zhang et al., 2020; Zhang et al., 2021). Generally speaking, we shall begin with the topic on the certain linear system with ARX model like Nguyen et al. (2017) for more easily understanding the principle of this class of discrete PPC method (Nguyen et al., 2017; Liu and Yang, 2018; Liu and Yang, 2019a; Liu and Yang, 2019b), because the adaptability for the uncertainty or nonlinearity is radically introduced by on-line parameters estimation according to certainty equivalence principle Åström and Wittenmark, 2013; Goodwin and Sin, 2009). In addition, Nguyen et al. (2017) mainly focuses on the tracking performance, which is determined by performance function, and disregards the transient characteristics which can be changed through the adjustment of the parameters in controller. Actually, the prescribed performance is not the only index of system performance, and many well-known and visible transient performance indexes of the system should be concerned.

This paper proposes a new sliding mode control with a prescribed performance function. To enhance the applicability and universality of the PPC method, we separate the proposed controller into two parts: the linear feedback part and the nonlinear part. The linear part stabilizes the system by placing the poles of the system at the origin point to hold the system output and satisfy the optimal control index. Meanwhile, the actual output zone is enlarged by adding the nonlinear part of the controller, while the output error is still guaranteed within the prescribed zone defined by Nguyen et al.

(2017) and Liu and Yang (2018), Liu and Yang (2019a), Liu and Yang (2019b). More precisely, simulations show that the linear part is the prerequisite for guaranteeing the convergence of tracking error within a predefined arbitrarily small zone. At the same time, the maximum overshoot is less than the desired constant.

The main contributions are summarized as follows: i) To address the issue where the linear part of the controller cannot ensure sufficiently small errors under conditions of inaccurate modeling, we modify the denominator of the controller by introducing a penalty constant to tune the system behaviors. In this way, we can guarantee the system's output within the prescribed zone and tune the introduced penalty constant to further improve the other system's transient characteristics, such as smoothness, response speed, overshoot, etc. ii) We replace the linear part of the PPC controller with a dynamic model of the exoskeleton, and propose an incremental PPC controller based on PID + dynamics to ensure that the tracking error remains within a predefined region. The proposed control method is then successfully applied on a lower limb exoskeleton.

The rest of the paper is organized as follows: Section 2 provides the problem formulation and preliminaries. Section 3 introduces the design procedure and stability analysis of the proposed method. Section 4 presents simulations to study the PPC method and an experiment to test the controller. The conclusion is given in Section 5.

## 2 Problem formulation and preliminaries

### 2.1 System description

Consider the single-input and single-output discrete-time linear dynamical system (Equation 1) as follow:

$$A(z^{-1})y(k) = B(z^{-1})u(k) + \zeta(k) \quad (1)$$

where  $u(k)$  is the control input;  $y(k)$  is the measured output;  $\zeta(k)$  is the unknown modeling errors and nonlinearities.  $A(z^{-1})$  and  $B(z^{-1})$  are defined as Equation 2

$$\begin{aligned} A(z^{-1}) &= 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{n_a}z^{-n_a} \\ B(z^{-1}) &= b_1z^{-1} + \dots + b_{n_b}z^{-n_b} \end{aligned} \quad (2)$$

where  $n_a$  and  $n_b$  are the orders of system output and control input, respectively.

**Assumption 1:** The unmodeled dynamics  $\zeta(k)$  is slowly varying with respect to the sampling frequency and the unmodeled dynamics estimation error satisfies the condition that  $|\zeta(k)| \leq \delta$ , where  $\delta$  is a small positive number.

### 2.2 Prescribed performance function

The prescribed performance control method proposed by Nguyen et al. (2017) and Liu and Yang (2018), Liu and Yang (2019a),

Liu and Yang (2019b) is designed to ensure adherence to specified tracking error constraints (Equation 3) as follows:

$$-\rho_k \underline{\chi}_k < e(k) = y_d(k) - y(k) < \rho_k \bar{\chi}_k \quad (3)$$

where  $y_d(k)$  represents the desired output of the system at the time  $k$ ;  $\underline{\chi}_k$  and  $\bar{\chi}_k$  represent the lower and upper bounds, respectively.  $\rho_k$  is a bounded and strictly positive decreasing function with the property (Equation 4) as follow:

$$\lim_{k \rightarrow \infty} \rho_k = \rho_{\infty} \quad (4)$$

and have given the performance function and the upper and lower bounds (Equations 5–7) as follow:

$$\rho_{k+1} = \kappa \rho_{\infty} + (1 - \kappa) \rho_k \quad (5)$$

$$\bar{\chi}_{k+1} = \kappa + (1 - \kappa) \bar{\chi}_k \quad (6)$$

$$\underline{\chi}_{k+1} = \kappa + (1 - \kappa) \underline{\chi}_k \quad (7)$$

where  $\rho_0 \geq \rho_k \geq \rho_{\infty} \geq 0$  and  $0 < \kappa < 1$ .

To address the constrained control issue (Equation 3), we transform the tracking error  $e(k)$  into an unconstrained equivalent form and give a strictly increasing function  $\Theta(\tau_k)$  of a transformed error  $\tau_k$ . Define

$$e(k) = \rho_k \Theta(\tau_k) \quad (8)$$

The strictly increasing function must satisfy the following conditions (Equations 9, 10).

$$\Theta(\tau_k) \in \left( -\underline{\chi}_k, \bar{\chi}_k \right) \quad (9)$$

$$\lim_{\tau_k \rightarrow +\infty} \Theta(\tau_k) = \bar{\chi}_k \text{ and } \lim_{\tau_k \rightarrow -\infty} \Theta(\tau_k) = -\underline{\chi}_k \quad (10)$$

Due to the above properties of  $\Theta(\tau_k)$  and  $\rho_k \geq \rho_{\infty} > 0$ , the inverse transformation can be obtained by Equation 11

$$\tau_k = \Theta^{-1} \left( \frac{e(k)}{\rho_k} \right) \quad (11)$$

With regard to the given tracking error  $e_0$ , if  $\rho_0$  is selected such that  $-\underline{\chi}_0 \rho_0 < e(0) < \rho_0 \bar{\chi}_0$  and  $\tau_k$  is bounded, then  $\Theta(\tau_k) \in \left( -\underline{\chi}_k, \bar{\chi}_k \right)$  holds and (Equation 3) is guaranteed.

We introduce a strictly increasing function (Equation 12) for control design.

$$\Theta(\tau_k) = \frac{\bar{\chi}_k e^{b\tau_k} - \underline{\chi}_k a}{e^{b\tau_k} + a} \quad (12)$$

Then we have the transformed error  $\tau_k$  as follows:

$$\tau_k = \frac{1}{b} \ln a \left( \frac{\underline{\chi}_k \rho_k + e(k)}{\bar{\chi}_k \rho_k - e(k)} \right) \quad (13)$$

When  $a = 1$  and  $b = 2$ , (Equation 13) will degenerate into the transformed error in Nguyen et al. (2017).

The above prescribed performance function design in Liu and Yang (2018), Liu and Yang (2019a), Liu and Yang (2019b) is designed for the system without consideration of unmodeled dynamics, then Nguyen et al. (2017) further modified the tracking error constraint (Equation 3) into

$$-\delta - \rho_k \underline{\chi}_k < e(k) < \rho_k \bar{\chi}_k + \delta \quad (14)$$

Which takes the unmodeled dynamics including the disturbance and nonlinearities and offset error into consideration.  $\delta$  is a small constant (Nguyen et al., 2017).

### 3 Prescribed performance control design

Consider the following sliding mode function (Equation 15):

$$s(k) = \theta(k) - 1 \quad (15)$$

where  $\theta(k)$  is a variable derived from the transformed error and is defined as Equation 16

$$\theta(k) = \frac{\underline{\chi}_k \rho_k + e(k)}{\bar{\chi}_k \rho_k - e(k)} \quad (16)$$

Then the reaching condition is given as follows:

$$s(k+1) - s(k) = -qTs(k) - \Lambda_s \text{sign}(s(k)) \quad (17)$$

where  $\Lambda_s > 0$ . In this note, we adopt

$$\Lambda_s = qT \quad (18)$$

We can obtain the one step-ahead tracking error as follows

$$\begin{aligned} e(k+1) &= y_d(k+1) - y(k+1) \\ &= y_d(k+1) + a_1(k) + \dots + a_{na} y(k - n_a + 1) \\ &\quad - b_1 u(k) - \dots - b_{nb} u(k - n_b + 1) - \zeta(k-1) \end{aligned} \quad (19)$$

Here, we define  $\hat{\zeta}(k)$  as the estimated unmodeled dynamics. According to the discrete-time perturbation estimation technique (Nguyen et al., 2017), we can estimate this unknown term by one-step delayed value as follows:

$$\begin{aligned} \zeta(k+1) \approx \zeta(k) &= y(k) + a_1 y(k-1) + \dots + a_{na} y(k - n_a) - b_1 u(k-1) \\ &\quad - \dots - b_{nb} u(k - n_b) \end{aligned} \quad (20)$$

We define the perturbation estimation error  $\tilde{\zeta}(k)$  as follow

$$\tilde{\zeta}(k) = \hat{\zeta}(k) - \zeta(k) \quad (21)$$

From Equations 17, 18, 20, 21, we have

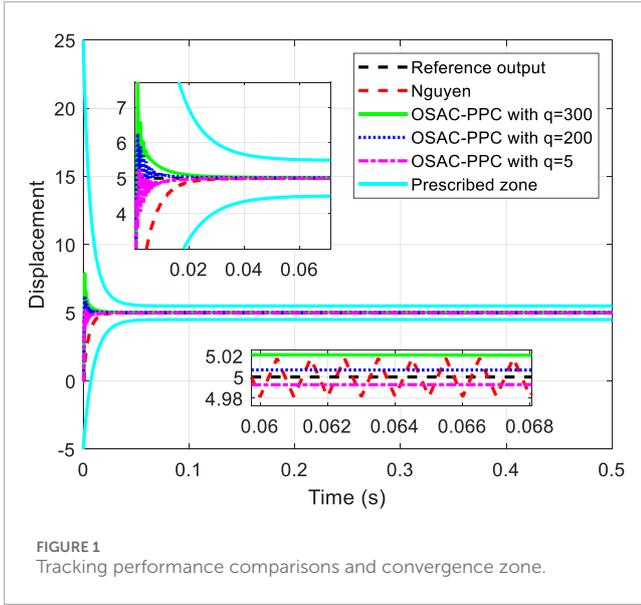


FIGURE 1 Tracking performance comparisons and convergence zone.

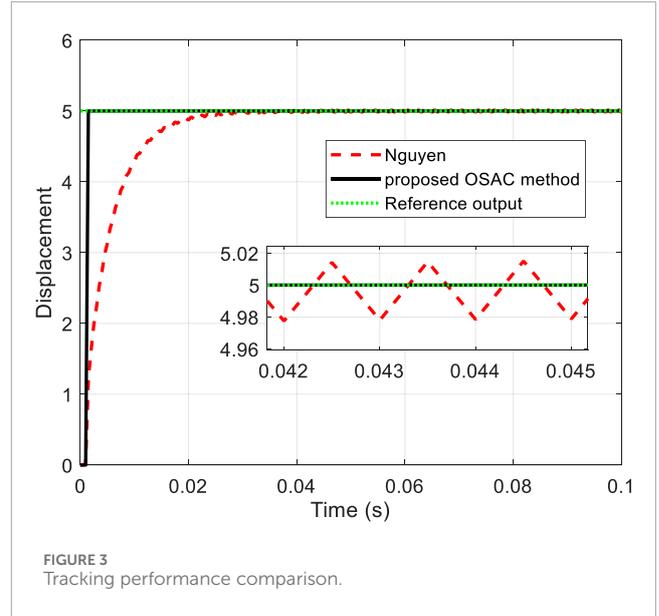


FIGURE 3 Tracking performance comparison.

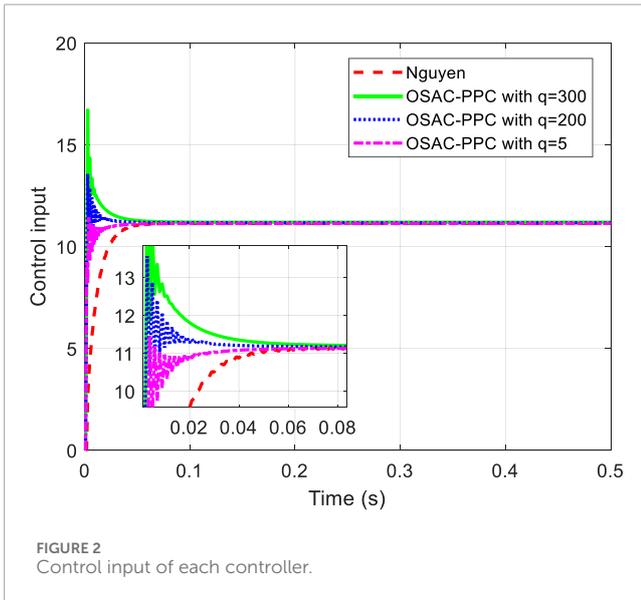


FIGURE 2 Control input of each controller.

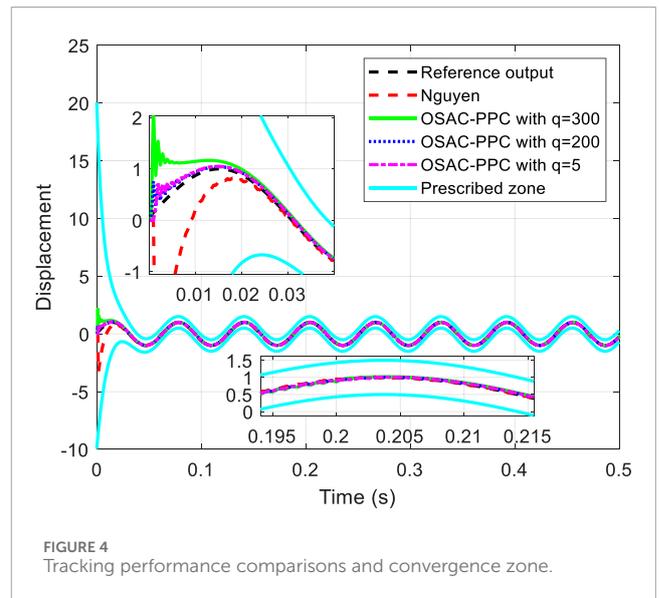


FIGURE 4 Tracking performance comparisons and convergence zone.

$$\frac{\chi_{-k+1}\rho_{k+1} + e(k+1)}{\bar{\chi}_{k+1}\rho_{k+1} - e(k+1)} = (1 - qT)\theta(k) + qT - \Lambda_s \text{sign}(s(k)) \quad (22)$$

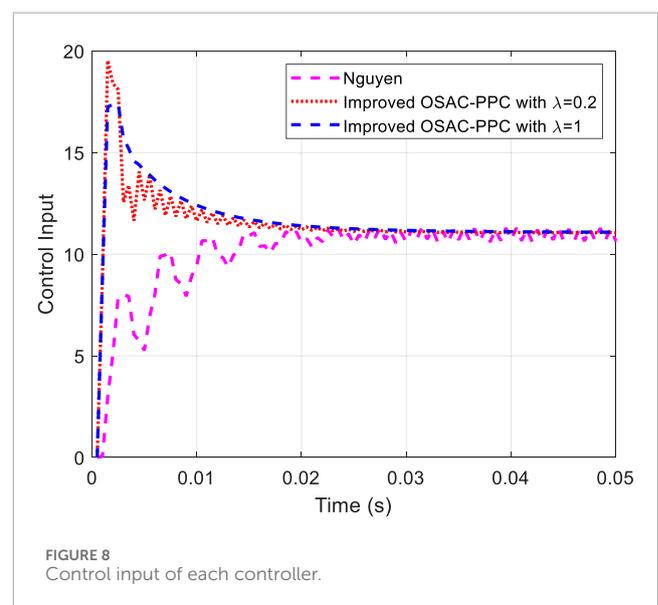
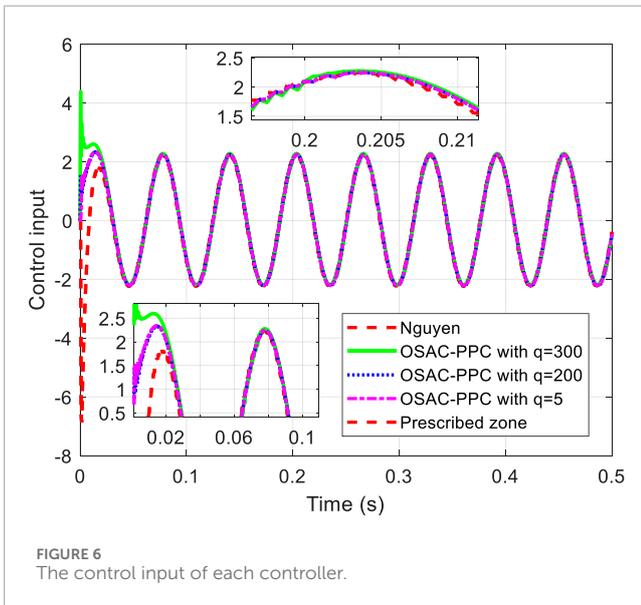
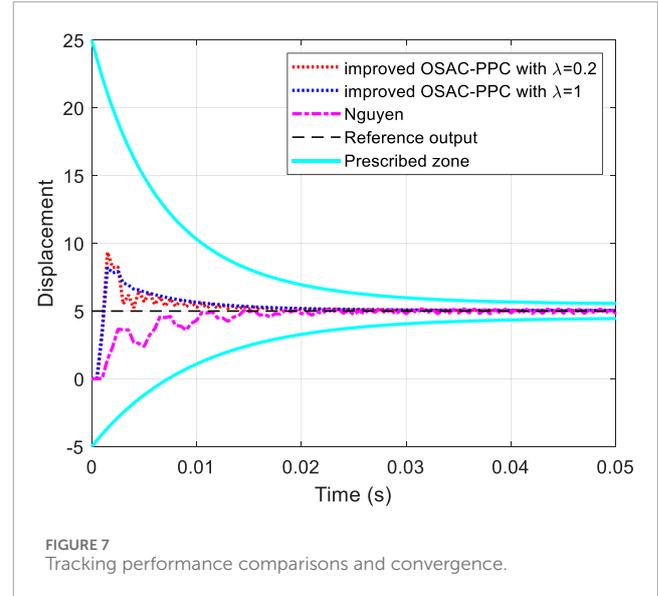
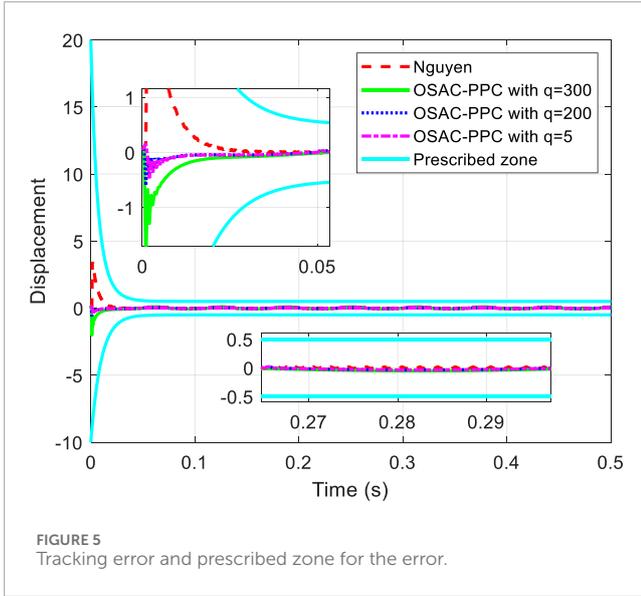
Then we have (Equation 23) by simplifying (Equation 22).

$$\begin{aligned} & e(k+1) + a_1\Delta y(k) + \dots + a_{n_a}\Delta y(k - n_a + 1) \\ & - b_1\Delta u(k) - \dots - b_{n_b}\Delta u(k - n_b + 1) + \tilde{\zeta}(k+1) \\ & \frac{\bar{\chi}_{k+1}[(1 - qT)\theta(k) + qT - \Lambda_s \text{sign}(s(k))] - \chi_{-k+1}}{(1 - qT)\theta(k) + qT + 1 - \Lambda_s \text{sign}(s(k))} \rho_{k+1} = 0 \quad (23) \end{aligned}$$

where  $\Delta y(k-i) = y(k-i) - y(k-1-i)$ , ( $i = 0, \dots, n_a$ ) and  $\Delta u(k-j) = u(k-j) - u(k-1-j)$ , ( $j = 0, \dots, n_b$ ). Because the unmodeled dynamics estimation error  $\tilde{\zeta}(k)$  is unknown in experiment (Nguyen et al., 2017), we can obtain the

control output (Equation 24) through solving (Equation 23) with the absence of  $\tilde{\zeta}(k)$ .

$$\begin{aligned} \Delta u(k) &= b_1^{-1} \left[ y_d(k+1) + a_1y(k) + \dots + a_{n_a}y(k - n_a + 1) - b_2u(k-1) \right. \\ & \quad \left. - \dots - b_{n_b}u(k - n_b + 1) - \tilde{\zeta}(k+1) \right. \\ & \quad \left. \frac{\bar{\chi}_{k+1}[(1 - qT)\theta(k) + qT - \Lambda_s \text{sign}(s(k))] - \chi_{-k+1}}{(1 - qT)\theta(k) + qT + 1 - \Lambda_s \text{sign}(s(k))} \rho_{k+1} \right] \\ &= b_1^{-1} \left[ e(k+1) + a_1\Delta y(k) + \dots + a_{n_a}\Delta y(k - n_a + 1) \right. \\ & \quad \left. - b_2\Delta u(k-1) - \dots - b_{n_b}\Delta u(k - n_b + 1) \right. \\ & \quad \left. \frac{\bar{\chi}_{k+1}[(1 - qT)\theta(k) + qT - \Lambda_s \text{sign}(s(k))] - \chi_{-k+1}}{(1 - qT)\theta(k) + qT + 1 - \Lambda_s \text{sign}(s(k))} \rho_{k+1} \right] \quad (24) \end{aligned}$$



To further study the principle of controller (Equation 24), we separate the controller into two parts as follows:

$$\Delta u_{linear}(k) = b_1^{-1} \left[ e(k+1) + a_1 \Delta y(k) + \dots + a_{n_a} \Delta y(k - n_a + 1) - b_2 \Delta u(k-1) - \dots - b_{n_b} \Delta u(k - n_b + 1) \right] \quad (25)$$

$$\Delta u_{nonlinear}(k) = -b_1^{-1} \frac{\chi_{k+1} - \bar{\chi}_{k+1}[(1-qT)\theta(k) + qT - \Lambda_s \text{sign}(s(k))]}{(1-qT)\theta(k) + qT + 1 - \Lambda_s \text{sign}(s(k))} \rho_{k+1} \quad (26)$$

where  $\Delta u_{linear}(k)$  is the linear part which places all the poles of the system to the original point and it can be regarded as one-step-ahead control (OSAC);  $\Delta u_{nonlinear}(k)$  denotes a nonlinear part which keeps system output not beyond the prescribed domain, although it may cause the output of system away from the desired trajectory.

We name (Equation 24) by OSAC-PPC and name (Equation 25) by OSAC for conveniently discussion in the simulations.

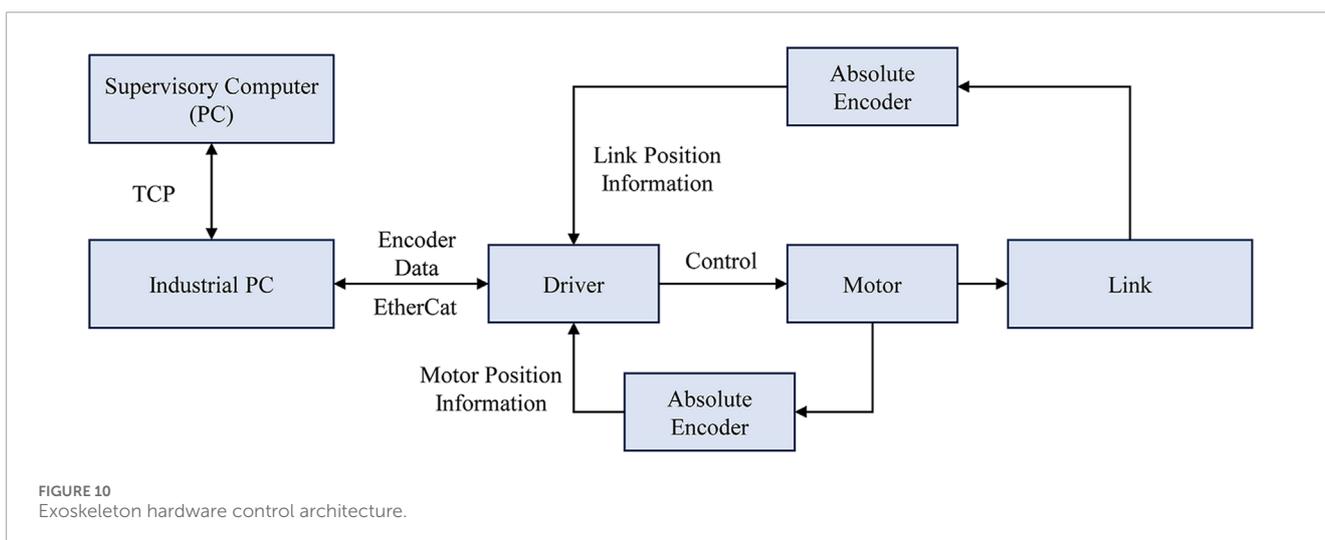
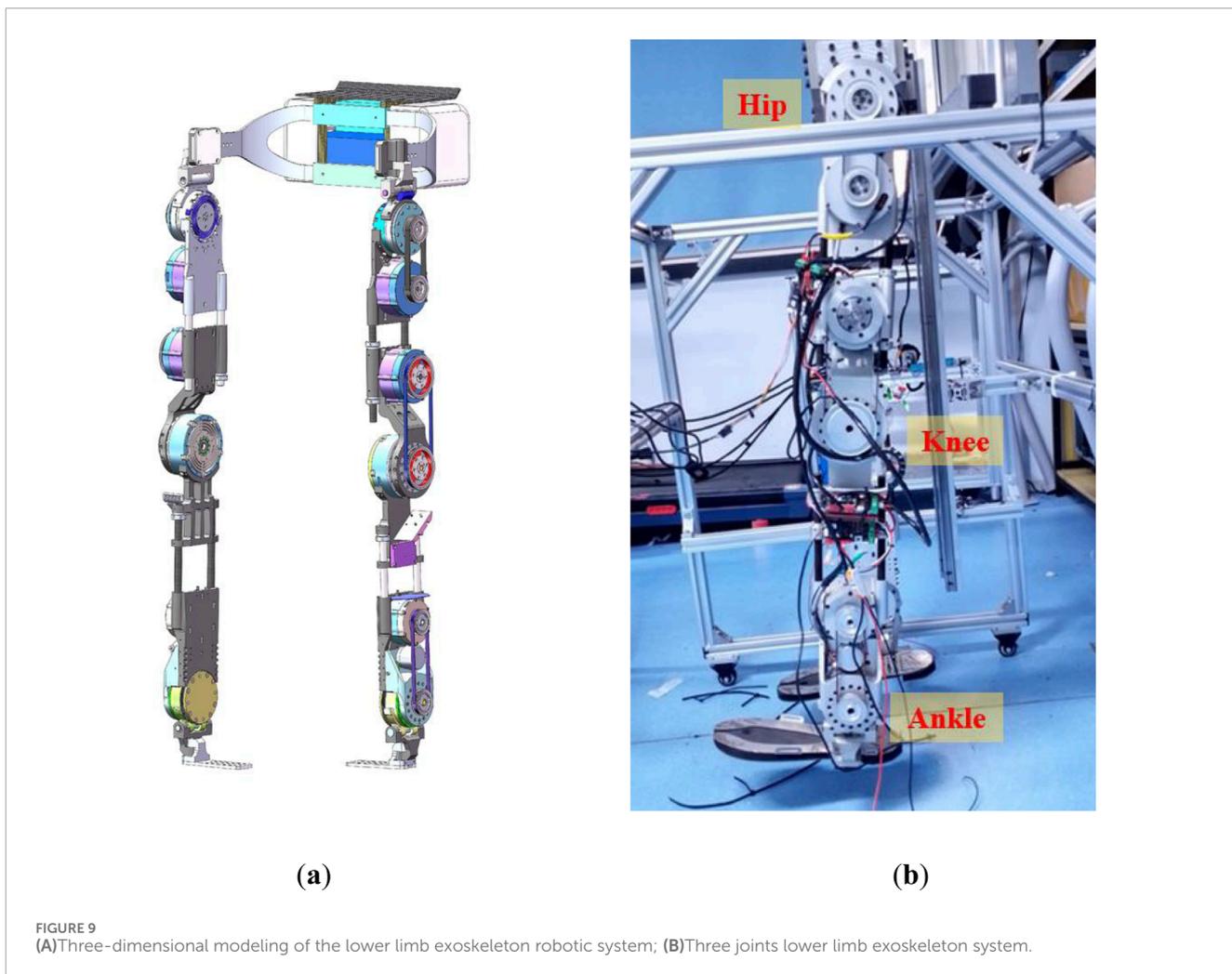
Then the control law (Equation 24) can be rewritten as Equation 27

$$u(k) = u(k-1) + \Delta u_{linear}(k) + \Delta u_{nonlinear}(k) \quad (27)$$

**Theorem 1:** Consider the nonlinear system (Equation 1) with sliding control law (Equation 24). If the initial parameters  $\bar{\chi}_0, \underline{\chi}_0$  and  $\rho_0$  are properly selected to satisfy  $-\delta - \underline{\chi}_0 \rho_0 < e(0) < \bar{\chi}_0 \rho_0 + \delta$ , the prescribed performance described by Equation 14 will be guaranteed for all  $k > 0$ .

Proof: The proof of Theorem 1 is given in the Appendix.

**Remark 1:** Similar to the existing PPC approaches in Nguyen et al. (2017) and Liu and Yang (2018), Liu and Yang (2019a), Liu and Yang (2019b), this paper discusses a transformed error algorithm combined with a new sliding mode control strategy to guarantee



the tracking error converges to a predefined zone. The control law is separated into two parts: the linear part and nonlinear part. The linear part defined as  $\Delta u_{linear}(k)$  in this note is to stabilize the system by placing the poles of the system to origin point for the optimal

control performance. The nonlinear part defined as  $\Delta u_{nonlinear}(k)$  in this note keeps the output trajectory not crossing the boundaries of the prescribed domain based on the linear part. More precisely, the actual output zone is comparatively enlarged when we add

TABLE 1 Dynamic parameters of exoskeleton.

Parameters	Hip (1)	Knee (2)	Angle (3)
$m_i(kg)$	2.09	1.6	0.55
$m_i L_{ci}(kg \cdot m)$	0.1567	0.0689	0.0024
$L_i(m)$	0.37	0.36	*
$J_i(kg \cdot m^2)$	-0.304	0.0002	0.0097
$f_{ci}(Nm)$	0.0847	0.6758	1.7567
$f_{vi}(Nm \cdot s/rad)$	-3.2967	-3.7335	-12.2408

TABLE 2 Controller parameters.

Parameters	Joint 1	Joint 2	Joint 3
$q^i$	5	5	5
$\kappa$	0.001	0.001	0.001
$\bar{\chi}_0^i$	1	1	1
$\chi_0^i$	1	1	1
$\rho_0^i$	2	2	2
$\rho_{\infty}^i$	0.05	0.05	0.03
$k_p^i$	300	300	400
$k_T^i$	3	1.5	4

the nonlinear part into the principal linear part of the controller, whereas the output error is still guaranteed within the prescribed zone which is depicted by  $\rho_0, \rho_{\infty}, \kappa, \chi_k$  and  $\bar{\chi}_k$ . In other words, the linear part is the key to guarantee the tracking error convergent to a predefined arbitrarily small zone with convergence rate no less than a preassigned value and the maximum overshoot less than a desired constant. To this end, we can replace the linear part with any other kind of model-based control methods, such as PID, self-tuning control and Model predictive control (MPC), etc., as long as they are able to guarantee the tracking error within a sufficient small range, so as to accommodate the effects caused by nonlinear part for the achievement of the prescribed performance in Nguyen et al. (2017) and Liu and Yang (2018), Liu and Yang (2019a), Liu and Yang (2019b).

On the other hand, to avoid the estimated parameter  $\hat{b}_1$  close to zero or to change the control effect of the linear part, we can modify the OSAC-PPC (24) into (28) with introducing the parameter  $\lambda$  in denominator as follow:

$$\Delta u(k) = \frac{\hat{b}_1}{\lambda + \hat{b}_1^2} \begin{bmatrix} e(k+1) + \hat{a}_1 \Delta y(k) + \dots + \hat{a}_{na} \Delta y(k - n_a + 1) \\ -\hat{b}_2 \Delta u(k-1) - \dots - \hat{b}_{nb} \Delta u(k - n_b + 1) \\ \bar{\chi}_{k+1} [(1 - qT)\theta(k) + qT - \Lambda_s \text{sign}(s(k))] - \chi_{k+1} \\ \frac{\chi_{k+1}}{(1 - qT)\theta(k) + qT + 1 - \Lambda_s \text{sign}(s(k))} \rho_{k+1} \end{bmatrix} \quad (28)$$

where  $\hat{a}_1, \dots, \hat{a}_{na}, \hat{b}_1, \dots, \hat{b}_{nb}$  represent the estimated values of  $a_1, \dots, a_{na}, b_1, \dots, b_{nb}$ , respectively. Then the modified controller (Equation 28) has the identical structure with those in Liu and Yang (2018), Liu and Yang (2019a), Liu and Yang (2019b). And we name (Equation 28) by improved OSAC-PPC.

### 4 Experiment and result

Example 1: Consider the system (Equation 29) in Nguyen et al. (2017).

$$y(k+1) = a_1 y(k) - a_2 y(k-1) + b_1 u(k) + b_2 u(k-1) + \zeta(k+1) \quad (29)$$

where  $a_1 = -0.1903, a_2 = -0.00906, b_1 = 0.4906$  and  $b_2 = 0.04723$ . We assume  $\zeta(k) = 0$  at beginning to exhibit the effects of controllers more clearly. All the simulation settings including reference output, convergence zone described by Equation 14, and the controller parameters  $T = 0.0005, \kappa = 0.05, \rho_0 = 10, \rho_{\infty} = 0.5, \chi_0 = 1$  and  $\bar{\chi}_0 = 2$  are chosen in common with Nguyen et al. (2017). The proposed OSAC-PPC method (24) choose controller parameter  $q$  with different values: 300, 200 and 5. To be consistent with Nguyen et al. (2017). Figure 1 shows the step response comparison between the proposed OSAC-PPC method and the Nguyen's method. Control input of each controller is shown in Figure 2.

Figure 1 obviously shows that each output of system controlled by these methods is in prescribed zone. From Figure 1, we can see that: i) from the time of  $[0, 0.02]$ , the tracking error of system controlled by the proposed OSAC-PPC method is smaller than that of Nguyen. Besides, we can change the transient performance by adjusting  $q$ . When the parameter  $q$  decreases, the convergence speed and the oscillation of system controlled by proposed method will increase; ii) from the time of  $[0.06, 0.066]$ , the output of system controlled by the proposed OSAC-PPC method with  $q = 5$  consistently converges to the desired trajectory; The output of the system controlled by the proposed OSAC-PPC method with  $q = 200$  passes through the output trajectory of Nguyen's controller; The output of the system controlled by the proposed OSAC-PPC method with  $q = 300$  is designed so as to be tangent to the actual output zone of Nguyen.

From Figure 1, we can conclude that the tracking error of the proposed method is smaller than that of Nguyen under the same prescribed performance. In addition, by decreasing the controller parameter  $q$ , we can change the system transient characteristics to obtain a better convergence speed, nevertheless the oscillation is enlarged. Therefore, the proposed OSAC-PPC method is more flexible than Nguyen's controller for introducing the key adjustable parameter  $q$ . As a result, we have more choices for the system behaviors and transient characteristics.

On the other hand, to further study this kind of prescribed performance method, we choose  $\rho_0 = 0$  and  $\rho_{\infty} = 0$  with an aim to remove the nonlinear part of the controller, i.e.,  $\Delta u_{\text{nonlinear}}(k) = 0$ . The output of the system controlled by the only linear part of OSAC-PPC method (Equation 24) [i.e., OSAC (Equation 25)] and Nguyen's method are shown in Figure 3.

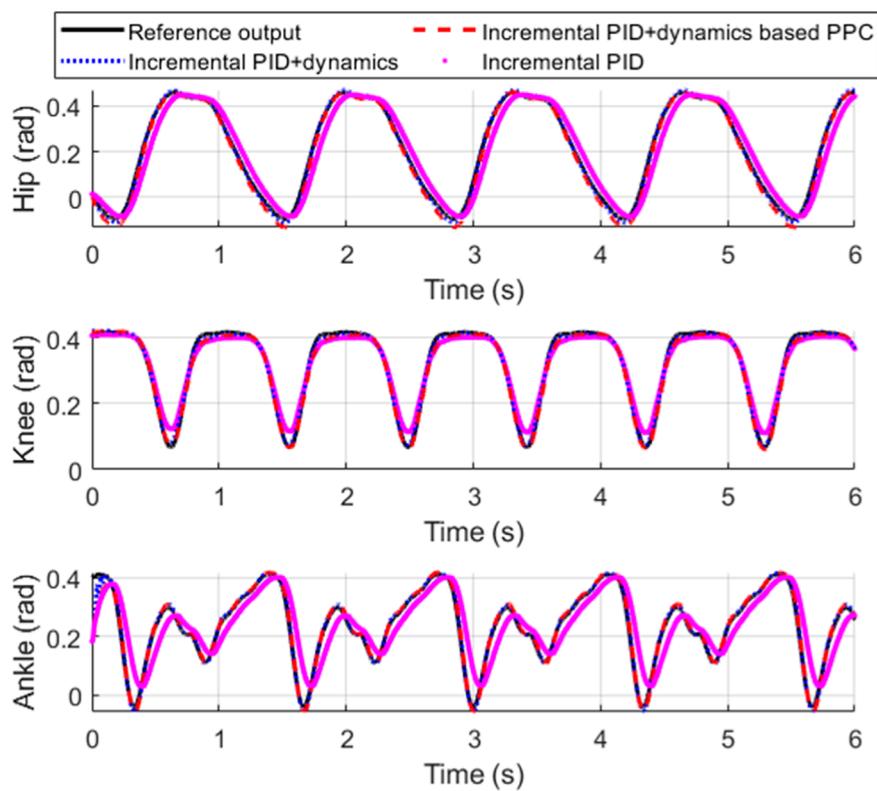


FIGURE 11  
Tracking performance comparisons.

From Figure 3, we can see that the system controlled by OSAC (Equation 25) exhibits a better performance even without overshoot in simulation. It is natural for us to conclude that the effectiveness of the proposed method can be separated into two parts: i) the stability of the system is guaranteed by the linear part of the OSAC-PPC controller (Equation 24) whose intrinsic design is for the achievement of the optimal performance. ii) The actual output zone is virtually widened by the nonlinear part of the controller (Equation 26); however, the system is still guaranteed within the prescribed zone depicted by  $\rho_0$ ,  $\rho_\infty$  and  $\kappa$ .

When we change the reference trajectory to  $y(k) = \sin(2 \times 10^{-5}t)$ , while keeping the controller parameters unchanged, the corresponding tracking performance is depicted in Figure 4. The tracking error is shown in Figure 5, and the control inputs are illustrated in Figure 6.

From Figures 4, 5, it can be observed that both the proposed OSAC-PPC method and Nguyen's method keep the system outputs and tracking errors within the prescribed zone. Moreover, the proposed OSAC-PPC method exhibits a smaller tracking error compared to Nguyen's method.

**Example 2:** This example shows that the improved OSAC-PPC (Equation 28) are more suitable for the case of inaccuracy of estimate parameters. We assume that the inaccurately offline estimated parameters are  $\hat{a}_1 = a_1 + 0.3$ ,  $\hat{a}_2 = a_2 + 0.7$ ,  $\hat{b}_1 = b_1 - 0.1$  and  $\hat{b}_2 = b_2 - 0.02$  in Example 1. Then the corresponding unmodeled

dynamics will be Equation 30

$$\zeta(k+1) = -0.3y(k) - 0.7y(k-1) + 0.1u(k) + 0.02u(k-1) \quad (30)$$

The output of the system controlled by Nguyen's controller and improved OSAC-PPC (28) with  $\lambda = 0.2$  and  $\lambda = 1$  is shown in Figure 7, respectively. The control input of each controller is shown in Figure 8.

From Figure 7, we can find that the output of the system controlled by Nguyen's controller or improved OSAC-PPC (28) with  $\lambda = 0.2$  oscillates at the beginning owing to the estimate parameter error. By comparison, when we choose  $\lambda = 1$ , the output oscillation of the system controlled by improved OSAC-PPC (Equation 28) is eliminated. On the other hand, Figure 8 also shows that the control input of Nguyen oscillates obviously from beginning to end. By comparison, the oscillation of the control input of improved OSAC-PPC (28) is eliminated through letting  $\lambda = 1$ , since the increment of penalty constant  $\lambda$  constrains the change of the control output. Owing to the system's transient characteristics can be changed through adjusting the introduced penalty constant  $\lambda$ , the improved OSAC-PPC (28) are more stable and suitable for the cases of inaccuracy of estimate parameters.

There are many well-known performance indexes for the system, we should not focus on only one index (i.e., prescribed performance) but should consider more system transient indexes such as system convergent speed, smoothness, overshoot and so on. These examples show that some transient performances can be improved by

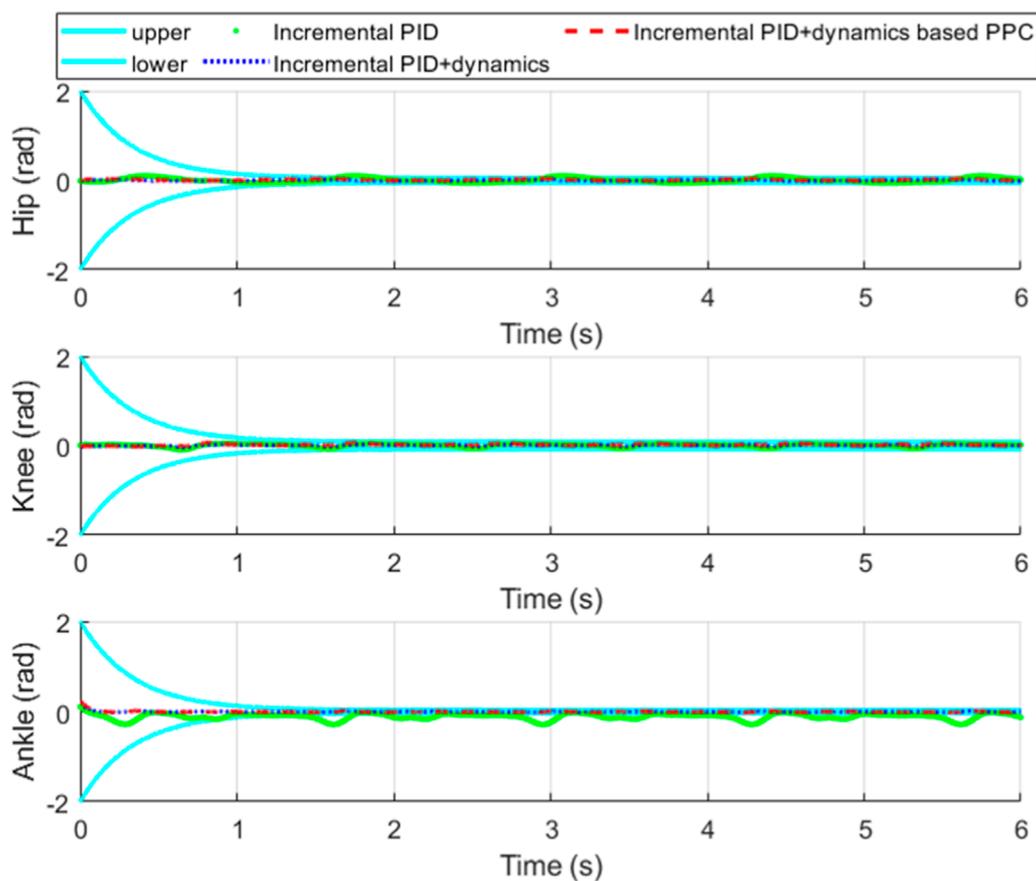


FIGURE 12  
Tracking error and its convergence zone.

our improvements meanwhile the prescribed performance is still satisfied.

Owing to the system's transient characteristics can be changed through adjusting the introduced penalty constant  $\lambda$ , Furthermore, we can replace the linear part of improved OSAC-PPC (28) with MPC or self-tuning PID. It will also achieve the comparable performance in the same way.

**Example 3:** We validate the PPC controller in a three joints SEA-based lower limb exoskeleton system, which is shown in Figure 9. The joints of this system are actively controlled through current loops, with each joint's torque values calculated by the controller and then converted into control inputs. This motor-driven approach enables the exoskeleton to follow the reference trajectories, which are scaled versions of gait cycle joint angles, specifically chosen to test the controller's performance. The exoskeleton operates on an EtherCAT-based motor drive system.

The platform is equipped with two 24 V serial batteries providing the necessary power for the lower limb exoskeleton. The drive module comprises six modular joints, divided equally between the left and right limbs, corresponding to the human body's hip, knee, and ankle joints. Absolute

encoders installed at each joint measure the angles of rotation.

Figure 10 illustrates the hardware control architecture of the exoskeleton. The heart of the exoskeleton's control system is an industrial PC (IPC) equipped with an i7 7600U processor running the Ubuntu operating system. This IPC serves as the control center where the core control strategies are implemented. Communication between the IPC and the supervisory computer is established via SSH, with the startup program initiated and control commands sent using TCP communication. The exoskeleton's status information is transmitted using LCM.

In terms of exoskeleton control commands, the IPC is connected to the drivers and operates as the EtherCAT master using the IgH EtherCAT Master for Linux, while the drivers function as slaves. The communication between the IPC and the exoskeleton includes sending and receiving control commands and sensor information with a control cycle of 1,000 Hz.

To establish a baseline for comparison, we included two control groups in our study. The first control group employs an incremental PID + dynamics controller that we developed, which integrates dynamic feedforward compensation to enhance control performance. The second group uses a traditional incremental PID controller, serving as a baseline to assess the advancements offered by our proposed solutions.

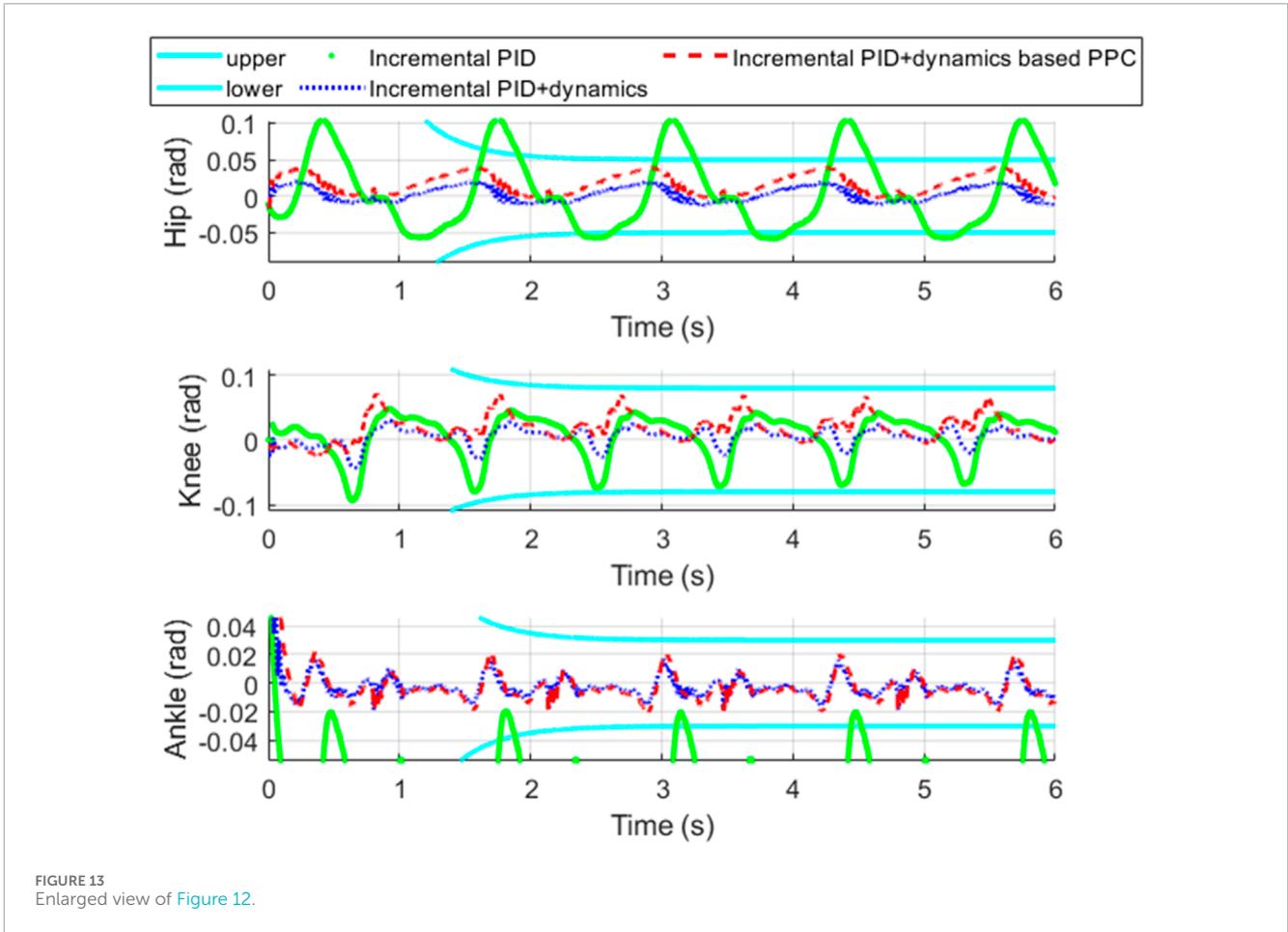


FIGURE 13  
Enlarged view of Figure 12.

The dynamic model (Equations 31, 32) is given by

$$(H(q) + S^T)\ddot{q} + (B + S)\ddot{\theta}_M + C(q, \dot{q})\dot{q} + G(q) + \tau_f = u \quad (31)$$

$$H(q) = M_L(q) + M_R(q) + S(q)B^{-1}S^T(q) \quad (32)$$

where,  $H(q)$  is the total inertia matrix;  $B$  is the inertia matrix of the motor considering the reduction ratio; and  $S$  is the motor-link coupling matrix, which is a constant matrix.  $M_R(q)$  is the added inertia matrix of the connecting link introduced by the motor (Li et al., 2017; Chen et al., 2019; Siciliano and Khatib, 2016).

The dynamic parameters of links are inaccurately identified and given in Table 1.

Since the leading coefficient of control input is an identity matrix, we can describe the linear part of the controller as

$$\begin{aligned} \Delta u_{\text{linear}}(k) = & K_p[e(k+1) - e(k)] + K_I e(k+1) + (H(q(k)) + S^T)\dot{q}(k) \\ & + (B + S)\ddot{\theta}_M(k) + C(q(k), \dot{q}(k))\dot{q}(k) + G(q(k)) + \tau_f \\ & - [(H(q(k-1)) + S^T)\dot{q}(k-1) + (B + S)\ddot{\theta}_M(k-1) \\ & + C(q(k-1), \dot{q}(k-1))\dot{q}(k-1) + G(q(k-1)) + \tau_f] \end{aligned} \quad (33)$$

where,  $K_p = \text{diag}(k_p^1, k_p^2, k_p^3)$  and  $K_I = \text{diag}(k_I^1, k_I^2, k_I^3)$ .

Meanwhile, the nonlinear part can be described as

$$\Delta u_{\text{nonlinear}}(k) = \begin{bmatrix} \bar{\chi}_{k+1}^1 [(1 - q^1 T)\theta^1(k) + q^1 T - \Lambda_s^1 \text{sign}(s^1(k))] - \bar{\chi}_{k+1}^1 \rho_{k+1}^1 \\ \bar{\chi}_{k+1}^2 [(1 - q^2 T)\theta^2(k) + q^2 T - \Lambda_s^2 \text{sign}(s^2(k))] - \bar{\chi}_{k+1}^2 \rho_{k+1}^2 \\ \bar{\chi}_{k+1}^3 [(1 - q^3 T)\theta^3(k) + q^3 T - \Lambda_s^3 \text{sign}(s^3(k))] - \bar{\chi}_{k+1}^3 \rho_{k+1}^3 \end{bmatrix} \quad (34)$$

where,  $\bar{\chi}_i$  represent the parameters for  $i$ -th joint.

Then we propose an incremental PID + dynamics based PPC controller as Equation 35

$$u(k) = u(k-1) + \Delta u_{\text{linear}}(k) + \Delta u_{\text{nonlinear}}(k) \quad (35)$$

The controllers used for comparison are incremental PID + dynamics feedforward control (Equation 33), and incremental PID (Equation 36).

$$\Delta u_{\text{PID}}(k) = K_p[e(k+1) - e(k)] + K_I e(k+1) \quad (36)$$

It should be noted that the incremental PID + dynamics-based PPC controller is obtained by summing the PID + dynamics feedforward compensation controller (Equation 33) with the nonlinear part (Equation 34). Both controllers use dynamics feedforward compensation in differential form.

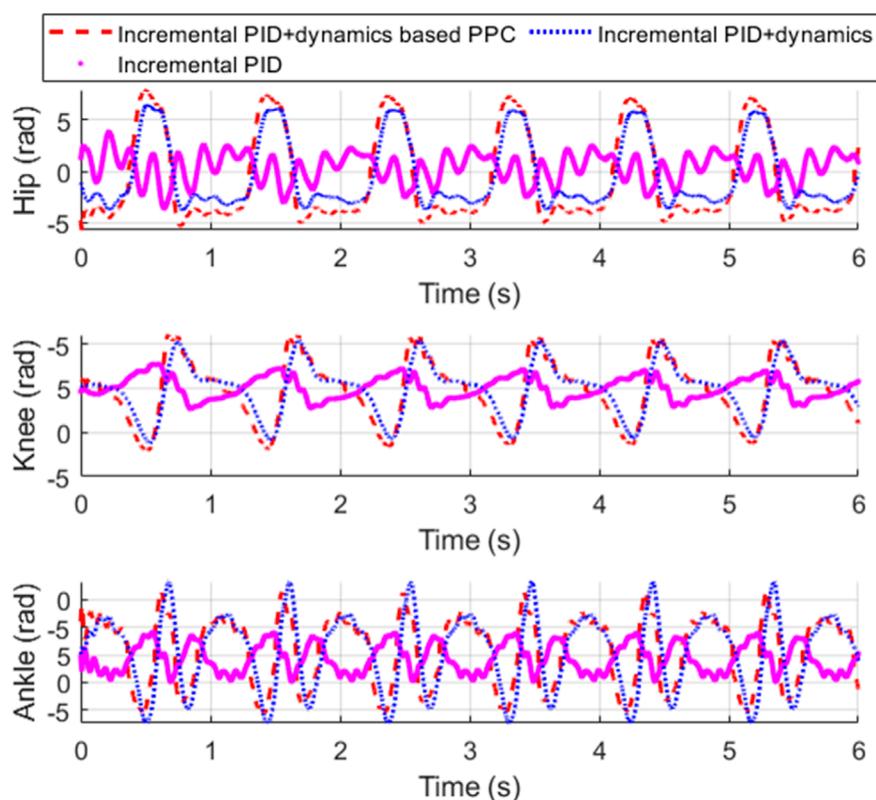


FIGURE 14  
Control input of each controller.

The controller parameters are given by Table 2. The sampling time  $T = 0.001$ s. Figure 11 shows the tracking performance of each joint of the exoskeleton. Figure 12 shows the tracking error of each joint and Figure 13 shows the enlarged view of Figure 12. The control input of each joint is shown in Figure 14.

From Figure 11, it is evident that compared to incremental PID, both incremental PID + dynamics-based PPC controller and incremental PID + dynamics controller significantly enhance the tracking accuracy of the three joints. This indicates that our designed differential form of dynamics-based feedforward effectively compensates for the accuracy deficiencies caused by the low stiffness and elastic characteristics of the SEA. From Figures 12, 13, we can observe that the system controlled by the incremental PID + dynamics-based PPC controller does not perform superior to the incremental PID + dynamics controller. However, the tracking errors for both remain within the prescribed zone. The design of the PPC is not intended to enhance the controller's performance; rather, control accuracy and response time might be reduced due to the prescribed performance index or constraint of the tracking error. Despite this, the method ensures that errors are maintained within the prescribed zone limits.

## 5 Conclusion

In this paper, we propose a discrete-time sliding-mode control method based on prescribed performance control

method to cope with a kind of tracking error constrained problem in the discrete-time dynamical system. By separating controller into two parts, we analyzed the principle of a family of PPC methods which aim to ensure that the tracking error converges to a predefined region. Moreover, we give some alternative improvements for the linear part of the controller for its alterable performance and yet not to impact the premise of prescribed performance. Simulations and experiments are provided to validate the established results and the effectiveness of the proposed method.

## Data availability statement

The datasets presented in this article are not readily available because The data that support the findings of this study are available on request from the author, upon reasonable request. Requests to access the datasets should be directed to Feilong Zhang, zhangfeilong@sia.cn.

## Author contributions

FZ: Conceptualization, Formal Analysis, Investigation, Methodology, Writing—original draft. TW: Project administration,

Resources, Validation, Visualization, Writing–review and editing. LZ: Funding acquisition, Project administration, Resources, Validation, Writing–review and editing. ES: Data curation, Software, Supervision, Writing–review and editing. CW: Methodology, Software, Writing–review and editing. NL: Project administration, Resources, Supervision, Writing–review and editing. YL: Funding acquisition, Project administration, Resources, Validation, Writing–review and editing. BZ: Conceptualization, Funding acquisition, Methodology, Project administration, Writing–original draft, Writing–review and editing.

## Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work is supported by the National Natural Science Foundation of China under Grants 62473361 and 62403457, and is also supported by the Natural Science Foundation of Liaoning Province under Grant 2023-MSLH-117.

## References

- Ahmed, A. A., Koh, B. K., and Lee, Y. I. (2017). A comparison of finite control set and continuous control set model predictive control schemes for speed control of induction motors. *IEEE Trans. Industrial Inf.* 14 (4), 1334–1346. doi:10.1109/tii.2017.2758393
- Åström, K. J., and Wittenmark, B. (2013). Adaptive control [M]. *Cour. Corp.*
- Bechlioulis, C. P., and Rovithakis, G. A. (2008). Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance. *IEEE Trans. Automatic Control* 53 (9), 2090–2099. doi:10.1109/tac.2008.929402
- Bechlioulis, C. P., and Rovithakis, G. A. (2009). Adaptive control with guaranteed transient and steady state tracking error bounds for strict feedback systems. *Automatica* 45 (2), 532–538. doi:10.1016/j.automatica.2008.08.012
- Bechlioulis, C. P., and Rovithakis, G. A. (2011). Robust partial-state feedback prescribed performance control of cascade systems with unknown nonlinearities. *IEEE Trans. Automatic Control* 56 (9), 2224–2230. doi:10.1109/tac.2011.2157399
- Bechlioulis, C. P., and Rovithakis, G. A. (2014). A low-complexity global approximation-free control scheme with prescribed performance for unknown pure feedback systems. *Automatica* 50 (4), 1217–1226. doi:10.1016/j.automatica.2014.02.020
- Chen, Q., Tao, M., He, X., and Tao, L. (2021). Fuzzy adaptive nonsingular fixed-time attitude tracking control of quadrotor UAVs. *IEEE Trans. Aerosp. Electron. Syst.* 57 (5), 2864–2877. doi:10.1109/taes.2021.3067610
- Chen, S., Zhao, J., and Yu, Z. (2019). “Adaptive sliding control for lower exoskeleton robot driven by the series elastic actuator,” in *2019 IEEE international conference on mechatronics and automation (ICMA)*. IEEE, 2439–2444.
- Deng, W., Zhou, H., Zhou, J., and Yao, J. (2022). Neural network-based adaptive asymptotic prescribed performance tracking control of hydraulic manipulators. *IEEE Trans. Syst. Man, Cybern. Syst.* 53 (1), 285–295. doi:10.1109/tsmc.2022.3178626
- Dong, H., and Yang, X. (2022). Finite-time prescribed performance control for space circumnavigation mission with input constraints and measurement uncertainties. *IEEE Trans. Aerosp. Electron. Syst.* 58 (4), 3209–3222. doi:10.1109/taes.2022.3148356
- Gao, S., Liu, X., Jing, Y., and Dimirovski, G. (2021). Finite-time prescribed performance control for spacecraft attitude tracking. *IEEE/ASME Trans. Mechatronics* 27 (5), 3087–3098.
- Gao, Z., Zhang, Y., and Guo, G. (2022). Finite-time fault-tolerant prescribed performance control of connected vehicles with actuator saturation. *IEEE Trans. Veh. Technol.* 72 (2), 1438–1448. doi:10.1109/tvt.2022.3209802
- Goodwin, G. C., and Sin, K. S. (2009). *Adaptive filtering prediction and control*. Dover Publications, Inc.
- Kim, S. K., and Ahn, C. K. (2018). Self-tuning proportional-type performance recovery property output voltage-tracking algorithm for DC–DC boost converter. *IEEE Trans. Industrial Electron.* 66 (4), 3167–3175. doi:10.1109/tie.2018.2849982
- Li, X., Pan, Y., Chen, G., and Yu, H. (2017). Multi-modal control scheme for rehabilitation robotic exoskeletons. *Int. J. Robotics Res.* 36 (5-7), 759–777. doi:10.1177/0278364917691111
- Liang, J., Wang, F., Wang, Z., and Liu, X. (2019). Minimum-variance recursive filtering for two-dimensional systems with degraded measurements: boundedness and monotonicity. *IEEE Trans. Automatic Control* 64 (10), 4153–4166. doi:10.1109/tac.2019.2895245
- Liang, Z., Wang, Z., Zhao, J., Wong, P. K., Yang, Z., and Ding, Z. (2022). Fixed-time prescribed performance path-following control for autonomous vehicle with complete unknown parameters. *IEEE Trans. Industrial Electron.* 70 (8), 8426–8436. doi:10.1109/tie.2022.3210544
- Liu, D., and Yang, G. H. (2018). Performance-based data-driven model-free adaptive sliding mode control for a class of discrete-time nonlinear processes. *J. Process Control* 68, 186–194. doi:10.1016/j.jprocont.2018.06.006
- Liu, D., and Yang, G. H. (2019a). Prescribed performance model-free adaptive integral sliding mode control for discrete-time nonlinear systems. *IEEE Trans. neural Netw. Learn. Syst.* 30 (7), 2222–2230. doi:10.1109/tnnls.2018.2881205
- Liu, D., and Yang, G. H. (2019b). Data-driven adaptive sliding mode control of nonlinear discrete-time systems with prescribed performance. *IEEE Trans. Syst. Man, Cybern. Syst.* 49 (12), 2598–2604. doi:10.1109/tsmc.2017.2779564
- Mir, A. S., and Senroy, N. (2020). Self-tuning neural predictive control scheme for ultrabattery to emulate a virtual synchronous machine in autonomous power systems. *IEEE Trans. neural Netw. Learn. Syst.* 31 (1), 136–147. doi:10.1109/tnnls.2019.2899904
- Nguyen, M. L., Chen, X., and Yang, F. (2017). Discrete-time quasi-sliding-mode control with prescribed performance function and its application to piezo-actuated positioning systems. *IEEE Trans. Industrial Electron.* 65 (1), 942–950. doi:10.1109/tie.2017.2708024
- Rosolia, U., and Borrelli, F. (2017). Learning model predictive control for iterative tasks: a data-driven control framework. *IEEE Trans. Automatic Control* 63 (7), 1883–1896. doi:10.1109/tac.2017.2753460
- Shi, Y., Xie, W., Zhang, G., and Silvestre, C. (2024). Event-Triggered quantitative prescribed performance neural adaptive control for autonomous underwater vehicles. *IEEE Trans. Syst. Man, Cybern. Syst.* 54, 3381–3392. doi:10.1109/tsmc.2024.3357252
- Siciliano, B., and Khatib, O. (2016). *Robotics and the handbook[M]//Springer handbook of robotics*. Cham: Springer International Publishing, 1–6.
- Sui, S., Chen, C. L. P., and Tong, S. (2021). Finite-time adaptive fuzzy prescribed performance control for high-order stochastic nonlinear systems. *IEEE Trans. Fuzzy Syst.* 30 (7), 2227–2240. doi:10.1109/tfuzz.2021.3077317
- Tajaddodianfar, F., Moheimani, S. O. R., and Randall, J. N. (2019). Scanning tunneling microscope control: a self-tuning PI controller based on online local

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript

## Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

barrier height estimation. *IEEE Trans. Control Syst. Technol.* 27 (5), 2004–2015. doi:10.1109/tcst.2018.2844781

Xi, R. P., Zhang, H. G., Huang, H., and Land Li, Y. H. (2024). Adaptive command filtered control for a class of random nonlinear systems under model-based event-triggered control design. *IEEE Trans. Syst. Man, Cybern. Syst.* 54, 5074–5084. doi:10.1109/TSMC.2024.3389994

Zhang, X., Liu, J., Gao, Q., and Ju, Z. (2020). Adaptive robust decoupling control of multi-arm space robots using time-delay estimation technique. *Nonlinear Dyn.* 100 (3), 2449–2467. doi:10.1007/s11071-020-05615-5

Zhang, X., Liu, J., Tong, Y., Liu, Y., and Ju, Z. (2021). Attitude decoupling control of semifloating space robots using time-delay estimation and supertwisting

control. *IEEE Trans. Aerosp. Electron. Syst.* 57 (6), 4280–4295. doi:10.1109/taes.2021.3094626

Zheng, D. D., Xie, W. F., Chai, T., and Fu, Z. (2016). Identification and trajectory tracking control of nonlinear singularly perturbed systems. *IEEE Trans. Industrial Electron.* 64 (5), 3737–3747. doi:10.1109/tie.2016.2645139

Zhou, P., Guo, D., Wang, H., and Chai, T. (2018). Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking. *IEEE Trans. neural Netw. Learn. Syst.* 29 (9), 4007–4021. doi:10.1109/tnnls.2017.2749412

## Proof: Appendix of Theorem 1

If the tracking error  $e(k)$  satisfies  $-\delta - \underline{\chi}_k \rho_k < e(k) < \bar{\chi}_k \rho_k + \delta$  at any time  $k$ , through the proposed controller (Equation 24), we want to demonstrate that one step ahead tracking error  $e(k+1)$  also satisfies the prescribed performance control:  $-\delta - \underline{\chi}_{k+1} \rho_{k+1} < e(k+1) < \bar{\chi}_{k+1} \rho_{k+1} + \delta$ . Thereafter, the control algorithm can be repeated in the next sampling cycle.

To demonstrate this result, based on Equations 18–21, 23, we have

$$e(k+1) = \frac{\bar{\chi}_{k+1}[(1-qT)\theta(k) + qT - \Lambda_s \text{sign}(s(k))] - \underline{\chi}_{k+1}}{(1-qT)\theta(k) + qT + 1 - \Lambda_s \text{sign}(s(k))} \rho_{k+1} + \xi(k+1) \tag{A1}$$

From Equations 21, A1, we can obtain Equation A2.

$$e(k+1) < \frac{\bar{\chi}_{k+1}[(1-qT)\theta(k) + qT + \Lambda_s] - \underline{\chi}_{k+1}}{(1-qT)\theta(k) + qT + 1 + \Lambda_s} \rho_{k+1} + \delta \tag{A2}$$

Obviously,  $\frac{\bar{\chi}_{k+1}[(1-qT)\theta(k) + qT + \Lambda_s] - \underline{\chi}_{k+1}}{(1-qT)\theta(k) + qT + 1 + \Lambda_s} \rho_{k+1}$  is a function with respect to  $\theta(k)$  and is monotonic increasing with  $\theta(k)$ . Therefore, we have

$$\begin{aligned} e(k+1) &< \lim_{\theta(k) \rightarrow \infty} \frac{\bar{\chi}_{k+1}[(1-qT)\theta(k) + qT - \Lambda_s] - \underline{\chi}_{k+1}}{(1-qT)\theta(k) + qT + 1 + \Lambda_s} \rho_{k+1} + \delta \\ &= \bar{\chi}_{k+1} \rho_{k+1} + \delta \end{aligned} \tag{A3}$$

With the help of Equations 21, A1, we have Equation A4.

$$e(k+1) > \frac{\bar{\chi}_{k+1}[(1-qT)\theta(k) + qT - \Lambda_s] - \underline{\chi}_{k+1}}{(1-qT)\theta(k) + qT + 1 - \Lambda_s} \rho_{k+1} - \delta \tag{A4}$$

Similar to the proof process of (Equation A3), we can infer that

$$e(k+1) > \lim_{\theta(k) \rightarrow 0} \frac{\bar{\chi}_{k+1}[qT - \Lambda_s] - \underline{\chi}_{k+1}}{qT + 1 - \Lambda_s} \rho_{k+1} - \delta = -\underline{\chi}_{k+1} \rho_{k+1} - \delta \tag{A5}$$

From (Equations A3, A5), it can be concluded that

$$-\delta - \underline{\chi}_{k+1} \rho_{k+1} < e(k+1) < \bar{\chi}_{k+1} \rho_{k+1} + \delta \tag{A6}$$