



Disturbance rejection sliding mode control for robots and learning design

Fangli Mou¹ · Dan Wu¹ · Yunfei Dong¹

Received: 25 May 2020 / Accepted: 15 February 2021 / Published online: 29 March 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

The control of a robot to achieve both good dynamic and static performance against external disturbances is a challenging task, especially when high speed and a wide range of motion is required. In this paper, a disturbance rejection sliding mode control (SMC) methodology is designed for a robot manipulator. This methodology synthesizes the SMC design with the active disturbance rejection control (ADRC) technique. An extended state observer is employed to estimate unknown disturbances, which is difficult to deal with in a conventional SMC design, and to simplify the SMC law design. A learning-based parameter tuning methodology is presented to autonomously obtain the control parameters offline. To develop a robust and transferring controller, a neural network is used to learn the joint actuation ability for the controller optimizing process. Compared with other state-of-the-art controllers, both numerical simulations and experiments of a 6-DOF robot are provided to demonstrate the proposed control method and design methodology. These results reveal that the proposed control method has a satisfying tracking performance and strong disturbance rejection ability.

Keywords Robotics · Controller design · Active disturbance rejection control (ADRC) · Sliding mode control (SMC)

1 Introduction

In recent years, robots have increasingly played a significant role in daily services and industrial operations. Generally, robots are always devised for accurate and prespecified continuous trajectory tracking in a structured environment or required human–robot interaction in an unstructured complicated environment. This results in potential trouble such as performance deterioration and contact impact with the working environment when unknown disturbances happened [1, 2]. Controlling a manipulator to guarantee a high tracking performance in the presence of unknown time-varying disturbances is still a challenge in the research community. In addition, conventional control approaches are often insufficient to deal with these problems effectively and require a lengthy design and complex tuning process [3]. Moreover, when unexpected contact/collision situations happened during many tasks such as assembly, grinding and deburring,

grasping, or manipulation of deformable and delicate objects, the accurate and rapid response ability against external disturbances essentially determines the feasibility and reliability of further robot application. On the other hand, the parameter tuning based on experiments makes the controller hard to achieve optimal state and inconvenient to implement. Exploring effective and practical methodologies for robot controller design with a strong disturbance rejection ability has been a main concern.

Many model-based control strategies have been developed to increase the tracking performance and reliability of robots [4–7]. These methods utilize exact robot dynamics for controller design and illustrate superior performance in simulation environments. However, it is well known that the development of a real-world easy-to-use robot system often suffers from many restrictions such as system modeling errors, environment uncertainties, and limited algorithm complexity. Therefore, modern control strategies based on an explicit model of a specific system mostly stay in the theoretical design and numerical simulation phase. The classic PID controller is the most widely used type in industrial applications. However, owing to its simple control law and limited parameter tuning range, it may be difficult for a PID controller to allow the robot to achieve both good dynamic

✉ Fangli Mou
mfl18@mails.tsinghua.edu.cn

¹ State Key Laboratory of Tribology, Department of Mechanical Engineering, Tsinghua University, Beijing, China

and static performance (the system performance after the transient process). In addition, when we need a robot to perform at high speed with a wide range of motion, the PID controller might no longer be effective or can even lead to instability of the controlled system [8].

To suppress disturbances, the sliding mode control (SMC) and disturbance rejection control strategies are effective owing to their strong robustness to unknown exogenous disturbances, parameter variations, and model perturbations [9–11]. The SMC is a widely utilized control strategy in real applications [12]; however, the traditional SMC cannot effectively handle fast variable disturbances and may cause a chattering phenomenon owing to modeling errors and uncertainties, which degrades its performance in robots control [13]. Wang et al. [14] proposed a robust SMC methodology for robotic systems with compliant actuators that employed a generalized proportional integral observer technique to estimate unknown disturbances. M. Van et al. [15] developed a tracking control approach for robot manipulators where an adaptive backstepping sliding mode control is used. However, these control methods are not designed for pure joint control, and hence the corresponding performance with six-joint simultaneous motion is not presented. With the increase in robot DOFs, the dynamic coupling increases significantly and the modeling accuracy decreases at the same time, which causes fast variable endogenous disturbances and makes robot control problems much more difficult. Thus, the conventional SMC limits its performance to real robot applications. In addition, the conventional design procedure requires prior knowledge and many tuning experiments.

Active disturbance rejection control (ADRC) is a type of disturbance rejection control method based on the PID concept [16, 17]. The ADRC method does not require an explicit plant model but designs a unique extended state observer (ESO) to estimate and compensate for the total disturbance before the plant output has an impact [18]. The ADRC method only needs to know some basic system information such as the order of the system and the control input/output. This has the advantages of strong disturbance rejection ability and strong control robustness. In recent years, the ADRC method has been widely used in servo control systems [19], industrial process control [20], aerospace [21], and other research fields [22–24], thus exhibiting a promising future in industrial applications.

However, ADRC has been less applied to robot manipulator control problems at present. Castaneda et al. [25] designed an adaptive controller based on ADRC to solve the trajectory tracking problem of a “Delta” parallel robot considering the uncertainty of the dynamics model. Talole et al. [26] designed an ESO-based feedback linearization controller for the trajectory tracking control of a flexible-joint robotic system. A rotary single-link robot experiment

indicated the efficacy of the ADRC approach. Xue et al. [27] integrated ADRC with an existing PD structure for the set-point tracking control of robots. The effectiveness of the proposed modularized ADRC design is tested with a 1-DOF rotary manipulator. Madonski [28] studied the problem of estimating and suppressing periodic disturbances in robot control. The framework of ADRC was used, and experiments on a 3-DOF torsional plant demonstrated the effectiveness of the proposed scheme. Ren et al. [29] proposed a collision detection method based on ontology sensors (encoder and torque sensor) for collaborative robots using the ESO approach. Dong et al. [30] proposed a cascaded torque controller with an ADRC velocity inner loop to improve the control quality of the joint torque. The authors proposed an efficient and simple robot controller based on the ADRC method to realize the rapid and stable trajectory tracking of a robot [31].

All of these studies show that the ADRC has a great potential for robot control. However, for robot controllers that need high speed and high precision, the conventional ADRC has a simple feedback law in which a residual estimate error causes system performance deterioration. In addition, the controller design methods such as parameter tuning are usually lengthy and based on experiments, which makes the obtained controller always achieve suboptimal performance and inconvenient to implement.

In summary, the design of a robot tracking controller needs to guarantee three major requirements: (1) fast transient response and high precision, (2) robustness to model uncertainties and strong disturbance rejection ability, and (3) a simple design and tuning process. Motivated by the above issues, in this work, we developed a practical and effective control method for a robot system’s trajectory tracking performance subjected to unknown time-varying disturbances. The main idea is to use the ADRC methodology to improve the robustness and accuracy of a traditional SMC controller. First, a tracking differentiator is used to obtain a smoothed reference position and velocity trajectory. Then, an ESO is employed to estimate and compensate for the model uncertainties and unknown time-varying disturbances toward simplifying the SMC law and thus improving the tracking accuracy and robustness of the robot system. Furthermore, a learning-based parameter tuning method is presented to autonomously obtain the control parameters offline. To obtain a robust and transferring controller (which means the obtained controller can have similar performance under uncertainties and can be easily transferred from simulation to the real robot), a multilayer perceptron (MLP) network [32] is used to learn the joint actuation ability. Simulations and experiments are conducted to validate the proposed control design methodology.

Succinctly, our main contributions of this paper are:

- Design of a practical and effective control method for a robot system’s trajectory tracking performance subjected to unknown time-varying disturbances.
- An autonomous learning-based controller design methodology is presented to obtain the optimal control parameters.
- An actuation network is proposed to learn the joint actuation ability in order to obtain a robust and transferring controller.

The rest of this study is organized as follows. A brief introduction of the dynamic model of a robot is given in Sect. 2. The design of the proposed disturbance-rejection SMC is presented in Sect. 3. The learning-based parameter tuning methodology, including a trained joint actuation network, is developed in Sect. 4. Numerical simulations and experimental results in Sect. 5 demonstrate the effectiveness of the proposed robust control method by comparing it with other three control methods. Finally, the conclusions are drawn in Sect. 6.

2 Disturbance rejection SMC

2.1 System dynamics modeling

Based on the Euler–Lagrangian method, the dynamics equations of an n -joint robot can be derived in terms of its joint variables as follows [33]:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + d(q, \dot{q}, \ddot{q}, t) = \tau \tag{1}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^{n \times 1}$, respectively, represent the joint angle, velocity, and acceleration; $\tau \in \mathbb{R}^{n \times 1}$ is the joint torque; $D(q) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite inertia matrix; $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ represents the nonlinear Coriolis and centrifugal forces acting on the system; $G(q) \in \mathbb{R}^{n \times 1}$ is the gravitational torque; and $d(q, \dot{q}, \ddot{q}, t) \in \mathbb{R}^{n \times 1}$ is the generalized system disturbance that contains the unmodeled system dynamics and external disturbances. For robot tracking control, the disturbances caused of \ddot{q} are less related [27] and \ddot{q} is usually slowly varying in planning, so we ignore

the influence of second derivative part \ddot{q} in the generalized system disturbance d in the following sections.

Defining the variables as $x_1 = q, u = \tau, y = x_1$, the system dynamics (1) can be written as the following state-space description:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = D^{-1}(x_1)(-C(x_1, x_2)x_2 - G(x_1) - d + u) \\ y = x_1. \end{cases} \tag{2}$$

2.2 Control strategy

In this section, we present a trajectory tracking control framework (disturbance rejection sliding mode control, or DRSMC) for robots with unknown time-varying disturbances. Generally, the DRSMC method consists of an observer-based SMC law and an ADRC-based control architecture. The SMC law provides the basic control torque for trajectory tracking, and the ADRC-based control architecture provides both control operational information used in the SMC law and online disturbance compensation. A block diagram of the DRSMC strategy is shown in Fig. 1.

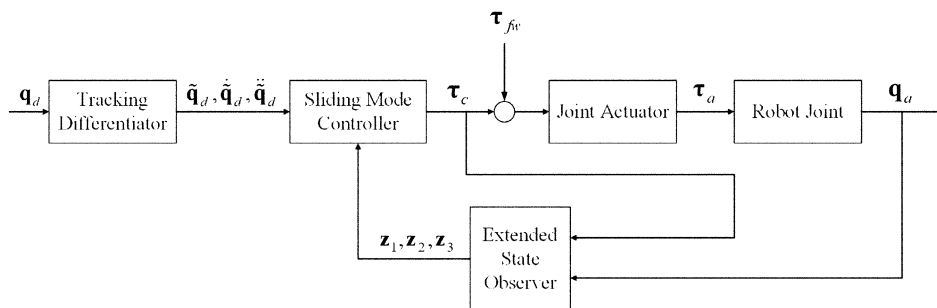
In Fig. 1, q_d is the given signal of the desired positions; $\tilde{q}_d, \dot{\tilde{q}}_d, \ddot{\tilde{q}}_d$ are reference trajectories obtained from the desired positions; q_a, \dot{q}_a are the actual joint angles and joint velocities; z_1, z_2, z_3 are the augmented system states; τ_c is the command control torque; and τ_{fw} is the feedforward torque generated by the feedforward controller.

The overall DRSMC scheme includes the tracking differentiator (TD) [17], extended state observer (ESO), and observer-based SMC law.

The TD is a preprocessing component that obtains the reference trajectory from the given signal of desired positions. A time-optimal differentiator can be obtained by solving the following equation:

$$\begin{cases} \dot{v}_1 = v_2 \\ \dot{v}_2 = -r \operatorname{sgn} \left(v_1 - v + \frac{v_2 |v_2|}{2r} \right), \end{cases} \tag{3}$$

Fig. 1 Control structure of DRSMC framework



where v_1 is the desired trajectory and v_2 is its derivative; the parameter r can be selected accordingly to speed up or slow down the transient profile. And the approximated discrete-time solution of Eq. (3) can be obtained as follows [17]

$$\begin{cases} v_1(t+h) = v_1(t) + h \cdot v_2(t) \\ v_2(t+h) = v_2(t) + h \cdot \text{fhan}(v_1(t) - v(t), c_1 v_2(t), r_0, h). \end{cases} \tag{4}$$

Using Eq. (4) twice, we can obtain the second derivative of the desired trajectory simultaneously

$$\begin{cases} q_1(t+h) = q_1(t) + h \cdot q_2(t) \\ q_2(t+h) = q_2(t) + h \cdot \text{fhan}(q_1(t) - q_d(t), c_1 q_2(t), r_0, h) \\ q_2'(t+h) = q_2'(t) + h \cdot q_3(t) \\ q_3(t+h) = q_3(t) + h \cdot \text{fhan}(q_2'(t) - q_2(t), c_2 q_3(t), r_1, h) \end{cases} \tag{5}$$

where $q_i = \tilde{q}_d^{(i-1)}$ are the generated reference trajectory for each joint, h is the controller instruction cycle, and $\text{fhan}(x_1, cx_2, r_0, h_0)$ is a nonlinear control function as follows:

$$\begin{cases} d = r_0 h_0^2, a_0 = h_0 c x_2, y = x_1 + a_0 \\ a_1 = \sqrt{d(d + 8|y|)} \\ a_2 = a_0 + \text{sgn}(y)(a_1 - d)/2 \\ s_1 = (\text{sgn}(y + d) - \text{sgn}(y - d))/2 \\ a = (a_0 + y - a_2)s_1 + a_2 \\ s_2 = (\text{sgn}(a + d) - \text{sgn}(a - d))/2 \end{cases}$$

$$\text{fhan}(x_1, cx_2, r_0, h_0) = -r_0(a/d - \text{sign}(a))s_2 - r_0 \text{sign}(a). \tag{6}$$

fhan is a time-optimal solution that guarantees the fastest convergence from generated reference trajectory to desired trajectory [34]. The parameter r_0 is called the tracking gain, which affects the rising speed of generated reference trajectory q_i and approximately determines the bandwidth of the TD. The parameter h_0 is a speed factor that eliminates high-frequency output oscillations and is usually set higher than the controller instruction cycle h , and c is the damping factor that determines the dynamic characteristic of the TD's transient tracking process. These parameters can be adjusted individually according to the desired speed and smoothness.

The basic idea of the ESO is to estimate the integrated system disturbance \mathbf{f}_w , which includes unmodeled dynamics and unknown time-varying disturbances. The ESO uses the control input and system output to augment the system

additional state. Considering robot dynamics (2), the integrated system disturbance \mathbf{f}_w can be given by

$$\mathbf{f}_w = -\mathbf{D}(\mathbf{x}_1)^{-1} \mathbf{d}. \tag{7}$$

Augmenting \mathbf{f}_w as a system additional state \mathbf{x}_3 , system (2) can be expressed in the linear augmented state-space form as

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 = \mathbf{x}_3 + \mathbf{D}^{-1} \mathbf{u} \\ \dot{\mathbf{x}}_3 = \mathbf{f}_w \\ \mathbf{y} = \mathbf{x}_1. \end{cases} \tag{8}$$

According to the above-mentioned ADRC design methodology, a third-order linear ESO can be designed to estimate the integrated system disturbance.

$$\begin{cases} \mathbf{e} = \mathbf{z}_1 - \mathbf{y} \\ \dot{\mathbf{z}}_1 = \mathbf{z}_2 - \beta_1 \cdot \mathbf{e} \\ \dot{\mathbf{z}}_2 = \mathbf{z}_3 - \beta_2 \cdot \mathbf{e} + \mathbf{b}_0 \cdot \mathbf{u} \\ \dot{\mathbf{z}}_3 = -\beta_3 \cdot \mathbf{e}, \end{cases} \tag{9}$$

where \mathbf{b}_0 is the estimated value of control amplification \mathbf{D}^{-1} , \mathbf{e} is the estimate error of the joint angles, \mathbf{u} is the control torque, \mathbf{y} is the actual joint angle, $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ are the estimated states of $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, respectively, and $\beta_1, \beta_2, \beta_3$ are the diagonal observer gain matrices of the ESO. Defining $\beta_{1i} = \beta_1(i, i), \beta_{2i} = \beta_2(i, i), \beta_{3i} = \beta_3(i, i), i = 1, 2, \dots, n$, increasing $\beta_{1i}, \beta_{2i}, \beta_{3i}$ can reduce the estimated error and accelerate the convergence. However, a greater $\beta_{1i}, \beta_{2i}, \beta_{3i}$ means that the ESO is more sensitive to system noise. Furthermore, $\beta_{1i}, \beta_{2i}, \beta_{3i}$ can be chosen as follows during preliminary design work by a pole-placement method [35]:

$$\beta_{1i} = 3\omega_{oi}, \beta_{2i} = 3\omega_{oi}^2, \beta_{3i} = \omega_{oi}^3, \tag{10}$$

where the tuning parameter ω_{oi} is the respective observer bandwidth.

Remark 1: Considering a linear ESO system (9), if integrated system disturbance \mathbf{f}_w is under the assumption that \mathbf{f}_w is bounded and continuously differentiable, and the observer gains $\beta_{1i}, \beta_{2i}, \beta_{3i}$ satisfy $\beta_{1i}, \beta_{2i}, \beta_{3i} > 0$, and $\beta_{1i}\beta_{2i} > \beta_{3i}$, then the estimate errors are bounded [36]. This assumption is practical for a real robot system as the physical energy is limited and the mechanical system has a characteristic to filter the physical signal. We use these results as the stability constraints for ESO design in Sect. 4.

We now introduce the observing errors:

$$\mathbf{E} = [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \boldsymbol{\eta}_3]^T = [\mathbf{z}_1 - \mathbf{x}_1, \mathbf{z}_2 - \mathbf{x}_2, \mathbf{z}_3 - \mathbf{x}_3]^T.$$

Then, the estimated states can be written as

$$\mathbf{z}_1 = \mathbf{x}_1 + \boldsymbol{\eta}_1, \mathbf{z}_2 = \mathbf{x}_2 + \boldsymbol{\eta}_2, \mathbf{z}_3 = \mathbf{x}_3 + \boldsymbol{\eta}_3. \tag{11}$$

Under the assumption that $\dot{\mathbf{f}}_w$ is bounded, the bounds of $\lim_{t \rightarrow \infty} \mathbf{E}$ yield [37]

$$\lim_{t \rightarrow \infty} |\mathbf{E}_i| \leq \begin{bmatrix} 0 \\ (1 - \frac{1}{\lambda_i} + \frac{1}{\lambda_i^2})M_i \\ \frac{3}{-\lambda_i}M_i \end{bmatrix}, \tag{12}$$

where \mathbf{E}_i is the submatrix of \mathbf{E} with respect to the i -th joint variables, for example, $\mathbf{E}_1 = [z_1 - x_1, z_2 - x_2, z_3 - x_3]^T$ where $z_i = \mathbf{z}_i(1), x_i = \mathbf{x}_i(1)$ are the corresponding states of joint 1. M_i is the i -th component of the upper bound $\sup |\dot{\mathbf{f}}_w|$, and λ_i is the maximum real eigenvalue of the error system matrix \mathbf{E}_i .

Denote by $\boldsymbol{\sigma}_3$ the solutions of (12), where,

$$\boldsymbol{\sigma}_3 = \left[\frac{3}{-\lambda_1}M_1, \frac{3}{-\lambda_2}M_2, \dots, \frac{3}{-\lambda_n}M_n \right]^T. \tag{13}$$

Then, we obtain,

$$\sup |\boldsymbol{\eta}_3| \leq \boldsymbol{\sigma}_3. \tag{14}$$

It should be noted that the bounds in Eq. (14) are a rather loose result, and that more accurate bounds can be obtained by assuming that disturbances that occur in engineering applications have typical forms. In addition, using appropriate numerical simulations can also help to determine more precise error bounds for a specific ESO.

Remark 2: The system model and some prior knowledge can be eliminated from the augmented system state \mathbf{x}_3 when the corresponding parts are eliminated from the ESO input \mathbf{u} simultaneously. This makes the ESO easily modifiable after previous design. Generally, having a more accurate system model and prior knowledge means fewer unknown uncertainties exist in the system. Thus, the ESO will produce more precise estimated results and require fewer observer gains or corresponding observer bandwidth.

The observer-based SMC law is designed to realize trajectory tracking. The sliding mode surface \mathbf{s} for robot system (1) is given by

$$\mathbf{s} = \mathbf{c}_1 \mathbf{e}_1 + \mathbf{e}_2, \tag{15}$$

where $\mathbf{e}_1 = \mathbf{q}_d - \mathbf{q}_a, \mathbf{e}_2 = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}_a$ represents the state tracking errors. $\mathbf{c}_1 = \text{diag}(c_{11}, c_{12}, \dots, c_{1n}), c_{11}, c_{12}, \dots, c_{1n}$ are constant sliding-mode surface parameters, and $c_{11}, c_{12}, \dots, c_{1n} > 0$.

Then, the approach law is

$$\dot{\mathbf{s}} = -\boldsymbol{\xi} \text{sgn}(\mathbf{s}) - \mathbf{k}\mathbf{s}, \tag{16}$$

where $\boldsymbol{\xi} = \text{diag}(\xi_1, \xi_2, \dots, \xi_n), \xi_1, \xi_2, \dots, \xi_n > 0$ and $\mathbf{k} = \text{diag}(k_1, k_2, \dots, k_n), k_1, k_2, \dots, k_n > 0$.

Substituting (9), (15) and (16) from (1), the observer-based SMC law is designed as follows:

$$\boldsymbol{\tau} = \mathbf{D}(\mathbf{c}_1 \dot{\mathbf{e}}_1 + \ddot{\mathbf{q}}_d + \boldsymbol{\xi} \text{sgn}(\mathbf{s}) + \mathbf{k}\mathbf{s}) + \mathbf{C}_0(\mathbf{q}, \dot{\mathbf{q}}\dot{\mathbf{q}} + \mathbf{G}_0(\mathbf{q}) - \mathbf{D}\mathbf{z}_3 + \mathbf{f}_c, \tag{17}$$

where $\mathbf{C}_0(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{G}_0(\mathbf{q})$ are the nominal system models of $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{G}(\mathbf{q})$, and \mathbf{f}_c is the estimated bound of the system error chosen as

$$\mathbf{f}_c = \mathbf{D}((\boldsymbol{\sigma}_3 + \mathbf{f}_u) \odot \text{sgn}(\mathbf{s}) - \mathbf{f}_l). \tag{18}$$

where \odot is the Hadamard product operator that represents the elementwise product of two matrices. \mathbf{f}_u and \mathbf{f}_l are the estimated upper bound and estimated lower bound of the initial states, respectively. Hence, $\mathbf{f}_u \geq \mathbf{f}_l$. A larger \mathbf{f}_c can cause greater chatter when defined errors $\mathbf{e}_1, \mathbf{e}_2$ arrive near sliding surface $\mathbf{s} = \mathbf{0}$. To obtain better control quality, we can include some decay factors $\zeta_i(t)$ to revise the estimated \mathbf{f}_c as

$$\mathbf{f}'_c = \mathbf{D}((\zeta_1(t)\boldsymbol{\sigma}_3 + \zeta_2(t)\mathbf{f}_u) \odot \text{sgn}(\mathbf{s}) - \zeta_3(t)\mathbf{f}_l), \tag{19}$$

where $\zeta_1(t)$ is monotonically decreasing, and $\zeta_2(t), \zeta_3(t)$ can be chosen as a piecewise function for which $\zeta_2(t), \zeta_3(t) = 0$ when $t \geq t_0$. t_0 is a given time.

Remark 3: The system model and some prior knowledge can be eliminated from the augmented system state \mathbf{x}_3 when the corresponding parts are also eliminated from the ESO input \mathbf{u} simultaneously. This makes the ESO easily modifiable after previous design. Generally, a more accurate system model and prior knowledge means that fewer unknown uncertainties exist in the system. Thus, the ESO produces more precise estimated results and requires fewer observer gains or corresponding observer bandwidth.

Summing up the above analysis, the DRSMC method illustrated in Fig. 1 can be obtained.

2.3 Stability analysis

Theorem 1: Considering a robot system (1) under the bounded time-varying disturbances, the observer-based SMC scheme in the form of (9), (15), (16), and (17) is given. If the observer (9) has bounded estimate errors, the tracking error of system (1) will converge to the desired equilibrium point asymptotically.

Proof: Combining (15) and (17), the derivative of the sliding surface (15) can be rewritten as follows:

$$\begin{aligned}\dot{\mathbf{s}} &= \mathbf{c}_1 \dot{\mathbf{e}}_1 + \dot{\mathbf{e}}_2 = \mathbf{c}_1 \dot{\mathbf{e}}_1 + \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} \\ &= \mathbf{c}_1 \dot{\mathbf{e}}_1 + \ddot{\mathbf{q}}_d - \mathbf{D}^{-1}(-\mathbf{f} - \mathbf{d}) - \mathbf{D}^{-1}\boldsymbol{\tau} \\ &= -\xi \text{sgn}(\mathbf{s}) - \mathbf{k}\mathbf{s} - \mathbf{D}^{-1}(-\mathbf{f} - \mathbf{d} - \mathbf{C}_0(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \\ &\quad - \mathbf{G}_0(\mathbf{q}) + \mathbf{f}_c) + \mathbf{z}_3.\end{aligned}\quad (20)$$

According to (11), we have

$$\dot{\mathbf{s}} = -\xi \text{sgn}(\mathbf{s}) - \mathbf{k}\mathbf{s} - \mathbf{D}^{-1}\mathbf{f}_c + \boldsymbol{\eta}_3. \quad (21)$$

Consider the following Lyapunov function:

$$V(\mathbf{s}) = \frac{1}{2}\mathbf{s}^T\mathbf{s}. \quad (22)$$

The derivative of $V(\mathbf{s})$ yields

$$\begin{aligned}\dot{V}(\mathbf{s}) &= \mathbf{s}^T \dot{\mathbf{s}} = \mathbf{s}^T(-\xi \text{sgn}(\mathbf{s}) - \mathbf{k}\mathbf{s} - \mathbf{D}^{-1}\mathbf{f}_c + \boldsymbol{\eta}_3) \\ &= -\mathbf{s}^T \xi \text{sgn}(\mathbf{s}) \\ &\quad - \mathbf{s}^T \mathbf{k}\mathbf{s} - \mathbf{s}^T((\boldsymbol{\sigma}_3 + \mathbf{f}_u) \odot \text{sgn}(\mathbf{s}) - \mathbf{f}_l - \boldsymbol{\eta}_3) \\ &= -\sum_{i=1}^n \xi_i |s_i| - \sum_{i=1}^n k_i s_i^2 \\ &\quad - \sum_{i=1}^n |s_i|((\boldsymbol{\sigma}_3)_i + (\mathbf{f}_u)_i - \text{sgn}(s_i)((\mathbf{f}_l)_i + (\boldsymbol{\eta}_3)_i)). \\ &\leq -\sum_{i=1}^n |s_i|((\boldsymbol{\sigma}_3)_i + (\mathbf{f}_u)_i - |(\mathbf{f}_l)_i| - |(\boldsymbol{\eta}_3)_i|) \\ &\leq 0\end{aligned}\quad (23)$$

This means the defined errors $\mathbf{e}_1, \mathbf{e}_2$ arrive at sliding surface $\mathbf{s} = \mathbf{0}$ in finite time. The sliding motion is then described as

$$\mathbf{c}_1 \mathbf{e}_1 + \mathbf{e}_2 = \mathbf{0}. \quad (24)$$

Since $\mathbf{c}_1 > 0$, the system (24) can be verified as exponentially stable. This shows that the tracking error will slide to the equilibrium point asymptotically under the proposed DRSMC control law. This completes the proof.

3 Learning-based controller design methodology

3.1 Optimal design of DRSMC parameters

From the analysis in the previous section, we can see that the proposed DRSMC has much more control parameters than that of the PID controller. In summary, the parameters in the TD tracker are the speed factor r_0 , filter factor h_0 , and damping factor c . The parameters in the ESO are $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3$, and \mathbf{b}_0 ; and the parameters in the observer-based SMC law are \mathbf{c}_1, ξ , and \mathbf{k} . The choice of different control parameters can greatly impact the closed-loop performance of the controlled system.

Among these parameters, speed factor r_0 , filter factor h_0 , and damping factor c can be easily chosen according to the rapidity requirement and the maximum acceleration that the system can actually provide. However, the other parameters need to be well-designed to obtain a satisfying result, and this design process is generally realized by utilizing empirical formulas or manual tuning.

Since the proposed robot controller directly controls the joint torques, we expect that the parameter tuning progress should be offline and automatic to reduce the potential risk of robot online operation. According to the separation principle [34], we can design the control parameters of the ESO and SMC law based on the system response. Then, these parameters are further adjusted according to the complete closed-loop system response.

In this paper, a genetic algorithm (GA) method [38] is used to optimize the residual control parameters offline according to the control objective function. This process only needs an excitation source that can achieve stable motion of the robot system. The excitation source we used here is a simple PID controller. Since the controller only needs to consider the system stability, it is very easy to design and implement. With an excitation source, we can let the robot track a group of desired trajectories and collect the control torques as well as the corresponding joint states including the joint angles, velocities, and torques. Note that the desired trajectories do not need to be well-designed when optimizing the control parameters; however, the actual joint trajectories should cover a relatively wide range of frequency spectra when training the actuator model [3].

The complete DRSMC parameter design process is as follows:

We want the used ESO can give fast and precise estimation with a certain degree of noise suppression ability, to optimize the ESO parameters in Eq. (8), we use an objective function as follows:

Algorithm 1	Optimal design of DRSMC parameters.
1:	Design a stable excitation source to achieve stable robot motion (we used a PID controller here, denoted as Controller 0);
2:	Let the robot track a group of desired trajectories by Controller 0 (we chose 10 sinuous desired trajectories with different frequencies here), and collect the control torques and the corresponding joint states;
3:	Optimize the ESO parameters offline according to the control torques and corresponding joint states collected in step 2, and average the obtained ESO parameters to get the optimal ESO;
4:	Train a neural network to learn the actuation model according to the collected data in step 2;
5:	Optimize the SMC law parameters offline utilizing the results in step 3 and step 4;
6:	Adjust these parameters according to the closed-loop system performance online.

$$\begin{aligned}
 J(n) &= \lambda_1 J_1(n) + \lambda_2 J_2(n) \\
 &= \frac{\lambda_1}{T} \int_{t=0}^T (\lambda_{01} t \|\eta_1(t, n)\|_1 + \lambda_{02} t \|\eta_2(t, n)\|_1) dt \\
 &\quad + \lambda_2 \left\| \frac{\mathbf{u}(t, n)}{\lambda_{03} \mathbf{b}_0(n) \mathbf{z}_3(t, n)} + \frac{\lambda_{03} \mathbf{b}_0(n) \mathbf{z}_3(t, n)}{\mathbf{u}(t, n)} \right\|_\infty.
 \end{aligned} \tag{25}$$

$$\begin{aligned}
 J_1(n) &= \frac{1}{T} \int_{t=0}^T (\epsilon_{01} \|\mathbf{e}_1(t, n)\|_2 + \epsilon_{02} \|\mathbf{e}_2(t, n)\|_2) dt \\
 J_2(n) &= \max \left(\left| \frac{\mathbf{e}_1(t, n)}{\mathbf{q}_d(t, n)} \right| \right), \text{ in } t \in [T_0, T]. \\
 J_3(n) &= \|\mathbf{e}_1(t, n)\|_1, \text{ in } t \in [T - \Delta T, T]
 \end{aligned} \tag{27}$$

where $\eta_i(t, n)$, $\mathbf{z}_3(t, n)$, $\mathbf{b}_0(n)$, $\mathbf{u}(t, n)$ represent the ESO estimate errors, augmented state, control amplification, and control torque for joint n , respectively; T is the simulation running time; and $\lambda_1, \lambda_2, \lambda_{01}, \lambda_{02}, \lambda_{03}$ are the given weighted factors.

In Eq. (25), J_1 is the optimization objective part, which reflects the ESO ability for fast and precise estimation. J_2 is the regularization part, which is designed to improve the disturbance suppression ability of the obtained ESO and also prevent the parameters from increasing boundlessly in the optimization process. Then, the optimization problem can be solved by using the GA method under the parameter stability constraints.

For control law design, we want the controlled system can have a fast and smooth transient period with small steady-state error. Similarly, the following objective function is constructed to optimize the SMC law parameters in Eq. (17) offline:

$$J_e(n) = \lambda_1 J_1(n) + \lambda_2 J_2(n) + \lambda_3 J_3(n). \tag{26}$$

where

In Eq. (27), J_1 reflects the general control ability considering the complete tracking process; J_2 represents the maximum relative tracking error for $t > T_0$, T_0 is a given time to eliminate insensitive errors in the initial tracking; J_3 represents the steady-state error; and $\lambda_1, \lambda_2, \lambda_3, \epsilon_{01}$, and ϵ_{02} are the given weighted factors.

When optimizing the SMC law parameters, we found that the given desired trajectory must make a difference after several iterations in optimization. Otherwise, the obtained control parameters make the robot performance sensitive to the desired input, which means the system can only achieve good performance in tracking this training trajectory and lacks robustness to disturbances and unknown uncertainties. Changing the trajectory amplitude and using different trajectory forms in optimization can help to obtain robust parameters.

Note that this design methodology is a general framework that can be used for other controllers such as PID, ADRC, and SMC. The offline tuning result can also provide the guidance for online tuning with the optimal performance would get.

3.2 Modeling the actuation ability

In this paper, the actuation ability (see Fig. 2) is the control-to-torque relationship that includes all communication delays, current-loop ability, measurement noise, and hardware dynamics within one control loop. As an analytical

Fig. 2 Actuation ability and training of actuation network

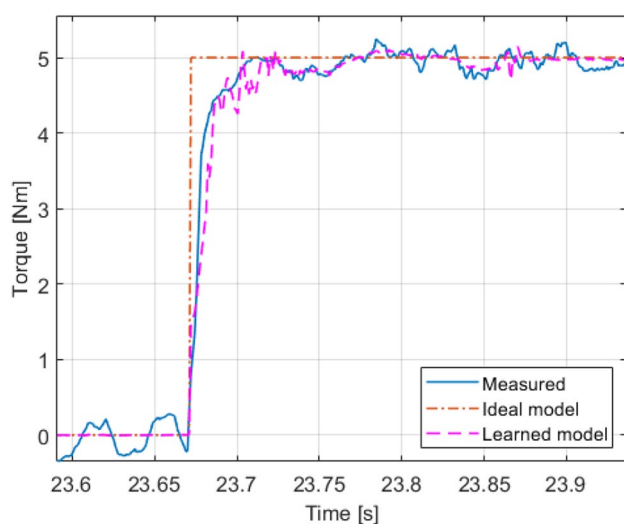
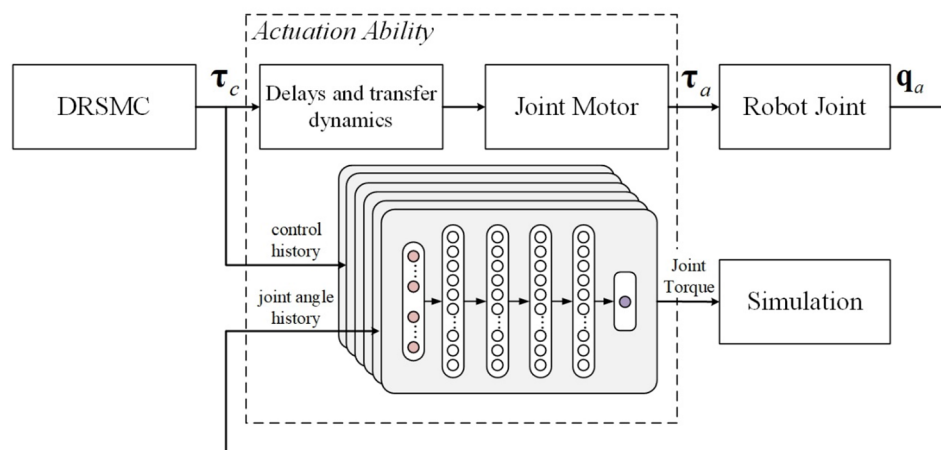


Fig. 3 Validation of learned actuation network for commanded torque response

actuation model is extremely difficult to describe [3], we used supervised learning to train an actuation network that outputs an estimated torque at the joints given a history of command control torques and the joint velocities. Note that this obtained actuation network is only used in simulations placed between the control output and the joint input torque to make the simulation model more realistic. This MLP network can provide lightly jitter output for controller training, that can prevent premature convergence and improve the robustness of obtained controller. We assumed that the joint actuations are independent of each other; hence, we trained the network of each joint separately.

More precisely, we used an MLP with 4 hidden layers of 16 units each, as shown in Fig. 2. A history consisting of the last 10 sampling periods of the command control torques and joint velocities that correspond to each training data has a history of the last 0.01-s command control torques

and joint velocities in this work. The length of each training data should be neither too long nor too short, as a too-dense history can make the training model more prone to overfitting and computationally more expensive. In addition, the length of the history should be sufficiently longer than the system communication delays and the mechanical response time, which is about three to four sampling periods in our system. For the features that were chosen, we found that the joint angle information is of no help in training this actuation network. By contrast, the joint velocity information is a necessary feature in this problem.

The dataset contains more than 500,000 samples, which are mentioned in the above section. About 80% of the generated data were used for training, and the rest were used for validation. We choose the commonly used ELU (Exponential Linear Unit [39]) activation function in the MLP network. The root mean square (RMS) of the prediction error is used to evaluate the trained actuation network. Training one network takes about 3 h on one NVIDIA 12G GTX1050 Ti GPU.

The validation result with the obtained actuation network is shown in Fig. 3, where the ideal model has a zero communication delay, zero mechanical response time, and infinite bandwidth. Hence, the model can generate any commanded torque instantly. It can be observed that the trained model can simulate the dynamic performance of the torque response and has an average absolute error of 0.297 Nm on the validation set, which is lower than that of the ideal model (0.578 Nm). Although the static torque performance cannot be predicted precisely (see the curve before the step occurs in Fig. 3), this is reasonable and acceptable for the simulation and controller design. Moreover, the trained actuation network can also add structural noise to the simulation system, which is considered an effective way to improve the training of a model [40].

We should note that the simulated system can hardly represent the real system perfectly, the modeling errors,

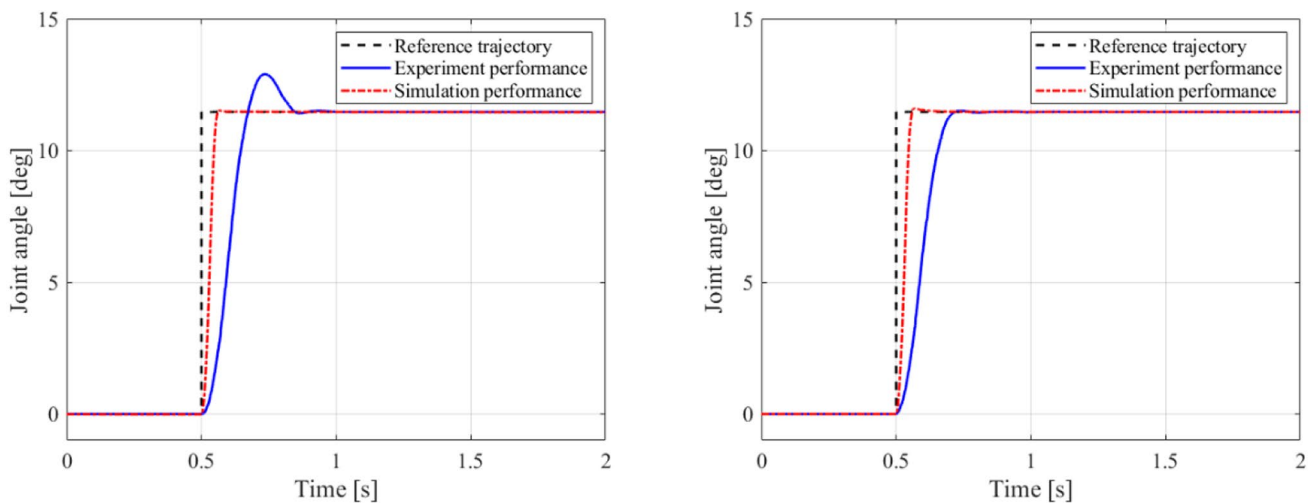


Fig. 4 Performance of trained controller without using the actuation network (left) and performance of the controller using the actuation network in training (right)

machining errors, the joint flexibility, the dynamics of joint gear reducer and other unmodeled dynamics would constrain the model’s accuracy. Therefore, the controller in simulation would suffer more constraints in the real robot system and the learning based offline training would get parameters which are not robust enough if we only use the robot model built by the MATLAB/SimMechanics toolbox. The MLP network can simulate the actuation ability (see Fig. 2) for training our controller, besides, this MLP network can provide lightly jitter output for controller training, that can prevent premature convergence and improve the robustness of obtained controller.

The controller parameters can be finally acquired by offline optimization using the trained actuation network. The performance of the obtained controller in simulation and experiment can be seen in Fig. 4. The settling time in the experiment is about 0.412 s, while that in the simulation is about 0.249 s. The overshoot in the experiment is about 0.001°, while that in the simulation is about 0.003°. As can be seen in Fig. 4, we can get a quite effective controller in simulation without using the actuation network; however, this controller may not obtain a satisfactory performance in the real robot. We can also see that using the actuation network in training can help to obtain a more robust controller, the simulation performance is similar and this controller performs much better in the real system. We can observe that the proposed learning-based autonomously design methodology is practical and effective for controlling a complex robot system.

4 Simulation and experimental results

In this section, the proposed control method is validated by simulation examples and experimental studies. Four comparative control strategies (conventional PID controller, conventional SMC method [41], linear ADRC method [31], and the proposed DRSMC method) were tested. Since the built-in PID controller cannot track the step input, we set the built-in parameters as initial values in PID optimization. All of these controllers’ parameters are optimally tuned by using the proposed methodology in Sect. 4. In this particular case, the weighted factors in Eq. (25–27) are chosen as

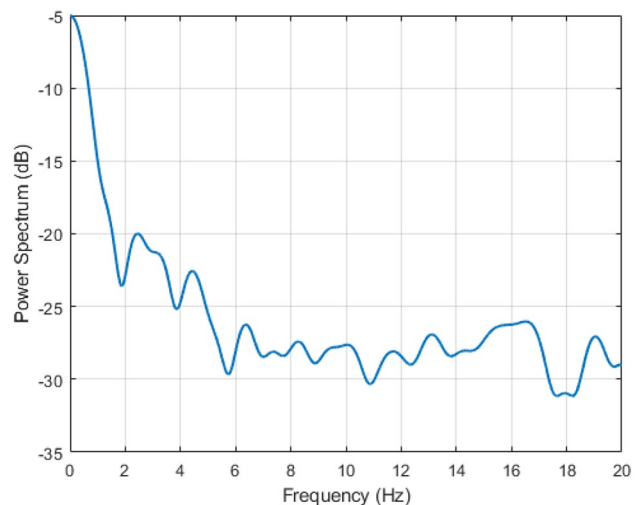


Fig. 5 Power spectrum of joint torque when external force is applied

$\lambda_1 = 2, \lambda_2 = 1, \lambda_{01} = 2,$
 $\lambda_{02} = 1, \lambda_{03} = 3$, in ESO optimization
 $\lambda_1 = 1, \lambda_2 = 10, \lambda_3 = 2000, \epsilon_{01} = 2,$
 $\epsilon_{02} = 1$, in control law optimization.

The primary parameters in the GA method are chosen as follows: population size is 400, maximum number of iterations is 400, crossover fraction is 0.8, mutation fraction is 0.2, and migration fraction is 0.2.

We note that the proposed method would not increase the computational burden obviously compared with the conventional PID method and SMC method [42, 43], the complete algorithm can be run well with 1 kHz frequency.

To realize feedforward compensation, we conduct dynamic identification for our robot using the method detailed in [44]. The joint friction is modeled as the following Coulomb viscous friction:

$$\tau_{f,n} = \mathbf{f}_{c1}(n)\text{sgn}(\dot{q}_n) + \mathbf{f}_{c2}(n)\dot{q}_n, \tag{28}$$

where $\tau_{f,n}$ represents the joint friction of joint n, and $\mathbf{f}_{c1}, \mathbf{f}_{c2} \in \mathbb{R}^{n \times 1}$ are the friction parameters.

The bandwidth of the external disturbances was tested to guide the design of simulations and experiments. In this case, we set the robot in zero-force mode, which means we only provide gravity compensation and manually drag the robot to move several trajectories. The frequency spectra of the joint torques under human force were analyzed to reveal the characteristics of potential external disturbances in our system. The spectrum of one joint torque can be shown in Fig. 5, in which the power spectrum decreases by about 3 dB below the 0-Hz value when the frequency is 0.5 Hz, decreases by about 10 dB when the frequency is 1.0 Hz,

and decreases by more than 20 dB when the frequency is 4.0 Hz, which means more than 90% of the power caused by external effect in joint space is centralized with the bandwidth of 4.0 Hz. This means external disturbances applied to the robot mainly generate disturbance torques with the bandwidth of 4.0 Hz in joint space. As all of these spectra have similar characteristics, the bandwidth of external disturbances in robot joint space can be approximately considered as 4.0 Hz.

4.1 Simulation results and discussion

The robot tracking process is simulated in the MATLAB/Simulink environment. The robot model of a 6-DOF manipulator (Fig. 9, which is used in our experiments) is set up using the MATLAB/SimMechanics toolbox, and the model physical parameters are set according to the given robot URDF file. The friction parameters of each joint in Eq. (28) are set according to the system dynamic identification results:

$$\mathbf{f}_{c1} = [5.6665, 2.951, 2.7750, 2.9656, 1.4458, 1.5185],$$

$$\mathbf{f}_{c2} = [10.4242, 13.1298, 9.6565, 3.5454, 2.4864, 2.0506],$$

To simulate the noise and communication delays in measurements, the feedback joint angles of the sensor were corrupted by zero-mean white noise with a standard deviation of 0.001° and 0.001 s of time delay. According to the robot manual, the joint torques are limited to corresponding rated torques of 85 Nm, 85 Nm, 40 Nm, 40 Nm, 10 Nm, and 10 Nm, respectively.

First, we compare the control performance of the control methods mentioned above. In this simulation case, all joints are expected to track the reference square-wave signals,

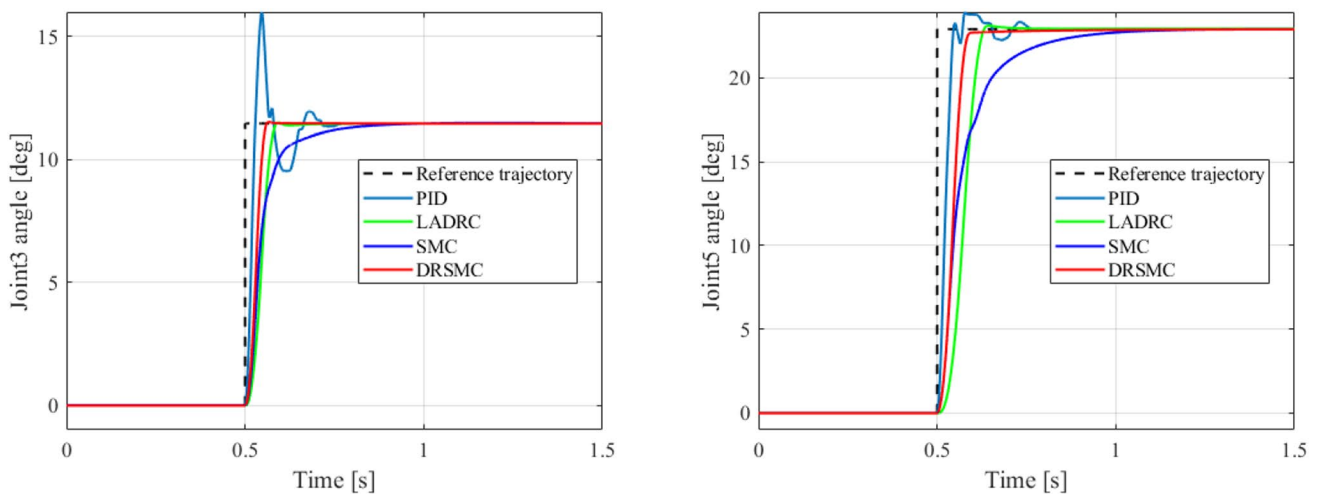


Fig. 6 Tracking trajectories of robot under different control methods

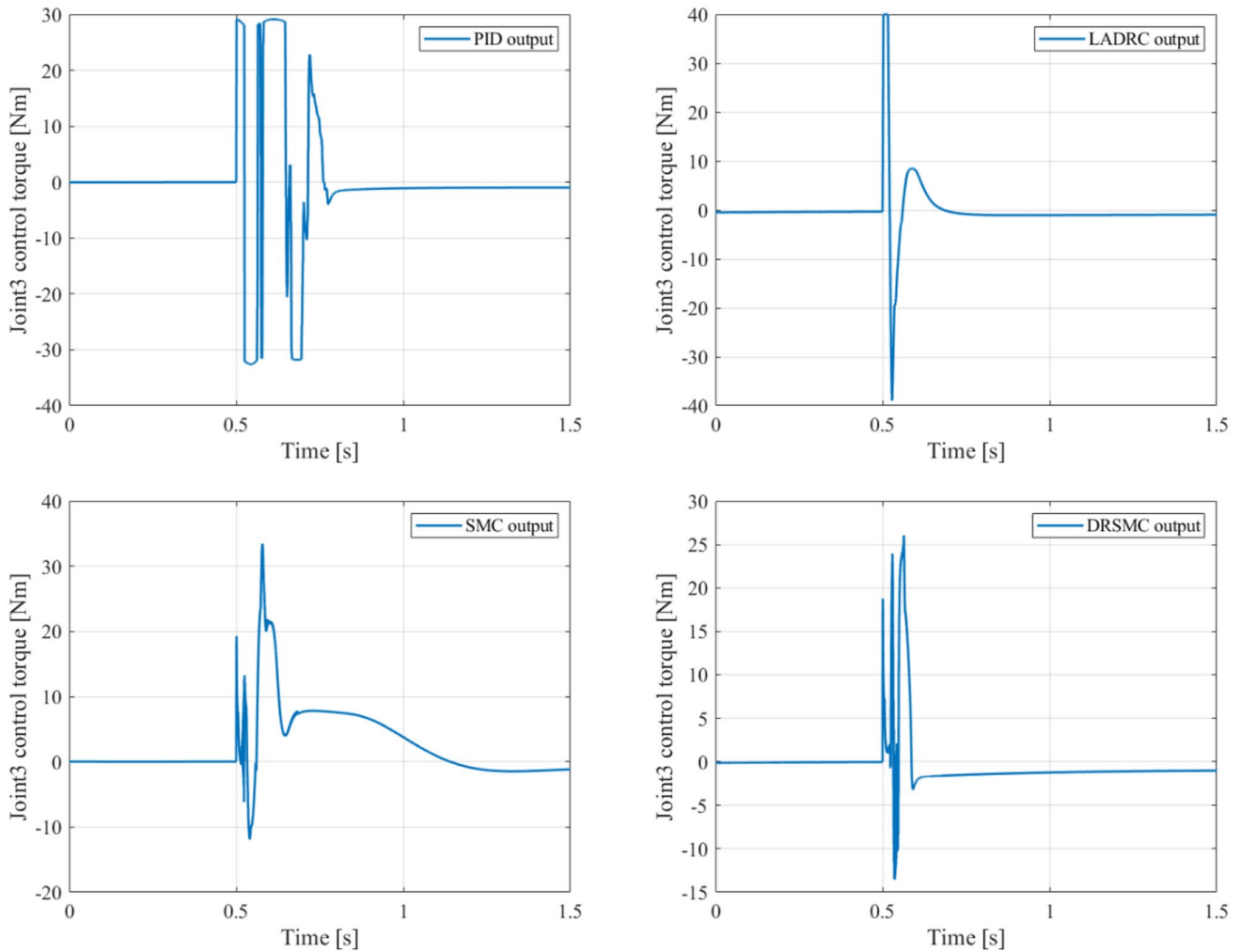


Fig. 7 Control torque (control input) of the comparative controllers

which have an amplitude greater than 10° (considered a wide range for robot to track), simultaneously. No extra disturbance torque is applied to the system in this case. Figure 6 shows the tracking curves of the joint angles under these comparative controllers. All joints of these controllers were optimally tuned using the same methodology in Sect. 4, and a description of the joints can be seen in Fig. 9.

As can be seen in Fig. 6, the PID controller exhibits good performance in rapidity but has a certain overshoot and relatively long settling time. The LADRC method also shows good performance in rapidity but has mostly no overshoot. The SMC method has a smooth tracking performance but the rising speed is a little slow. The DRSMC method provides the best tracking performance considering the rapidity and steady-state errors, which demonstrates that the proposed control strategy can achieve promising tracking performance. The control torque (or control input) of comparative controllers are shown in Fig. 7.

To illustrate the robustness of these controllers against unknown time-varying disturbances applied to the robot, a disturbance torque is applied to each joint simultaneously. The controller parameters remain the same, and corresponding reference trajectory is a sine wave (amplitude is 11.46° , and frequency is 1.0 Hz). We consider that a time-varying disturbance is the superposition of a sine-wave disturbance and constant disturbance, as follows:

$$dis_i(\text{Nm}) = \begin{cases} 0, & t < t_1 \\ Tq_i + Tq_i \sin(2\pi f_{dis}(t - t_1)), & \text{else} \end{cases} \quad (29)$$

where Tq_i is chosen as a 25% value of the maximum i -th joint output torque. The sine-wave disturbance frequency f_{dis} is chosen as 4 Hz, which is the bandwidth of potential external disturbances in our system (see Fig. 5). We should note that the higher frequency of disturbance applied, the larger the tracking error would have; here we give the results of a

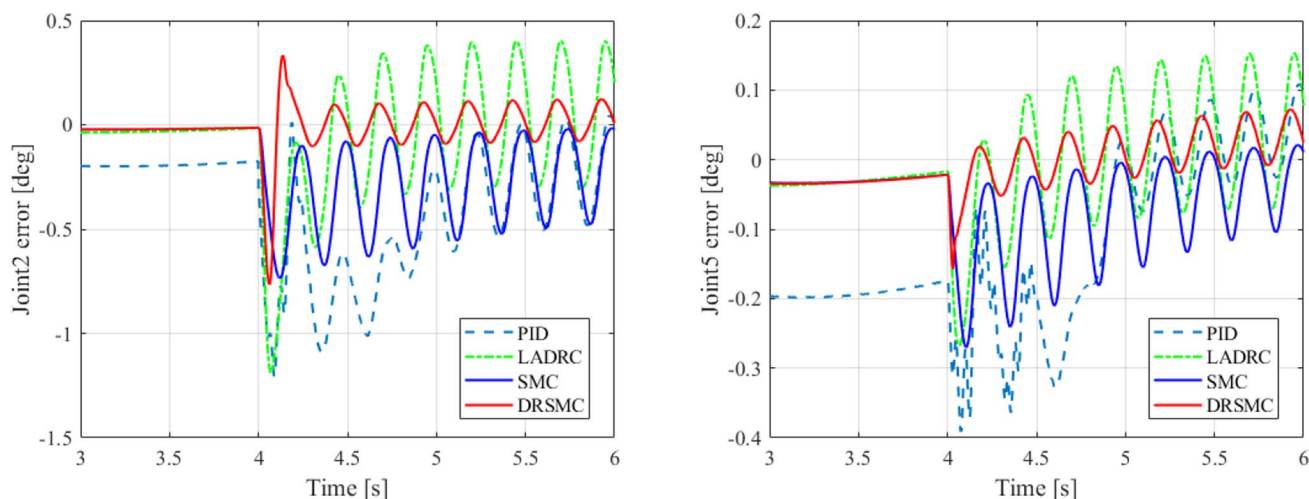


Fig. 8 Tracking trajectories of robot against unknown time-varying disturbances under different control methods

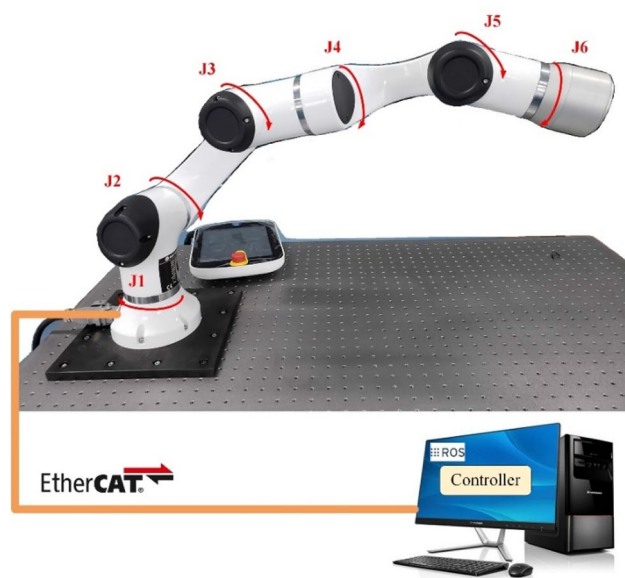


Fig. 9 Illustration of 6-DOF experimental robot system

disturbance with 4 Hz since it is the upper limit frequency of disturbances, which provides the worst condition for the robot controller.

The trajectory tracking curves of the joint errors against unknown time-varying disturbances are shown in Fig. 8, where the disturbance is applied at $t = 4.0$ s. We can observe that the DRSMC method provides the least peak-valley error and average error after a short settling time of suppressing disturbances. The simulation results verify that the proposed DRSMC method can adequately suppress a wide range of unknown time-varying disturbances compared to the PID, LADRC, and SMC methods.

4.2 Experimental results and discussion

The robot used in the experiments is a 6-DOF Elfin collaborative robot, as shown in Fig. 9. A PC-based controller is implemented to process data and control the robot directly via an EtherCAT bus. The real-time (RT) running frequency of the robot system is set to 1 kHz (the bus cycle is 1 ms). A control algorithm is employed using the ROS environment under an RT-Linux core. The parameters of the PID controller used in the data acquisition phase in Sect. 2 are set as $P = 10$, $I = 0.1$, and $D = 5$.

In the first experiment, we compare the control performance of different control methods with no extra disturbance torque applied to the system. All joints are expected to simultaneously track the reference square-wave signals in a wide range simultaneously. All controllers are further tuned with proper gains to provide a good performance with regard to the closed-loop tracking response considering a compromise between the response rapidity and steady-state error. The experimental results are shown in Fig. 10. In these figures, the dotted lines represent the reference trajectory of each joint, and the solid lines represent the joint tracking trajectories using comparative control strategies.

As can be seen in Fig. 10, the conventional PID controller can barely control the robot to track the trajectories with a large amplitude step change. This occurs because the required steady-state error and system stability restricted the parameters' tuning range. Generally speaking, utilizing the ADRC control scheme or the SMC approach can improve the system control performance in contrast to the PID controller. For further discussion, four representative step-response performance characteristics are calculated to compare these control strategies and are given in Table 1. t_r , t_s , $\sigma\%$, and e_{ss} are the average rise time (here, the rise time represents the

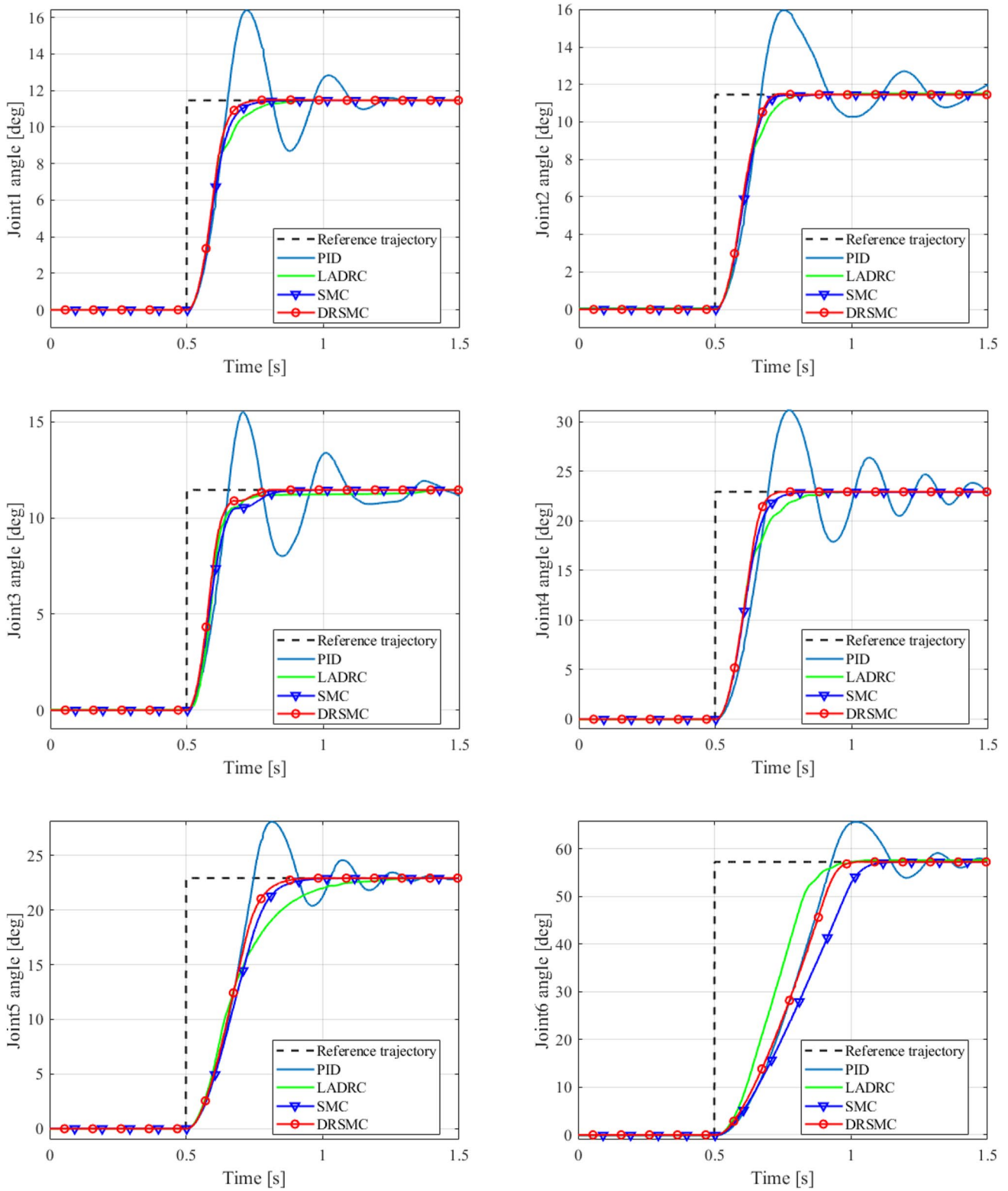


Fig. 10 Tracking trajectories of experiment robot under different control methods

first time that the response curve arrives at 90% of the stable-state value), average settling time within the 2% error band, average percentage overshoot, and average absolute

steady-state error of the six robot joints, respectively. These first three characteristics t_r , t_s , $\sigma\%$ show the dynamic performance and the characteristic e_{ss} demonstrates the static

Table 1 Step-response performance characteristics of four comparative control strategies

	PID	LADRC	SMC	DRSMC
t_r [s]	0.2097	0.2482	0.2442	0.2162
t_s [s]	0.9834	0.4096	0.3213	0.2763
$\sigma\%$	29.7891%	0.1747%	0.0001%	0.0984%
e_{ss} [°]	0.0300	0.0217	0.0159	0.0018

Table 2 Disturbance rejection ability of four comparative control strategies

	PID	LADRC	SMC	DRSMC
M_p [°]	1.1862	0.3480	0.1290	0.0749
M_{p-p} [°]	1.4449	0.2059	0.0865	0.0851
M_d [°]	0.4187	0.0457	0.0753	0.0015

performance. The control torque (or control input) of comparative controllers are shown in Fig. 11.

The results demonstrate that the LADRC method can greatly reduce the system settling time (by about 58.3%) and percentage overshoot (by about 29.6%) compared with the PID controller. Meanwhile, the steady-state error is reduced by about 27.6%. This shows that the usage of TD and ESO can balance the response rapidity and steady-state error because the discontinuous reference trajectories can be smoothed by TD to bound the calculated errors, and the ESO provides estimation and compensation of unknown disturbances. However, further decreasing the system steady-state error on the basis of maintaining a satisfying response rapidity can be difficult under the LADRC method since the ESO cannot accurately estimate and compensate the joint static friction. Then, the problem becomes similar to that for the PID controller.

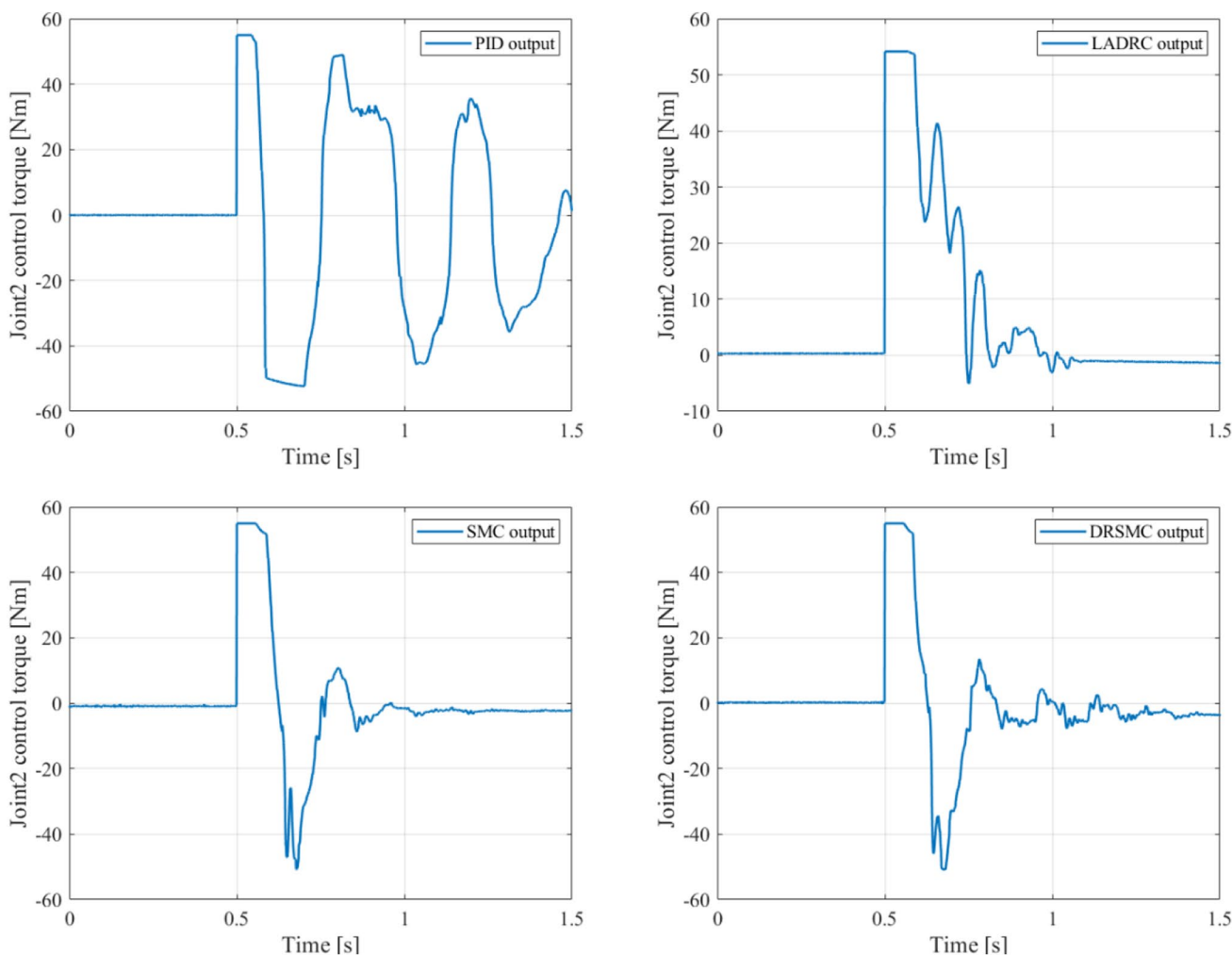


Fig. 11 Control torque (control input) of the comparative controllers

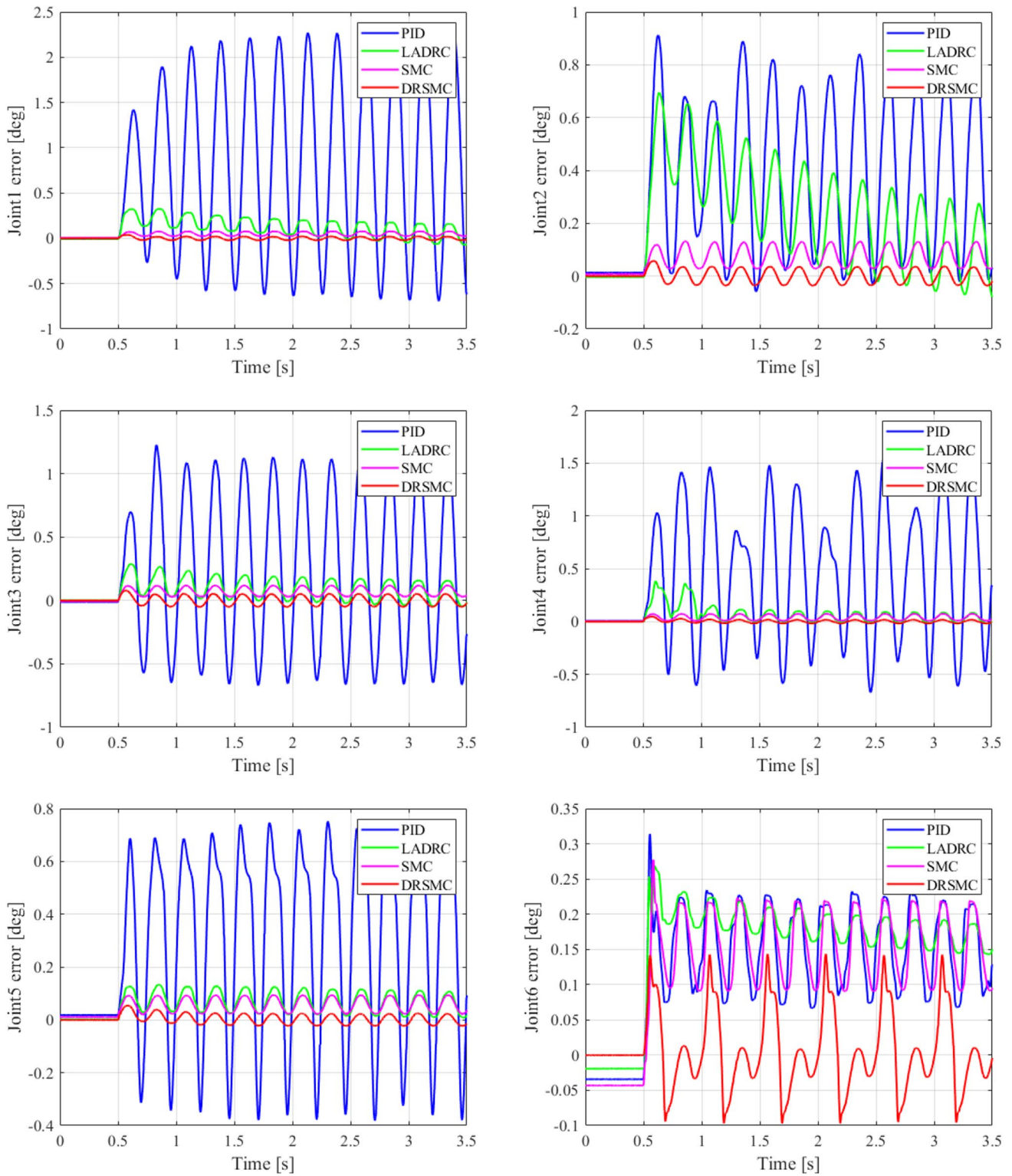


Fig. 12 Tracking trajectories of joint errors against unknown time-varying disturbances under different control methods

The SMC method exhibits better tracking performance than LADRC method and can reduce the system settling time by about 21.5% and decrease the steady-state error by

about 26.7%. The reason is that the designed sliding motion guarantees the control performance of the nominal system. The proposed DRSMC method has the best tracking

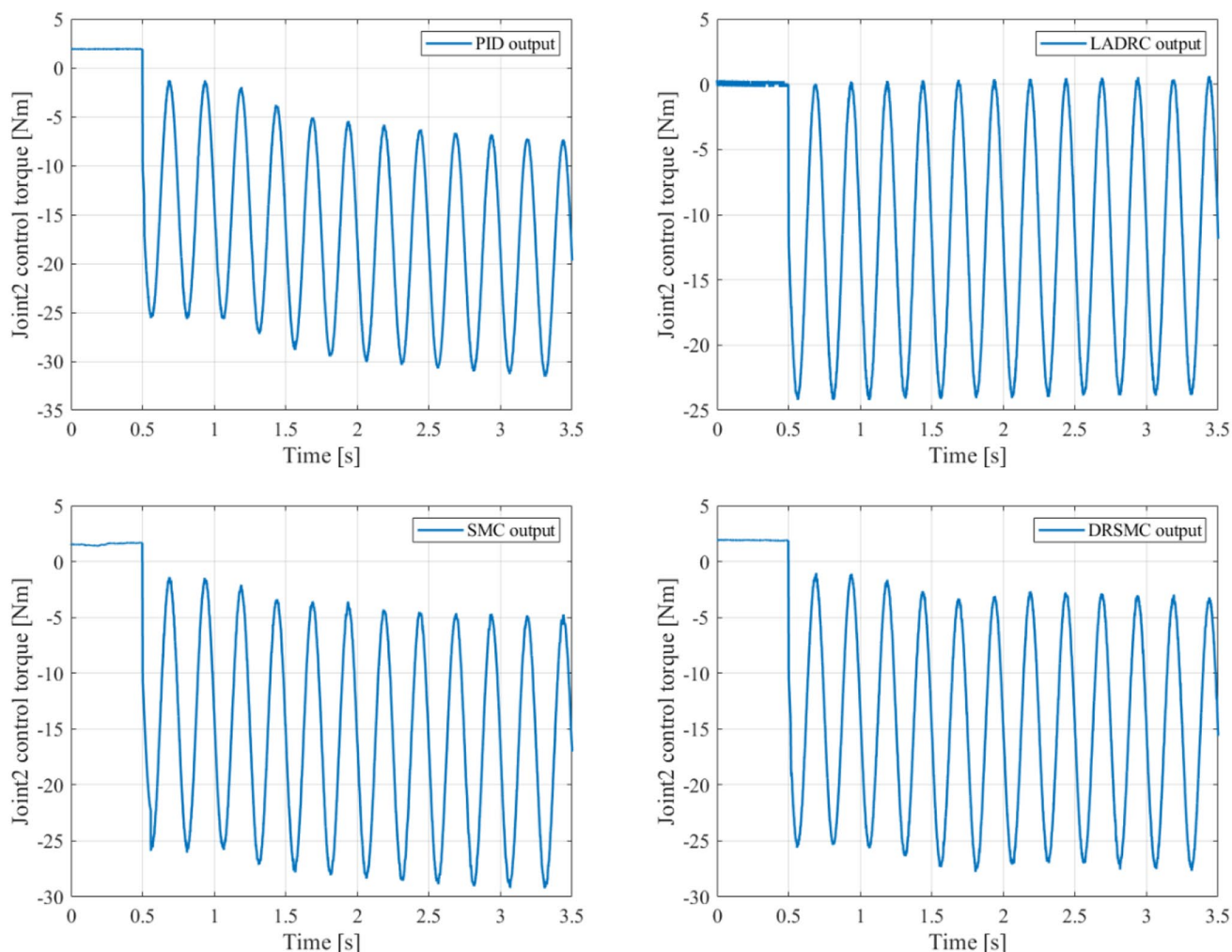


Fig. 13 Control torque (control input) of the comparative controllers

performance with regard to the response rapidity and steady-state error and can have a similar rise time as the PID controller, which approximates the shortest rise time our system can provide. Meanwhile, its settling time is also the shortest and approximates the rise time, which indicates that the corresponding transient period is smooth and steady. In addition, the DRSMC method mostly exhibits no overshoot or steady-state error owing to the sliding motion and the ESO compensation.

The results of the contrast experiments using the above four controllers against unknown time-varying disturbances are shown in Fig. 12. The representative characteristics are calculated in Table 2, where M_p , M_{p-p} , and M_a represent the average maximum value of the joint error, average peak-valley error in one cycle, and average joint error in 10 stable cycles of the 6 robot joints after disturbances are applied, respectively. The corresponding disturbances are given as Eq. (29), where the disturbance frequency is chosen as 4 Hz, and t_1 is chosen as 0.5 s. It should be noted that the

disturbances applied here are comparatively significant, with a peak value of 50% of the rated torque with the upper-bound disturbance frequency. In addition, all controller parameters here remain the same as in the first experiment. The control torque (or control input) of comparative controllers are shown in Fig. 13.

From Fig. 12 and Table 2, we can observe that the PID controller has the largest error caused by applied dynamic disturbances and the largest stable error mainly caused by constant disturbances. This shows that it is difficult for the PID controller to suppress strong time-varying disturbances regardless of whether they are dynamic or constant. Compared with the PID method, the LADRC method can effectively suppress the time-varying disturbances, especially constant disturbances. According to (14), we know that the ESO can estimate bounded disturbances and precisely estimate the constant disturbance. Therefore, the LADRC method can exhibit good performance despite unknown disturbances.

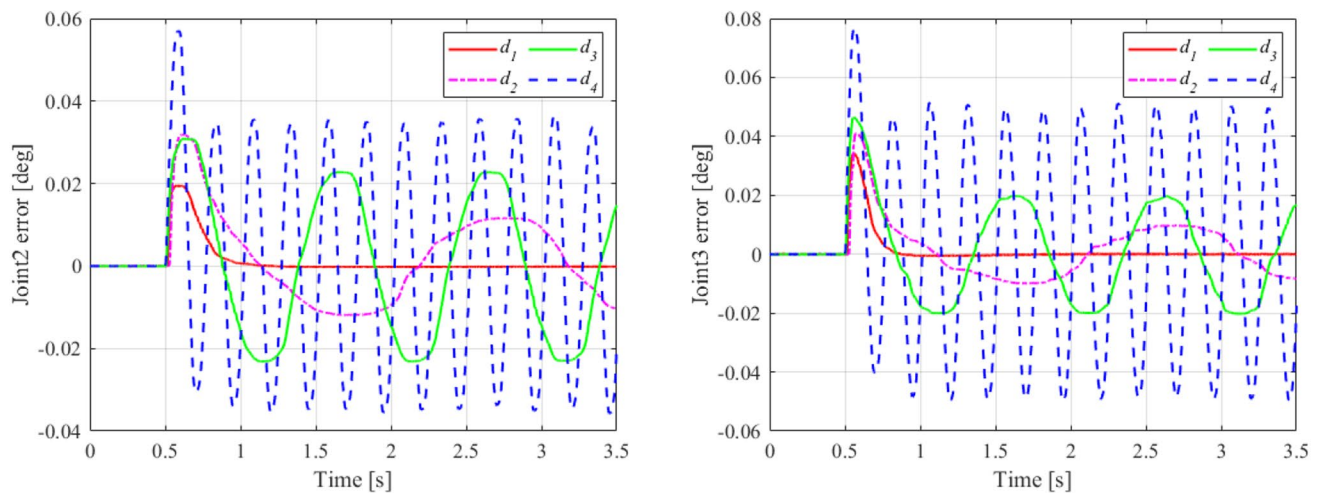


Fig. 14 Tracking trajectories of joint errors against different disturbance frequencies under DRSMC method

The SMC method can greatly decrease the maximum error caused by dynamic disturbances since the sliding motion along the sliding surface can rapidly reduce the joint error after disturbances are applied. However, its stable error is larger than that of the LADRC method. The proposed DRSMC method can combine the advantages of the LADRC method and SMC method, which can largely suppress both dynamic disturbances and constant disturbances. Figure 14 illustrates the suppressing disturbance ability of the DRSMC method under a given disturbance bandwidth. The disturbance frequencies in d_1 , d_2 , d_3 , and d_4 are chosen as 0 Hz, 0.5 Hz, 1 Hz, and 4 Hz.

As shown in Fig. 14, the DRSMC method has a great ability for disturbance rejection over the disturbance frequency range; furthermore, the DRSMC can greatly suppress low frequencies, especially static disturbances. The joint tracking error can be reduced to less than 0.01° over the -3 dB bandwidth even when the disturbance has a peak value of 50% of the rated torque. These experimental results prove that the proposed DRSMC method has a strong ability for unknown time-varying disturbance rejection and good application potential.

5 Conclusions

A practical and effective trajectory tracking control framework with a strong disturbance rejection ability for robots was presented in this paper. By combining the active disturbance rejection scheme, an ESO-based SMC law was developed to realize effective trajectory tracking while actively estimating and compensating unknown disturbances and system uncertainties simultaneously. A learning-based controller design methodology was introduced to realize the optimal design of the proposed controller, and an autonomous

learning method was developed for transferring the robot joint actuation ability.

To obtain a robust and transferring controller, a neural network was used to learn the joint actuation ability for the controller optimizing process. Simulation results and experimental results verified that the proposed controller design methodology is effective and robust, and the proposed control strategy can achieve satisfying tracking performance with a strong disturbance rejection ability. As an extension of this research project, future work will develop a scheme to adaptively adjust the controller online and use more feedback information to create a control policy for certain applications.

Funding Supported by the National Natural Science Foundation of China (51575306).

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

References

1. Yu H, Huang S, Chen G, Pan Y, Guo Z (2015) Human-robot interaction control of rehabilitation robots with series elastic actuators. *IEEE Trans Robot* 31(5):1089–1100. <https://doi.org/10.1109/tro.2015.2457314>
2. Lunardini F, Casellato C, d'Avella A, Sanger TD, Pedrocchi A (2016) Robustness and reliability of synergy-based myoelectric control of a multiple degree of freedom robotic arm. *IEEE Trans Neural Syst Rehabil Eng* 24(9):940–950. <https://doi.org/10.1109/TNSRE.2015.2483375>
3. Hwangbo J, Lee J, Dosovitskiy A, Bellicoso D, Tsounis V, Koltun V, Hutter M (2019) Learning agile and dynamic motor skills for legged robots. *Sci Robot* 4(26):13. <https://doi.org/10.1126/scirobotics.aau5872>

4. Petit F, Daasch A, Albu-Schaffer A (2015) Backstepping control of variable stiffness robots. *IEEE Trans Control Syst Technol* 23(6):2195–2202. <https://doi.org/10.1109/Tcst.2015.2404894>
5. Hoover RC, Roberts RG, Maciejewski AA, Naik PS, Ben-Gharbia KM (2015) Designing a failure-tolerant workspace for kinematically redundant robots. *IEEE Trans Autom Sci Eng* 12(4):1421–1432. <https://doi.org/10.1109/tase.2014.2337935>
6. Chen WJ, Tomizuka M (2014) Dual-stage iterative learning control for MIMO mismatched system with application to robots with joint elasticity. *IEEE Trans Control Syst Technol* 22(4):1350–1361. <https://doi.org/10.1109/Tcst.2013.2279652>
7. Chen CY, Huang PH (2012) Review of an autonomous humanoid robot and its mechanical control. *J Vib Control* 18(7):973–982. <https://doi.org/10.1177/1077546310395974>
8. Zhang D, Wei B (2017) A review on model reference adaptive control of robotic manipulators. *Annu Rev Control* 43:188–198. <https://doi.org/10.1016/j.arcontrol.2017.02.002>
9. Zhao Z, Yang J, Li S, Yu X, Wang Z (2018) Continuous output feedback tsm control for uncertain systems with a DC–AC inverter example. *IEEE Trans Circuits Syst II Express Briefs* 65(1):71–75. <https://doi.org/10.1109/tcsii.2017.2692256>
10. Cao Y, Chen XB (2014) Disturbance-observer-based sliding-mode control for a 3-DOF nanopositioning stage. *IEEE/ASME Trans Mechatron* 19(3):924–931. <https://doi.org/10.1109/tmech.2013.2262802>
11. Lima GS, Trimpe S, Bessa WM (2020) Sliding mode control with gaussian process regression for underwater robots. *J Intell Rob Syst*. <https://doi.org/10.1007/s10846-019-01128-5>
12. Bae J, Kong K, Tomizuka M (2011) Gait phase-based control for a rotary series elastic actuator assisting the knee joint. *J Med Devices* 5(3):6. <https://doi.org/10.1115/1.4004793>
13. Yang J, Li S, Yu X (2013) Sliding-mode control for systems with mismatched uncertainties via a disturbance observer. *IEEE Trans Ind Electron* 60(1):160–169. <https://doi.org/10.1109/tie.2012.2183841>
14. Wang H, Pan Y, Li S, Yu H (2019) Robust sliding mode control for robots driven by compliant actuators. *IEEE Trans Control Syst Technol* 27(3):1259–1266. <https://doi.org/10.1109/tcst.2018.2799587>
15. Van M, Mavrouniotis M, Ge SS (2019) An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. *IEEE Trans Syst Man Cybern Syst* 49(7):1448–1458. <https://doi.org/10.1109/tsmc.2017.2782246>
16. Gao Z-Q (2013) On the foundation of active disturbance rejection control. *Control Theory Appl* 30(12):1498–1510. <https://doi.org/10.7641/cta.2013.31087>
17. Han J-Q (2002) From PID technique to active disturbances rejection control technique. *Control Eng China (China)* 9(3):13–18
18. Chen Z, Wang Y, Sun M, Sun Q (2018) Global and asymptotical stability of active disturbance rejection control for second-order nonlinear systems. *Control Theory Appl* 35(11):1687–1696
19. Dan W, Ken C (2009) Design and analysis of precision active disturbance rejection control for noncircular turning process. *IEEE Trans Ind Electron* 56(7):2746–2753. <https://doi.org/10.1109/tie.2009.2019774>
20. Cheng Y, Chen Z, Sun M, Sun Q (2017) Multivariable inverted decoupling active disturbance rejection control and its application to a distillation column process. *Acta Autom Sin* 43(6):1080–1088
21. Huang Y, Xu KK, Han JQ, Lam J, Ieee I (2001) Flight control design using extended state observer and non-smooth feedback. In: Proceedings of the 40th IEEE conference on decision and control. Vols 1–5. IEEE conference on decision and control, pp. 223–228. Ieee, New York
22. Martinez-Fonseca N, Castaneda LA, Uranga A, Luviano-Juarez A, Chairez I (2016) Robust disturbance rejection control of a biped robotic system using high-order extended state observer. *ISA Trans* 62:276–286. <https://doi.org/10.1016/j.isatra.2016.02.003>
23. Guerrero-Castellanos JF, Rifai H, Arnez-Paniagua V, Linares-Flores J, Saynes-Torres L, Mohammed S (2018) Robust active disturbance rejection control via control lyapunov functions: application to actuated-ankle-foot-orthosis. *Control Eng Pract* 80:49–60. <https://doi.org/10.1016/j.conengprac.2018.08.008>
24. Long Y, Du ZJ, Cong L, Wang WD, Zhang ZM, Dong W (2017) Active disturbance rejection control based human gait tracking for lower extremity rehabilitation exoskeleton. *ISA Trans* 67:389–397. <https://doi.org/10.1016/j.isatra.2017.01.006>
25. Castaneda LA, Luviano-Juarez A, Chairez I (2015) Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control. *IEEE Trans Control Syst Technol* 23(4):1387–1398. <https://doi.org/10.1109/tcst.2014.2367313>
26. Talole SE, Kolhe JP, Phadke SB (2010) Extended-state-observer-based control of flexible-joint system with experimental validation. *IEEE Trans Ind Electron* 57(4):1411–1419
27. Xue W, Madonski R, Lakomy K, Gao Z, Huang Y (2017) Add-on module of active disturbance rejection for set-point tracking of motion control systems. *IEEE Trans Ind Appl* 53(4):4028–4040
28. Madonski R, Ramírez-Neria M, Gao Z, Yang J, Li S (2019) Attenuation of periodic disturbances via customized ADRC solution: a case of highly oscillatory 3DOF torsional plant. 2019 IEEE 8th data driven control and learning systems conference (DDCLS), Dali, China, 2019, pp. 1111–1116
29. Ren T, Dong Y, Wu D, Chen K (2018) Collision detection and identification for robot manipulators based on extended state observer. *Control Eng Pract* 79:144–153. <https://doi.org/10.1016/j.conengprac.2018.07.004>
30. Dong YF, Ren TY, Wu D, Chen K (2020) Compliance control for robot manipulation in contact with a varied environment based on a new joint torque controller. *J Intell Rob Syst*. <https://doi.org/10.1007/s10846-019-01109-8>
31. Fangli MOU, Dan WU, Yunfei DONG (2020) Active disturbance rejection control with multilayer perceptron compensating network for robot systems. *Control Theory Appl* 37(6):1397–1405
32. Khatib O, Burdick J (1986) Motion and force control of robot manipulators. In: Proceedings. 1986 IEEE international conference on robotics and automation, 7–10 April 1986, pp. 1381–1386
33. Bishop C (2006) *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, NY, USA
34. Han J (2009) From PID to active disturbance rejection control. *IEEE Trans Ind Electron* 56(3):900–906
35. Zhiqiang G (2003) Scaling and bandwidth-parameterization based controller tuning. Proceedings of the 2003 American Control Conference, 2003, Denver, CO, USA, 2003, pp. 4989–4996. Doi: <https://doi.org/10.1109/ACC.2003.1242516>
36. Zhou W, Shao S, Gao Z (2009) A stability study of the active disturbance rejection control problem by a singular perturbation approach. *Appl Math Sci* 3(10):491–508
37. Yang XX, Huang Y (2009) Capabilities of extended state observer for estimating uncertainties. *P Am Contr Conf*. pp 3700–3705. Doi: <https://doi.org/10.1109/Acc.2009.5160642>
38. Conn AR, Gould NIM, Toint PhL (1997) A globally convergent augmented lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Math Comput* 66(217):261–288
39. Clevert D-A, Unterthiner T, Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (ELUs). arXiv:1511.07289. [Online]. Available: <https://arxiv.org/abs/1511.07289>
40. Odena A, Olah C, Shlens J (2017) Conditional image synthesis with auxiliary classifier GANs. ArXiv.Org, 2017. Available: <https://search.proquest.com/docview/2076097206?accountid=14426>

41. Parra-Vega V, Arimoto S, Yun-Hui L, Hirzinger G, Akella P (2003) Dynamic sliding PID control for tracking of robot manipulators: theory and experiments. *IEEE Trans Robot Autom* 19(6):967–976. <https://doi.org/10.1109/tra.2003.819600>
42. Zurita-Bustamante EW, Linares-Flores J, Guzmán-Ramírez E, Sira-Ramírez H (2011) A comparison between the GPI and PID controllers for the stabilization of a DC-DC ‘buck’ converter: a field programmable gate array implementation. *IEEE Trans Ind Electron* 58(11):5251–5262
43. Sun J, Yang J, Zheng W, Li S (2016) GPIO-based robust control of nonlinear uncertain systems under time-varying disturbance with application to DC–DC converter. *IEEE Trans Circ Syst II Exp Briefs* 63(11):1074–1078
44. Calanca A, Capisani LM, Ferrara A, Magnani L (2011) MIMO closed loop identification of an industrial robot. *IEEE Trans Control Syst Technol* 19(5):1214–1224. <https://doi.org/10.1109/tcst.2010.2077294>