Terminal Traction Control of Teleoperation Manipulator With Random Jitter Disturbance Based on Active Disturbance Rejection Sliding Mode Control

PENG JI\textsuperscript{1}, FENGYING MA\textsuperscript{1}, AND FENG MIN\textsuperscript{2}

\textsuperscript{1}School of Electrical Engineering and Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
\textsuperscript{2}Department of Intelligent Manufacturing, Shandong Labor Vocational and Technical College, Jinan 250353, China

Corresponding author: Fengying Ma (mfy@qlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61903207; in part by the Key Research and Development Plan of Shandong Province under Grant 2019JZZY010731; and in part by the Open Fund Project of Jiangsu Provincial Key Laboratory of Remote Measurement and Control under Grant 2242019k30036.

\textbf{INDEX TERMS} Teleoperation manipulator, active disturbance rejection, sliding mode control, extend state observer.

\textbf{I. INTRODUCTION}

Some armed reconnaissance robots are equipped with a multi-degree of freedom manipulator, and a reconnaissance camera is installed at the end of the manipulator. Thus, the operator can aim and shoot the target efficiently by controlling the vehicle-mounted manipulator. On the whole, there are two control modes for the vehicle-mounted teleoperation manipulator, i.e., single-joint control mode and terminal trajectory control mode. The former usually requires the operator to control the angle of each joint through the knobs or joysticks of the teleoperation controller, to achieve the control of the position and posture of the manipulator’s end. During the control, the operator is required to adjust the angle of each joint several times to make the manipulator’s end in the ideal position [1]. The latter requires the operator to generate the motion trajectory in Cartesian space by using the human-computer interaction (HCI) device, then solve the corresponding joint trajectory of the motion trajectory in the joint space by the robot inverse kinematics algorithm,
and lastly enable each joint of the manipulator to track the corresponding joint trajectory based on the joint trajectory tracking algorithm. Accordingly, the end trajectory control of the manipulator control (i.e. terminal traction control) can be achieved. During terminal traction of manipulators through the HCI device, non-uniform traction will likely occur for human factors (unintentionally or intentionally). Non-uniform traction will cause the trajectory of the manipulator’s end to change suddenly, thereby leading to the dramatic change of the joint angular acceleration and random jitter. The high acceleration and random jitter can reduce the precision of trajectory tracking and cause significant residual vibration [2].

Specific to several fast teleoperation control tasks (e.g., fast aiming and shooting task), the operator should use HCI devices to control the teleoperation manipulator to efficiently aim at the target and track the target with fast random movement. This requires that the terminal traction control can make the manipulator’s end track a time-varying trajectory in real-time, and the tracking process should ensure good stability, sensitive response speed, and high accuracy. However, to enable the manipulator’s end to aim at the random target efficiently, the operator can deliberately adopt the non-uniform traction operation to adapt to the randomness of the target, which will land the manipulator into random jitter in the tracking process, and then lose the target and lead to mission failure. Moreover, the manipulator refers to a highly coupled nonlinear system with multiple inputs and outputs, and there are uncertainties (e.g., time-varying disturbance, internal friction, parameter perturbation, and modeling error), thereby hindering the high-speed and high-precision trajectory tracking control of the manipulator.

Through the years, considerable research effort has been made in the teleoperation task and many control algorithms have been proposed for the trajectory tracking of teleoperation manipulator. For instance, computed torque control, adaptive control, sliding mode control, neural network control, and active disturbance rejection control are used.

Computed torque control (CTC), also known as inverse dynamics control, generally linearizes the robot system globally through feedback linearization in the inner loop and decouple it into a double integral system, and then uses the PID method to control the robot in the external loop [3]. In [4], the author applied CTC to a visual tracking robot (a 2-DOF robotic arm). To overcome actuator saturation, a genetic algorithm is adopted to optimize the PID parameters in CTC. To achieve the trajectory tracking of a 2-DOF manipulator, a composite CTC is proposed by combining CTC with the implicit Lyapunov function in [5], and the uncertainties attributed to load variation and base vibration are solved successfully. As is well known, linear feedback, in general, cannot achieve a fast response without causing the response to overshoot [6]. Besides, CTC needs an accurate dynamic model of the robot, and the control accuracy depends on the accuracy of the model, which is inevitably affected by the uncertainties (e.g., modeling error, load change, and external disturbance). When the model is accurate and there is no external disturbance, CTC can make the tracking error converge to zero. However, in practical application, it is impossible to obtain a completely accurate model due to the existence of uncertainty. Therefore, when there are uncertainties (e.g., internal friction and time-varying disturbance), it is difficult to obtain the desired tracking effect. To obtain an ideal control effect, the uncertainty in the manipulator system should be estimated. To remedy modeling uncertainties and external disturbances, in [7], the author used a fuzzy compensator in CTC and achieved the trajectory tracking of three joint manipulators. In [8], the extended state observer is adopted to estimate the constructed and unstructured uncertainties and is successfully applied to the trajectory tracking of a 2-joint manipulator.

Adaptive control is capable of regulating the parameters of the controller in real-time to adapt to the uncertainty of parameter perturbation, modeless dynamics, or external disturbance of the control object. In [9], a new robust adaptive model reference impedance controller is developed for nonlinear robot manipulator with parameter uncertainties. In [10], a novel bio-inspired dynamics-based adaptive tracking control is proposed for improving the energy efficiency of adaptive active control of vehicle suspension systems, considering the issues simultaneously (e.g., robustness, stability, actuator saturation, and nonlinear benefits). In [11], an adaptive fault-tolerant compensation controller is proposed for the uncertain nonlinear pure-feedback systems possessing dead-zone actuators and stochastic failures. A common adaptive controller applied to trajectory tracking of the manipulator is model reference adaptive control [12]. Besides, neural networks have been extensively employed as adaptive adjusters of controller parameters [14], [15]. For instance, in [16], [17], [46], adaptive control is proposed for manipulators with unknown dynamics and motion constraints of position and velocity. The controller employs a radial basis function (RBF) neural network to adaptively remedy the nonlinearities of the plant. Since adaptive control fits the change of object or environment by altering the controller parameters, such a control strategy of “responding to change through change” is subjected to the risk of causing the system to be destabilized.

Specific to sliding mode control, the state of the system slides along the sliding surface by switching the control variables. As a result, the system turns invariant when undergoing parameter perturbation and external interference. Besides, the sliding mode control algorithm is simple, fast-responding, and robust to external disturbance and parameter perturbation. This method exhibits the defect that when the state trajectory reaches the sliding surface, the system cannot slide rigorously along the sliding surface towards the equilibrium point, whereas it traverses back and forth on both sides of the sliding surface, which creates chattering [18]. The conventional sliding mode control cannot ensure the system state constantly exhibit the expected dynamic performance at this stage before entering the sliding mode. To remedy
such a defect, the integral sliding mode control (ISM) is proposed. By setting the initial state of the integrator reasonably, the initial state of the system is on the sliding mode surface at the beginning, thus eliminating the arrival phase and improving the robustness of the control system [19]. However, ISM exhibits insensitivity to the matched and unmatched uncertainties throughout the system response. In [23], an unknown system dynamics estimator (USDE) based sliding mode control was proposed for servo mechanisms with unknown dynamics and modeling uncertainties. To reduce chattering, a novel reaching law is formulated based on hyperbolic functions to ensure that the sliding mode variable infinitely approaches the equilibrium point, instead of crossing it. In [24], a novel dead zone sliding mode reaching law capable of reducing chattering was developed for uncertain discrete-time systems. In the conventional sliding mode control, when the system state reaches the sliding mode, the steady-state error does not converge to zero in finite time. To make the system state converge to the equilibrium point in finite time, terminal sliding mode control is proposed [20], [21]. To expedite the convergence, a fast terminal sliding mode is proposed. [22] investigated a novel digital fast terminal sliding mode control (FTSMC) approach for DC-DC buck converters with mismatched disturbances. To address the singular problem of terminal sliding mode control, a nonsingular terminal sliding mode was proposed [25]–[29]. [29] proposed a non-singular fast terminal sliding mode control for a class of second-order uncertain nonlinear manipulator systems with only available position measurement. The mentioned control algorithms require the upper bound of the uncertainties in the robot system. Overall, the upper bound of the uncertainty is hard to precisely define.

Neural network (NN) exhibits a highly parallel structure, strong learning ability, nonlinear function approximation ability, and fault tolerance ability. Numerous researchers attempt to use neural networks to remedy nonlinear uncertainties [30]. In [31], an adaptive neural network control based on the full-state feedback method has been applied for manipulators with output constraints and uncertainties. To handle the unmodeled dynamics, the NN is adopted to approximate the uncertain dynamics. Fuzzy logic control also can approximate nonlinear functions. Moreover, it can simulate the human experience and decision behavior, and it has extensive applications in remediying unknown non-linearities and disturbance. In [32], a fuzzy logic control strategy was proposed to solve trajectory tracking control problems of an uncertain manipulator. Fuzzy logic is adopted to remedy nonlinear uncertainties in manipulator dynamics. A considerable number of researchers combined neural networks or fuzzy systems with other control methods and exploited neural networks and fuzzy systems to approximate and compensate time-varying disturbances and internal friction. In [33], an RBF-based fuzzy sliding-mode control method was proposed to make the manipulator track the given trajectory at an ideal dynamic quality. [34] presented a novel adaptive nonsingular terminal sliding mode controller for the trajectory tracking of robotic manipulators using radial basis function neural networks (RBFNNs). The neural network, however, has numerous super parameters required to be adjusted (e.g., learning rate, number of hidden layer neurons, type of activation function, and loss function), and these super parameters may result in an over-fitting problem or under-fitting problem [35]. The design of a fuzzy controller is complicated, especially when the system complexity is very high, the appropriate membership function and fuzzy rules are difficult to determine [36], [37].

On the whole, active disturbance rejection control (ADRC) consists of a tracking differentiator(TD), an extended state observer (ESO), as well as error feedback control laws. Its basic idea is to treat the uncertainties and disturbances in the system as lumped disturbance, estimate and eliminate the lumped disturbance online through the extended observer in real-time, and decouple the nonlinear uncertain system into a double integral system [38]. ADRC controllers are structurally simple, efficient, and excellent at anti-interference, which have been applied for robot control [39]–[42]. However, ADRC is subjected to two major problems, i.e., the disturbance estimation errors and parameter variations, which cannot be included in the lumped disturbance and then estimated with the extended observer. The estimation errors of the lumped disturbance are commonly neglected, while parameter variations of the control input are not considered [43]. As revealed from the former analysis, the sliding mode control exhibits insensitivity to external disturbance and parameter perturbation, which can remedy the defect of ADRC.

The present study proposes an advanced terminal traction control system by exploiting the attractive features of ADRC and SMC to make the operator use the HCI devices to efficiently control the manipulator and enable its end to efficiently aim and track the target with fast random movement. First, the multi-input multi-output nonlinear and strong coupling manipulator system is decoupled into a single input single output joint group. Subsequently, the effect of non-uniform traction on each joint of the robot is considered interference, and the anti-interference ability of the Active Disturbance Rejection Sliding Mode Controller (ADRSMC) is proposed to eliminate the effect interference. Accordingly, each joint can track the desired joint trajectory efficiently, accurately, and steadily to achieve the terminal traction control of the manipulator in a fast, stable, and high-precision manner by using an HCI device. The proposed ADRSMC consists of three main steps. First, a differential tracker (TD) is adopted to filter the trajectory. Second, a model-assisted nonlinear extended state observer (MNESO) is employed to estimate the lumped disturbance online. Lastly, to track the desired joint trajectories, the SMC law is formulated to eliminate the lumped disturbance. The proposed ADRSMC is then tested through a series of contrastive experiments. It has shown that the proposed ADRSMC has a better performance than the classic CTC in terms of the trajectory tracking error.
In brief, the major contributions of the present study are as follows. 1) A terminal traction control system is proposed to reduce the joint jitter attributed to non-uniform traction during terminal traction of the teleoperation manipulator by HCI device. 2) An active disturbance rejection sliding mode controller is developed, capable of treating the effect of non-uniform traction on joints as lumped interference and remedying it by online estimation, as well as reducing the chattering of the control output signal by incorporating a low-pass filter. 3) The global stability of the system is verified when combining the sliding mode controller and ADRC based on the Lyapunov stability theory.

The rest of the paper is organized as follows. The modeling and description of the multi-DOF manipulator system are given in Section II. The robust ADRSMC strategy and the proof of stability are given in Section III, which includes the system decoupling, TD, MNESSO and SMC. Section IV presents the verification experiments. Finally, the conclusion is given in Section V.

II. PROBLEM FORMULATION AND PRELIMINARIES

According to the Lagrange equation, for a rigid N-degree-of-freedom manipulator system, which suffers external disturbance and model uncertainty, the manipulator system can be established by the following dynamic equation:

\[ M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(q) + d(t) = \tau \]  

(1)

where, \( q \in R^{n \times 1} \), \( \dot{q} \in R^{n \times 1} \) and \( \ddot{q} \in R^{n \times 1} \) is the vector of joint position, joint speed, and joint acceleration; \( M(q) \) is the positive definite inertia matrix; \( C(q, \dot{q}) \in R^{n \times n} \) is centrifugal force and Coriolis force; \( G(q) \in R^{n \times 1} \) is gravity; \( F(q) \in R^{n \times 1} \) is friction; \( d \) is unknown external interference; \( \tau \) is the control torque.

In practice, it is difficult to obtain the complete information of the manipulator model. In general, the nominal model of the manipulator can be obtained. Suppose the nominal model of the manipulator is \( M_0(q), C_0(q, \dot{q}), G_0(q) \). Let \( \Delta M = M_0 - M \), \( \Delta C = C_0 - C \), \( \Delta G = G_0 - G \), and the following equation can be obtained by combining the kinetic equation:

\[ [M_0(q) - \Delta M] \ddot{q} + [C_0(q, \dot{q}) - \Delta C] \dot{q} + G_0(q) - \Delta G + F(q) + d(t) = \tau \]  

(2)

Therefore,

\[ M_0(q) \ddot{q} + C_0(q, \dot{q}) \dot{q} + G_0(q) = \tau + \tau_d(q, \dot{q}, \ddot{q}, t) \]  

(3)

where \( \tau_d(q, \dot{q}, \ddot{q}, t) \in R^{n \times 1} \) represents the unmodeled items, friction, and unknown disturbance of the manipulator, and its expression is \( \tau_d(q, \dot{q}, \ddot{q}, t) = \Delta M \ddot{q} + \Delta C \dot{q} + \Delta G - F(q) - d(t) \). It should be noted that the value of \( \tau_d(q, \dot{q}, \ddot{q}, t) \) is generally unknown and needs to be estimated or observed to compensate in the controller. In different references, \( \tau_d(q, \dot{q}, \ddot{q}, t) \) may be located on the left or right side of the dynamic equation. It doesn’t matter. In short, it represents an unknown term. As long as it can be estimated or observed, it can be compensated in the controller.

When \( x_1 = q, x_2 = \dot{q}, \tau = u \) is defined, the above equation can be written as a state-space equation:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= M_0(x_1)^{-1} [u + \tau_d - C_0(x_1, x_2) x_2 - G_0(x_1)] \\
y &= x_1 
\end{align*}
\]  

(4)

The above equation can be rewritten as the following equation:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= F(x_1, x_2) + B(x_1) u + D(x_1) \\
y &= x_1 
\end{align*}
\]  

(5)

where \( y \) is the joint displacement vector of the manipulator; \( B(x_1) = M_0(x_1)^{-1} F(x_1, x_2) = -M_0(x_1)^{-1}[C_0(x_1, x_2) x_2 + G_0(x_1)] \), \( D = M_0(x_1)^{-1} \tau_d \).

Considering the saturation constraint of the joints’ actuators, we design the saturation function to limit the control vector \( u \):

\[
\begin{align*}
u &= \left\{ \begin{array}{ll} 
\tau_{\text{upper}}, & \tau > \tau_{\text{upper}} \\
\tau, & -\tau_{\text{lower}} \leq \tau \leq \tau_{\text{upper}} \\
-\tau_{\text{lower}}, & \tau < -\tau_{\text{lower}} 
\end{array} \right.
\]  

(6)

where \( \tau_{\text{upper}} \) and \( -\tau_{\text{lower}} \) are the known values of the saturation function.

The objective here is to make the operator use the HCI (e.g., a mouse, joystick, or force feedback handle) devices to efficiently control the manipulator and enable its end to efficiently aim and track the target with fast random movement. In fact, the above objective requires the operator to achieve the fast terminal traction control of the manipulator through the HCI devices, which requires the following steps. First, the traction trajectory generated by the HCI device is mapped into the motion trajectory under the base coordinate system of the manipulator; subsequently, the corresponding joint solution of the motion trajectory in the joint space is solved by using the inverse kinematics algorithm; lastly, the trajectory tracking control algorithm is adopted to let each joint of the manipulator track the solved joint solution (e.g., desired joint angle, velocity, and angular acceleration). The corresponding symbols are \( q_d, \dot{q}_d, \) and \( \ddot{q}_d \).

Thus, the goal here is to design a controller, enabling the joint angle \( x_1 \) and joint velocity \( x_2 \) of the manipulator system to track the \( q_d \) and \( \dot{q}_d \) generated by the HCI device efficiently, stably, and accurately. The tracking position errors and speed errors of the system are expressed as:

\[
\begin{align*}
e &= x_1 - q_d \\
\dot{e} &= x_2 - \dot{q}_d 
\end{align*}
\]  

(7)

This study aims to make the error of the expressed formula stable to zero in finite time. The tracking position errors and speed errors of the system efficiently approach zero in finite time, suggesting that the developed controller enables each joint of the manipulator to track the desired trajectory efficiently, accurately, and stably. Besides, it is ensured that...
the operator can efficiently regulate the manipulator’s end to aim at the randomly moving target under his intention, as an attempt to achieve the goal of shooting where he points.

III. TERMINAL TRACTION CONTROL SYSTEM DESIGN

A. OVERVIEW AND ROBOT SYSTEM DECOUPLING

In the teleoperation control system, the operator usually controls the manipulator’s end through an HCI device (e.g., a mouse, joystick, gesture, or force feedback device). During terminal traction of manipulator through HCI devices, non-uniform control will likely occur due to human factors (e.g., hand tremor or random traction). This leads to abrupt changes in the motion trajectory generated by an HCI device. After solving the inverse kinematics of the robot, the desired joint trajectory corresponding to the motion trajectory generated by the HCI device will also appear abrupt acceleration changes. Because the manipulator system is a nonlinear strong coupling system, when each joint of the manipulator system tracks the desired joint trajectory with abrupt acceleration, the manipulator’s end will appear flutter phenomenon during tracking the desired trajectory and even lead to instability. To reduce the flutter or instability of the manipulator’s end attributed to non-uniform control, an active disturbance rejection sliding mode control system is proposed in the present study. The overall schematic diagram of the system is shown in Fig. 1 below.

The control flow of the system is as follows: 1) decoupling the MIMO nonlinear and strong coupling manipulator system into a single input single output joint group; 2) mapping the traction trajectory generated by the HCI device into the motion trajectory in the base coordinate system of the manipulator; 3) filtering the motion trajectory through a low-pass filter to filter out the high-frequency noise attributed to non-uniform traction; 4) solving the corresponding joint solution of motion trajectory in joint space; 5) regarding the effect of non-uniform traction on each joint as disturbance, and introducing a “virtual control variable” for each joint, in which the “virtual control variable” is achieved by an active disturbance rejection sliding mode controller (ADRSMC); 6) restoring the “virtual control variable” to the actual control torque of the whole manipulator, and sending the actual control torque to the teleoperation manipulator after low-pass filtering. The core of the system is to treat the effect of non-uniform traction on the robot joint as interference and eliminate the effect interference through the anti-interference ability of the active disturbance rejection sliding mode controller, so that each joint can track the joint trajectory corresponding to the terminal traction trajectory efficiently, accurately and stably, to achieve the fast, stable and high-precision terminal traction control of the manipulator by the HCI device.

The manipulator is a multi-input multi-output nonlinear coupling system. To realize joint decoupling, we define $x = [x_1, x_2, \ldots, x_n]^T$, $F = [F_1, F_2, \ldots, F_n]^T$, $u = [u_1, u_2, \ldots, u_n]^T$, $D = [D_1, D_2, \ldots, D_n]^T$. For the MIMO nonlinear affine system represented by the system (5), a “virtual control variable” $U = B\{x_i\}u$ is introduced in this paper, where $U = [U_1, U_2, \ldots, U_n]^T$. Then the robot system (5) becomes

$$
\begin{align*}
\dot{x} &= F(x) + U + D(x) \\
y &= x
\end{align*}
$$

The input-output relationship of the $i$-th joint in the robot system (5) becomes:

$$
\begin{align*}
\dot{x}_{i1} &= x_{i2} \\
\dot{x}_{i2} &= F_i(x_{i1}, x_{i2}) + U_i + L_i(x_{i1}) \\
y_i &= x_{i1}
\end{align*}
$$

That is, the input of the $i$-th joint is $U_i$, and the output is $y_i$. In this way, the relationship between input control $U_i$ and output $y_i$ is a SISO relationship. That is to say, the relationship between input control $U_i$ and output $y_i$ is completely decoupled. $D_i(x_{i1})$ can represent the interference to the $i$-th joint due to non-uniform traction. The actual control variable $u = [u_1, u_2, \ldots, u_n]^T$ of the manipulator can be represented by the virtual control variable $U = [U_1, U_2, \ldots, U_n]^T$. Since the active disturbance rejection controller is used in this paper, to reduce the chattering of the control output, a low pass filter is applied to the actual control torque in this paper, as shown in the following formula:

$$
u = LBF\left(B^{-1}U\right)
$$

B. SINGLE JOINT CONTROLLER DESIGN

To eliminate the chattering or instability attributed to un-uniform traction, we first decouple the $n$-joint
MIMO manipulator system and then apply the framework of the active disturbance rejection controller (ADRC) to each joint after decoupling. To improve the robustness of ADRC, an ADRSMC with better robustness is proposed for the class of second-order uncertain manipulator systems (8) to achieve the end traction control with random jitter disturbance. The schematic diagram of ADRSMC of a single joint is shown in Fig.2. First, the tracking differentiator (TD) is employed to filter the desired joint trajectory and calculate its velocity. Then, the model-assisted extended state observer is adopted to estimate and eliminate the disturbance of each joint online. Lastly, the sliding mode control algorithm is adopted to make each joint track the filtered joint trajectory with low chattering. To reduce the jitter of control torques in the sliding mode controller, the desired acceleration involved in sliding mode control is processed with a mean filter in advance.

C. TRACKING DIFFERENTIATOR

Tracking differentiator (TD) is critical to the active disturbance rejection controller (ADRC). It largely aims to extract continuous signals and the differential signal from the signal with random noise. The non-uniform traction during teleoperation will cause the joint trajectory to change suddenly. Such a process is equated with introducing a step response signal into the reference signal, thereby causing the instantaneous tracking error of the system to increase suddenly. Under the inertia and time delay of the manipulator system, to reduce tracking errors, the control gain should increase commonly, which will produce large overshoot and easily cause chattering or instability. If the control gain does not increase, the real-time tracking performance cannot be ensured.

A suitable transition process should be set for the reference signal to track the reference signal with a mutation in time and without error. To track the discontinuous input signal, the tracking differentiator (TD) is adopted to preprocess the joint trajectory.

To make the manipulator system track the reference signal with random noise instantly and accurately, a suitable transition process should be set for the reference signal. In the present study, a tracking differentiator (TD) is employed to preprocess the joint trajectory.

To facilitate numerical calculation, the discrete form of the tracking differentiator (TD) is used in this paper, as shown in the following formula:

$$\begin{align*}
    x_{i1d}(k+1) &= x_{i1d}(k) + h \cdot x_{i2d}(k) \\
    x_{i2d}(k+1) &= x_{i2d}(k) + h \cdot \tanh(x_{i1d}(k))
\end{align*}$$

(11)

where $x_{i1d}$ and $x_{i2d}$ is the expected state of the $i$-th joint, $h$ is the sampling period, and $\theta_{id}$ is the input signal to be tracked, $r_{i0}$ is the parameter of tracking speed, $h_{i0}$ is the step size different from the sampling period, $\tanh(x_{i1d}(k) - \theta_{id}, x_{i2d}(k), r_{i0}, h_{i0})$ is expressed as follows:

$$\begin{align*}
    d &= r_{i0} h_{i0}^2 \\
    a_0 &= h_{i0} x_{i2d} \\
    y &= x_{i1d} + a_0 \\
    a_1 &= \sqrt{d} (d+8 |y|) \\
    a_2 &= a_0 + \frac{\text{sign}(y)}{2} (a_1 - d) \\
    s_1 &= \frac{[\text{sign}(y+d) - \text{sign}(y-d)]}{2} \\
    s_2 &= \frac{[\text{sign}(a+d) - \text{sign}(a-d)]}{2} \\
    f_{\text{tanh}} &= -r_{i0} \left[ \frac{a}{d} \right] s_2 - r_{i0} \text{sign}(a)
\end{align*}$$

(12)

By (11) and (12), $x_{i1d}$ can track the input signal $\theta_{id}$ quickly without overshoot and output the approximate differential
signal of \( \theta_{id} \). If \( \theta_{id} \) is a signal with noise, the differential tracker can realize filtering and output the filtered signal \( x_{id} \) at the same time.

**D. MODEL ASSISTED NONLINEAR EXTENDED STATE OBSERVER**

The abrupt change of joint trajectory attributed to non-uniform traction can lead to a sudden increase of instantaneous tracking error of the system. The instantaneous effect attributed to non-uniform traction is considered here as a type of interference superimposed on the manipulator system. If there is a way to estimate such interference, it can be eliminated before adversely affecting the tracking system. Besides, the disturbance attributed to the non-uniform traction and the uncertainty in the system is recognized as the lumped disturbance. Subsequently, an extended state observer (ESO) is constructed to estimate and eliminate the lumped disturbance online, as an attempt to convert the nonlinear uncertainty system to a double integral system.

In the second-order linear uncertain system described by (9), \( F_i(x_{i1}, x_{i2}) \) is the known model information. To reduce the computational burden of ESO and reduce the delay of disturbance estimation, a model-assisted extended state observer is designed. Take \( D_i(x_{i1}) \) in (9) as state system \( x_{i3} \), and assume that \( D_i(x_{i1}) \) is differentiable and its derivative is bounded, then (9) can be written as follows:

\[
\begin{align*}
\dot{x}_{i1} &= x_{i2} \\
\dot{x}_{i2} &= F_i(x_{i1}, x_{i2}) + U_i + x_{i3} \\
\dot{x}_{i3} &= D_i(x_{i1}) \\
y_i &= x_{i1}
\end{align*}
\]  

(13)

If \( z_{i1} = y_i, z_{i2} = \dot{y}_i, z_{i3} = D_i(x_{i1}) \), then \( z_i = [z_{i1}, z_{i2}, z_{i3}]^T = [y_i, \dot{y}_i, D_i(x_{i1})]^T \) is the estimated state including the disturbance. (13) can be further rewritten as

\[
\begin{align*}
\dot{z}_i &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} z_i + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [U_i + F_i(x_{i1}, x_{i2})] \\
&+ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{D}_i(x_{i1}) \\
y_i &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} z_i
\end{align*}
\]  

(14)

To obtain the estimated value \( \hat{z}_i \) of the expanded state \( z_i \), the model-assisted state expanded state observer designed in this paper is shown as follows:

\[
\begin{align*}
\dot{z}_i &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \hat{z}_i + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [U_i + F_i(x_{i1}, x_{i2})] \\
&+ L (y_i - \hat{y}_i) \\
\hat{y}_i &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \hat{z}_i
\end{align*}
\]  

(15)

where \( L = [\beta_{i1}, \beta_{i2}, \beta_{i3}]^T \) is the observer error feedback gain matrix, \( \beta_i > 0 (i = 1, 2, 3) \). Since \( \dot{D}_i(x_{i1}) \) is unknown and can be estimated by the correction term, \( \dot{D}_i(x_{i1}) \) is omitted from the above equation. Rewritten (15) can be obtained as follows:

\[
\begin{align*}
\dot{z}_i &= \begin{bmatrix} -\beta_{i1} & 1 & 0 \\ -\beta_{i2} & 0 & 1 \\ -\beta_{i3} & 0 & 0 \end{bmatrix} z_i + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} [U_i + F_i(x_{i1}, x_{i2})] \\
&+ \begin{bmatrix} \beta_{i1} \\ \beta_{i2} \\ \beta_{i3} \end{bmatrix} y_i \\
\hat{y}_i &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \hat{z}_i
\end{align*}
\]  

(16)

Make \( e_i = \tilde{z}_{i1} - y_i \), then (16) can be written in the form of a linear extended observer as follows:

\[
\begin{align*}
e_i &= \tilde{z}_{i1} - y_i \\
\dot{\tilde{z}}_{i1} &= \tilde{z}_{i2} - \beta_{i1} e_i \\
\dot{\tilde{z}}_{i2} &= \tilde{z}_{i3} - \beta_{i2} e_i + F_i(x_{i1}, x_{i2}) + U_i \\
\dot{\tilde{z}}_{i3} &= -\beta_{i3} e_i
\end{align*}
\]  

(17)

To speed up the convergence of the observer and prevent overshoot, the nonlinear function \( fal(e_i, \alpha_i, \delta_i) \) with “large error and small gain, small error and large gain” is used instead of \( e_i \) in the linear extended observer. Then the linear extended observer can be rewritten as the following nonlinear extended state observer (NESO):

\[
\begin{align*}
e_i &= \tilde{z}_{i1} - y_i \\
\dot{\tilde{z}}_{i1} &= \tilde{z}_{i2} - \beta_{i1} e_i \\
\dot{\tilde{z}}_{i2} &= \tilde{z}_{i3} - \beta_{i2} fal(e_i, \alpha_i, \delta_i) + F_i(x_{i1}, x_{i2}) + U_i \\
\dot{\tilde{z}}_{i3} &= -\beta_{i3} fal(e_i, \alpha_i, \delta_i)
\end{align*}
\]  

(18)

where,

\[
fal(e_i, \alpha_i, \delta_i) = \begin{cases} \frac{e_i}{\delta_i - \alpha_i}, & |e_i| \leq \delta_i \\ |e_i| \alpha_i \text{ sign}(e_i), & |e_i| > \delta_i \end{cases}
\]  

(19)

In (19), \( e_i \) is the input error variable; \( \alpha_i \) is a nonlinear factor and \( 0 < \alpha_i < 1; \delta_i \) is the filter factor.

Due to \( \tilde{D}_i(x_{i1}) \) is bounded, that is, there exists such a positive integer \( \gamma_{max} \) makes \( |\tilde{D}_i(x_{i1})| \leq \gamma_{max} \) is established. Then the following corollary holds: in a finite time \( t_i \), there exists suitable \( t_i > 0 \) and \( \beta_{i1} > 0, \beta_{i2} > 0, \beta_{i3} > 0 \), such that \( \forall t \geq t_i, |z_{in} - z_{in}| \leq \sigma_{in} (n = 1, 2, 3) \), where \( \sigma_{in} \) is a small positive number. In other words, the nonlinear extended observer represented by (18) can realize \( \tilde{z}_{i1} \to z_{i1}, \tilde{z}_{i2} \to z_{i2}, \tilde{z}_{i3} \to z_{i3} = D_i(x_{i1}) \) in finite time.

**E. SLIDING MODE CONTROL**

Since the sliding mode can be designed, and it is independent of plant parameters and disturbances, sliding mode control exhibits a fast response and insensitivity to parameter perturbation and disturbance. To enhance the robustness of ADRC, the nonlinear feedback control law of ADRC is formulated here as a sliding mode control law.
The trajectory tracking error of the i-th joint is defined as $e_{i1} = x_{i1} - x_{i1d}$, the sliding surface is designed as follows:

$$s_i = c_i e_{i1} + \dot{e}_{i1}$$  \hspace{1cm} (20)

where, $c_i$ should guarantee the polynomial $\lambda_i + c_i$ is Hurwitz, which requires that the eigenvalue of polynomial $\lambda_i + c_i = 0$ has a negative real part, that is, $c_i > 0$.

By deriving $s_i$, we can get the following results:

$$\dot{s}_i = c_i \dot{e}_{i1} + \ddot{e}_{i1} = c_i \dot{e}_{i1} + \dot{x}_{i1} - \ddot{x}_{i1d}$$  \hspace{1cm} (21)

The reaching law is taken as the constant velocity law, and its expression is as follows:

$$\dot{s}_i = -K_i \text{sign}(s) \cdot K_i > 0$$  \hspace{1cm} (22)

From (13), we can get:

$$\ddot{x}_{i1} = F_i (x_{i1}, x_{i2}) + U_i + x_{i3}$$  \hspace{1cm} (23)

By substituting (22) and (23) into (21), the sliding mode control law of the i-th joint is obtained as follows:

$$U_i = \ddot{x}_{i1d} - K_i \text{sign}(s) - c_i e_{i2} - F_i (x_{i1}, x_{i2}) - x_{i3}$$  \hspace{1cm} (24)

To weaken the impact of non-uniform traction on the joint trajectory, a moving average filter (MAF) is carried out for the desired joint acceleration $\ddot{x}_{i1d}$, and the processed desired joint acceleration is expressed by the following formula:

$$\ddot{q}_{id} (k) = \left\{ \begin{array}{ll}
\ddot{x}_{i1d} (k), & k \leq N \\
\frac{\sum_{j=k-N}^{k} \ddot{x}_{i1d} (j)}{N}, & k > N
\end{array} \right.$$  \hspace{1cm} (25)

When the system state is estimated by the nonlinear extended state observer, $\dot{\tilde{z}}_{i1} \rightarrow x_{i1}, \dot{\tilde{z}}_{i2} \rightarrow x_{i2}, \dot{\tilde{z}}_{i3} \rightarrow x_{i3}$ can be obtained. The trajectory tracking error of the i-th joint is actually $\ddot{e}_{i1} = \ddot{x}_{i1} - x_{i1d}, \dddot{e}_{i1} = \dddot{x}_{i1} - \dddot{x}_{i1d}$, and then the sliding mode control law of the i-th joint can be expressed as follows:

$$U_i = \dddot{q}_{id} - K_i \text{sign}(\dot{s}) - c_i \dddot{e}_{i1} - F_i (x_{i1}, x_{i2}) - \dddot{x}_{i3}$$  \hspace{1cm} (26)

where, $\dddot{s} = c_i \dddot{e}_{i1} + \dddot{e}_{i1}$.

To reduce the chattering, the output torque of the sliding mode control is processed by a low-pass filter before the torque is input to the manipulator, as shown in the following:

$$\tau_i = LPF \left( B^{-1} (U_i) \right)$$  \hspace{1cm} (27)

**F. STABILITY PROOF OF CLOSED-LOOP CONTROL SYSTEM**

Taking the Lyapunov function of the i-th joint as $V_{is} = \frac{1}{2} \dddot{s}_i^2$, we can get the following:

$$\dot{V}_{is} = s_i \dddot{s}_i$$  

$$= s_i (c_i \dddot{e}_{i1} + \dddot{e}_{i1})$$  

$$= s_i (c_i \dddot{e}_{i1} + \dddot{x}_{i1} - \dddot{x}_{i1d})$$  

$$= s_i (c_i \dddot{e}_{i1} + F_i (x_{i1}, x_{i2}) + U_i + x_{i3} - \dddot{x}_{i1d})$$  

$$= s_i \left( -K_i \text{sign} (\dddot{s}) + c_i \dddot{e}_{i1} - c_i \dddot{e}_{i1} \right.$$  

$$+ x_{i3} - \dddot{x}_{i3} + \dddot{q}_{id} - \dddot{x}_{i1d} \left. \right)$$  \hspace{1cm} (28)

Let $\dddot{s}_i = s_i - \dot{s}_i$, then $\dddot{s}_i = c_i \dddot{e}_{i1} + \dot{e}_{i1} - \dddot{e}_{i1} - \dddot{e}_{i1}$. Therefore, (28) can be written as follows:

$$\dot{V}_{is} = s_i \left( -K_i \text{sign} (s_i) + K_i \text{sign} (\dddot{s}_i) + c_i \dddot{e}_{i1} \right.$$  

$$- c_i \dddot{e}_{i1} + x_{i3} - \dddot{x}_{i3} + \dddot{q}_{id} - \dddot{x}_{i1d} \right)$$  

$$= -s_i K_i \text{sign} (s_i) + s_i \left( K_i \text{sign} (\dddot{s}_i) + c_i \dddot{e}_{i1} \right.$$  

$$- c_i \dddot{e}_{i1} + x_{i3} - \dddot{x}_{i3} + \dddot{q}_{id} - \dddot{x}_{i1d} \right)$$  \hspace{1cm} (29)

where, $K_i \text{sign} (\dddot{s}_i) + c_i \dddot{e}_{i1} - \dddot{e}_{i1} + x_{i3} - \dddot{x}_{i3}$ in (29) is the sum of the observation errors of each state of the extended observer, $\dddot{q}_{id} - \dddot{x}_{i1d}$ is the difference between the desired acceleration before and after filtering, and its absolute value is bounded. Therefore, let $|K_i \text{sign} (\dddot{s}_i) + c_i \dddot{e}_{i1} - \dddot{e}_{i1} + x_{i3} - \dddot{x}_{i3}| \leq \Delta_{imax}$, then (29) can be changed into the following inequality:

$$\dot{V}_{is} \leq -K_i s_i^2 + \frac{1}{2} \left( s_i^2 + \Delta_{imax}^2 \right)$$  

$$= - \left( K_i - \frac{1}{2} \right) s_i^2 + 2 \Delta_{imax}^2$$  

$$= -(2K_i - 1) V_i + \frac{1}{2} \Delta_{imax}^2$$  \hspace{1cm} (30)

**Lemma 1:** For $V : [0, \infty) \in R$, the solution of the inequality equation $\dot{V} \leq -\alpha V + f, \forall t \geq 0 \geq 0$ is as follows:

$$V (t) \leq e^{-\alpha(t-t_0)} V (t_0) + \int_{t_0}^{t} e^{-\alpha(t-t')} f (t) \tau \tau$$  \hspace{1cm} (31)

where $\alpha$ is an arbitrary constant.

According to the above **Lemma 1**, let $\alpha = 2K_i - 1, f = \frac{1}{2} \Delta_{imax}^2$, then the solution of the inequality equation $\dot{V}_{is} \leq -(2K_i - 1) V_i + \frac{1}{2} \Delta_{imax}^2$ is as follows:

$$V_{is} (t) \leq e^{-\alpha(t-t_0)} V (t_0)$$  

$$+ \frac{1}{2} \Delta_{imax}^2 \int_{t_0}^{t} e^{-\alpha(t-t')} d \tau$$  

$$= e^{-\alpha(t-t_0)} V (t_0)$$  

$$+ \frac{1}{2} \Delta_{imax}^2 \int_{t_0}^{t} e^{-\alpha(t-t')} (\alpha (t-t'))$$  

$$= e^{-2K_i - 1) (t-t_0)} V (t_0)$$  

$$+ \frac{1}{2} (2K_i - 1) \Delta_{imax}^2 \left( 1 - e^{-2K_i - 1) (t-t_0)} \right)$$  \hspace{1cm} (32)

Take $K_i > \frac{1}{2}$ and $t \rightarrow \infty$, there are

$$\lim_{t \rightarrow \infty} V_{is} (t) \leq \frac{1}{2} (2K_i - 1) \Delta_{imax}^2$$  \hspace{1cm} (33)

In other words, when $t \rightarrow \infty$, $V_{is} (t)$ converges to $\frac{1}{2} (2K_i - 1) \Delta_{imax}^2$ exponentially and the convergence rate depends on $K_i$. When $K_i$ is large enough, $\lim_{t \rightarrow \infty} V_{is} (t) \rightarrow 0, s_i \rightarrow 0$. By substituting (33) into equation (30), we can get $\dot{V}_{is} \leq -(2K_i - 1) V_i + \frac{1}{2} \Delta_{imax}^2 = -\frac{1}{2} \Delta_{imax}^2 + \frac{1}{2} \Delta_{ima}^2 = 0$.

The convergence of the extended state observer can be proved by reference [44]. The Lyapunov function of the
extended state observer is defined as $V_{io}$. Considering the single-joint closed-loop control system composed of nonlinear extended observer and sliding mode controller systematically, and the Lyapunov function can be taken as $V_i = V_{io} + V_i$. When appropriate $K_i$ and $\beta_1, \beta_2, \beta_3$ are taken, that is, when $V_i \geq 0$ can be guaranteed, $V_i = V_{io} + V_i \leq 0$. That is, when $t \to \infty$, then $s_i \to 0$, $e_i \to 0$, and its convergence rate depends on $K_i$ and $\beta_1, \beta_2, \beta_3$. This shows that the closed-loop control system of the $i$-th joint of the manipulator system (8) is stable and the sliding surface is reachable.

When the closed-loop control systems of all joints of the manipulator system (8) are stable and the sliding surfaces are reachable, the closed-loop control system of the whole manipulator system is also stable and the sliding surface is available. That is, when $t \to \infty$, then $s_i \to 0$, $e_i \to 0$ and $s_i \to 0$. That is to say, the sliding mode controller based on nonlinear extended observer proposed in this paper can make (7) approach zero stably.

**IV. SIMULATION RESULTS AND DISCUSSIONS**

**A. EXPERIMENTAL SETUP**

To verify the feasibility of the proposed ADRSMC, a series of simulation experiments were performed on a 6-DOF manipulator based on the V-REP (Virtual Robot Experiment Platform). V-REP refers to a robot dynamics simulation platform equipped with a physical engine. It is capable of simulating the robot system in the real physical world realistically. The 6-DOF manipulator applied in V-REP was a UR5 manipulator. UR5 manipulator is a serial robot consisting of six revolute joints (e.g., waist, shoulder, elbow, and wrist joint with three degrees of freedom). The D-H parameters and dynamics parameters of UR5 originated from the ur5.urdf file on the website (https://github.com/ros-industrial/universal_robot), and the main parameters are listed in Table 1. Fig.3 (1) illustrates the coordinate system established by D-H parameters, where $\odot$ denotes the tail of the arrow, and $\oplus$ represents the head of the arrow. The physical engine applied in V-REP was Bullet2.78, with the simulation step size of 10ms. UR5 robot adopted torque control mode; its joint angle range and its initial joint angle were set to $[180^\circ, -180^\circ]$ and $[90^\circ, -160^\circ, 150^\circ, 0^\circ, 90^\circ, 0^\circ]$, respectively.

To achieve the traction control of the manipulator’s end by using the HCI device, the motion trajectory of the HCI device (e.g., joysticks) should be mapped into the motion space of the robot. To be specific, the motion range of the manipulator’s end relative to the base coordinate system should be obtained. In the present study, the Monte Carlo algorithm was adopted to obtain the range of motion of the manipulator’s end in Cartesian space. First, the point cloud of the UR5 manipulator end was obtained by the Monte Carlo algorithm (Fig.3 (2)). Subsequently, by calculating the boundary range of the position point cloud, the motion range of the UR5 manipulator end was obtained as:

$$[TX_{min}, TX_{max}] = [-0.9419m, 0.9256m]$$

$$[TY_{min}, TY_{max}] = [-0.9267m, 0.9307m]$$

$$[TZ_{min}, TZ_{max}] = [-0.8486m, 1.0218m]$$

HCI device (e.g., mouse or joysticks) generally has multiple degrees of freedom, and the motion range of each DOF is generally $[P_{min}, P_{max}]$. For instance, the three-axis joystick generally has three degrees of freedom, and the motion ranges of the X-axis, Y-axis, and R-axis are overall $[0~65535]$, $[0~65535]$, $[0~65535]$, respectively. The mouse generally has two degrees of freedom, and its motion range on the screen is $[0, P_w]$, $[0, P_h]$, where $P_w$ and $P_h$ respectively denote the width and height of the mouse movement area. To achieve the traction control of the manipulator’s end by

**TABLE 1. D-H parameters and dynamic parameters of the UR5 robot.**

| Kinematics | theta(rad) | d[m] | a[m] | alpha(rad) | Dynamics | mass[kg] | center of mass[m] |
|------------|------------|------|------|------------|----------|----------|------------------|
| Joint1     | $\theta_1$ | 0.89159 | 0    | $\pi/2$    | Link1    | 3.7      | [0.0,0.02561,0.001193] |
| Joint2     | $\theta_2$ | 0    | -0.425 | 0          | Link2    | 8.393    | [0.0,0.02125,0.011336] |
| Joint3     | $\theta_3$ | 0    | -0.39225 | 0          | Link3    | 2.33     | [0.0,0.11993,0.00265] |
| Joint4     | $\theta_4$ | 0.10915 | 0    | $\pi/2$    | Link4    | 2.191    | [0.0,0.00018,0.01634] |
| Joint5     | $\theta_5$ | 0.09465 | 0    | $-\pi/2$   | Link5    | 2.191    | [0.0,0.00018,0.01634] |
| Joint6     | $\theta_6$ | 0.0823 | 0    | 0          | Link6    | 0.1879   | [0.0,0.0001159] |

**FIGURE 3. Structure and coordinate system of UR5.**

220254 VOLUME 8, 2020
employing the HCI device, the motion trajectory of the HCI device should be mapped into the motion trajectory of the manipulator in Cartesian space, and then the corresponding joint angle of the motion trajectory is solved by using the inverse kinematics algorithm of the robot. Lastly, the joint tracking in the joint space is achieved by the trajectory tracking algorithm.

If the trajectory of each DOF of the HCI device (e.g., mouse or joysticks) is set as \( P \), then the trajectory of the DOF is mapped into the trajectory \( T \) of a certain dimension in Cartesian space of the manipulator, and the mapping relationship is as follows:

\[
T = \frac{P}{P_{\text{max}} - P_{\text{min}}} \times (T_{\text{max}} - T_{\text{min}}) + T_{\text{min}} \tag{34}
\]

where \( T_{\text{max}} \) and \( T_{\text{min}} \) denote the maximum and minimum of the motion range of the manipulator end in a certain dimension of Cartesian space, respectively.

According to Fig.3 (1), the second joint axis, the third joint axis, and the fourth joint axis of the UR5 manipulator are parallel. For this reason, the UR5 robot satisfies the Pieper criterion of robot mechanism, and the closed solution of its inverse kinematics can be yielded under a variable separation method by complying with the results of robot forward kinematics.

**B. SYNTHETIC TERMINAL TRACTION CONTROL EXPERIMENT**

To verify the feasibility of ADRSMC, ADRSMC and the comparative algorithms are applied to synthetic terminal traction control experiments, respectively.

Since ADRSMC is designed under the framework of ADRC, some parameters of ADRSMC can also learn from the design experience of ADRC. For instance, the higher the value of \( r_{\delta 0} \), the faster the tracking speed of differential tracker, and the higher the value of \( h_{0} \), the more significant filtering effect the differential tracker will exert. On the whole, the value of \( \delta_i \) can be taken as \( 5h \leq \delta_i \leq 10h \). In the experiment here, the parameters of ADRSMC were set below: \( r_{\delta 0} = 1600, h_{0} = 2h \); In nonlinear extended state observer: \( \beta_{i1} = 60, \beta_{i2} = 300, \beta_{i3} = 600, \alpha_{11} = 0.5, \alpha_{12} = 0.25, \delta_i = 10h \); In sliding mode control law: \( K_i = 20, c_i = 30 \). Based on [44], the mentioned parameters were regulated by the empirical trial and error manner during the experiment.

As a comparison, we used the computed torque controller (CTC) and a computed torque controller based on RBF compensator [17], [46] (CTC-RBF) to control the UR5 manipulator respectively for the same experiment. The computed torque controller used in the present study was: \( \tau = M_0(q) \left( \dot{q}_{\text{d}} + K_p \dot{e} + K_d \dot{e} + K_i \int e \text{dt} \right) + C_0(q, \dot{q}) \dot{q} + G_0(q) \), where \( K_p = 100, K_d = 2\sqrt{100}, K_i = 1 \) refers to the optimal set of parameters after many tests. CTC-RBF applied in the present study is: \( \tau = M_0(q) \left( \dot{q}_{\text{d}} + K_p \dot{e} + K_d \dot{e} + K_i \int e \text{dt} \right) + C_0(q, \dot{q}) \dot{q} + G_0(q) + \dot{\ddot{f}}(\cdot) \), where \( \dot{\ddot{f}}(\cdot) \) denotes the output of RBF, i.e., the system uncertainty. The number of hidden layers of RBF was set to 5, the initial weight was set to 0.1, and the parameters of the Gaussian function were set to \( c_1 = [-2 -1 0 1 2] \) and \( b_1 = 10 \). For fairness, other parameters in CTC-RBF were identical to those in CTC, and the desired joint acceleration \( \ddot{q}_{\text{d}} \) in CTC and CTC-RBF were processed with a moving average filter as well. For fairness, the desired joint acceleration \( \ddot{q}_{\text{d}} \) in CTC was also processed with a moving average filter.

In the present section, two groups of experiments are set: (a) using CTC, CTC-RBF, and ADRSMC to control the UR5 manipulator for respectively tracking the uniform trajectory, with the experimentally achieved results presented in Fig.4 (1), (3), (5); (b) using CTC, CTC-RBF, and ADRSMC to control the UR5 manipulator for the respective track of the non-uniform trajectory, with the experimentally achieved results shown in Fig.4 (2), (4), (6). To be specific, the non-uniform trajectory referred to a circular trajectory with \([0–0.2]\) random jitter introduced into the radius of the uniform circular trajectory, which aimed to simulate the non-uniform terminal traction (e.g., hand tremor) during the teleoperation traction.

Fig.4 (1) compares the manipulator’s end trajectories under the use of CTC, CTC-RBF, and ADRSMC to control the UR5 manipulator for the respective track of the uniform trajectory. Fig.4 (3) shows a comparison of square root errors generated by the three algorithms during the traction. Fig.4 (5) shows the comparison of mean error and variance. Fig.4 (2) shows a comparison of the manipulator’s end trajectories under the use of CTC, CTC-RBF, and ADRSMC to control the UR5 manipulator for the respective track of the non-uniform trajectory. Fig.4 (4) shows a comparison of square root errors generated by the three algorithms in the traction process. Fig.4 (6) shows the comparison of mean error and variance.

It can be seen from Fig.4 (1) and Fig.4 (2) that all algorithms can track the uniform and non-uniform trajectories stably. It can be seen from Fig.4 (3) and Fig.4 (4) that ADRSMC can rapidly reduce the tracking error at the initial stage of traction. Although the error rebounds slightly later, it decreases again rapidly. Compared with CTC and CTC-RBF, the tracking error of ADRSMC is smaller in the middle and late stages of traction. According to the mean error and variance presented in Fig.4 (5) and Fig.4 (6), the average level of tracking error and the consistency between the tracking trajectory and the desired trajectory during the whole traction are indicated. Fig.4 (5) and Fig.4 (6) suggest that compared with CTC and CTC-RBF, the proposed ADRSMC is capable of achieving synthetic terminal traction with less error.

**C. REAL TERMINAL TRACTION CONTROL EXPERIMENT**

To verify the performance of ADRSMC in depth, the terminal traction control of the UR5 manipulator was achieved by the HCI device. In this experiment, the HCI device was an ordinary mouse (it can indeed be a joystick or force feedback device, etc.). We achieved the terminal traction control of the UR5 manipulator through the mouse point traction and judged the performance of the control algorithm by...
comparing the consistency of the traction trajectory of the mouse and the motion trajectory of the manipulator’s end. The comparison algorithms still used CTC and CTC-RBF, as employed in the previous section, and the parameter settings complied with the parameters in the previous section.

In this section, two groups of experiments were set, i.e., the mouse pointer was used to guide the UR5 manipulator, and the manipulator’s end was made to track the rectangular and X-shaped traction track, as generated by the mouse pointer. The control algorithm in the traction process used CTC, CTC-RBF, and ADRSMC respectively. To deliberately produce non-uniform traction, the speed of the mouse during rectangular traction and X-shaped traction is non-uniform. The specific method is elucidated below. During the mouse traction, the mouse accelerates sharply from the starting point, then decelerates sharply at the turning point, accelerates sharply after the turning point, and subsequently decelerates sharply before reaching the next turning point, etc.; next, the rectangular track traction control and X-shaped track traction control are completed.

Fig. 5 (1) shows the comparison between the actual trajectory of the UR5 manipulator’s end and the mouse trajectory when CTC was used and the mouse traction trajectory was a non-uniform rectangular trajectory. Fig. 5 (2) shows the
Comparison between the actual trajectory of the UR5 manipulator’s end and the mouse trajectory when CTC-RBF was used and the mouse traction track was the non-uniform rectangular trajectory. Fig. 5 (3) shows the comparison between the actual trajectory of the UR5 manipulator’s end and the mouse trajectory under the use of ADRSMC and the mouse traction track pertaining to the non-uniform rectangular trajectory. Fig. 5 (4) presents the square root error between the
Fig. 6 (1)∼(5) shows the trajectory comparison, square root error change, mean value, and variance comparison of the UR5 manipulator’s end actual trajectory and mouse traction trajectory (mouse traction trajectory was non-uniform X-shaped) when CTC, CTC-RBF, and ADRSMC were used respectively. It can be seen from Fig.6 (1) and (2) that the tracking trajectory generated by CTC and CTC-RBF are not non-uniform rectangular traction trajectory and the actual trajectory of the UR5 manipulator’s end under the use of CTC, CTC-RBF, and ADRSMC. Fig.5 (5) illustrates the mean error and variance of the three algorithms, which can show the consistency of the two trajectories. In other words, the smaller the mean error and variance, the more consistent the manipulator’s end tracking trajectory will be with the mouse trajectory. It can be seen from Fig.5 (1), (2), (3), and (4) that all the algorithms have large errors at the initial stage of mouse traction. This is due to the long distance between the initial position of the manipulator’s end and the initial position of the mouse pointer. However, in the middle and late stages of traction, ADRSMC enables the trajectory of the UR5 manipulator’s end to efficiently follow the mouse trajectory. However, CTC and CTC-RBF cannot make the UR5 manipulator’s end completely follow the mouse trajectory. Fig.5 (5) indicates that the tracking trajectory generated by ADRSMC is more consistent with the mouse trajectory compared with CTC and CTC-RBF.

Fig. 7. Comparison results of trajectory change and error change of each joint during non-uniform rectangular traction.
FIGURE 8. Comparison results of trajectory change and error change of each joint during non-uniform X-shaped traction.

very consistent with the traction trajectory, especially at the turning point of the traction trajectory, the tracking trajectory vibrates violently. It can be seen from the green dotted line and black double dash line in Fig.6 (4) that the tracking trajectory error attributed to CTC and CTC-RBF fluctuates violently. As can be seen in Fig.6 (3), the initial error between the tracking trajectory generated by ADRSMC and the traction trajectory is large, but the two kinds of trajectories can be basically consistent soon. As indicated by the red solid line in Fig.6 (3), the square root error can be reduced to a lower level efficiently. According to Fig.6 (5), compared with CTC and CTC-RBF, the tracking trajectory generated by ADRSMC is more consistent with the mouse traction trajectory.

During the experiment of the non-uniform rectangular traction and non-uniform X-shaped traction, this study reported that with the use of CTC and CTC-RBF, the trajectory of the UR5 manipulator’s end vibrated violently and then diverged multiple times. In other words, the terminal traction control of the UR5 manipulator is difficult to achieve by applying CTC and CTC-RBF under the non-uniform traction. However, under the use of ADRSMC, it enabled the UR5 manipulator to track the traction trajectory stably, whether in uniform or non-uniform situation.
FIGURE 9. Mean error and variance of each joint track for non-uniform rectangular traction and non-uniform X traction.

Fig.7 and Fig.8 present the compared results between the joint position of each joint of the UR5 manipulator and the desired joint position, as well as those of joint tracking error when the mouse traction trajectory pertained to the non-uniform rectangular trajectory and non-uniform X-shaped trajectory.

As indicated by Fig.7, under the use of ADRSMC, the position of each joint (red solid line) is more consistent with the desired position (blue dotted line). However, the position of each joint (green dotted line and black double dash) significantly deviates from the desired position (blue dotted line) of the manipulator under the use of CTC and CTC-RBF. From the compared results of joint position errors in Fig.7 and Fig.8, it can be seen that the fluctuation of the red solid line (ADRSMC) is smaller than that of the green dotted line (CTC) and black double dash (CTC-RBF), and it is closer to zero levels. This indicates that when the mouse traction trajectory is non-uniform, ADRSMC can make the UR5 manipulator produce a smaller tracking error.

To more vividly show the performance difference between CTC, CTC-RBF, and ADRSMC, we show the trajectory errors of the UR5 manipulator joints in the form of an error bar graph when the traction trajectory was non-uniform rectangular trajectory and non-uniform X-shaped trajectory, as shown in Fig.9.

Fig.9 (1) shows the compared results of mean error and variance of each joint of the UR5 manipulator under non-uniform rectangular traction. Fig.9 (2) shows the compared results of mean error and variance of each joint of the UR5 manipulator under non-uniform X-shaped traction. The blue histogram in Fig.9 (1) and Fig.9 (2) shows the average error and variance of each joint under the use of CTC. The orange histogram shows the comparison of the average error and variance of each joint under the use of CTC-RBF. The yellow histogram compares the average error and variance of each joint under the application of ADRSMC.

Fig.9 (1) and Fig.9 (2) indicate that under the use of ADRSMC, the mean error between the tracking trajectory of each joint of the manipulator and the traction trajectory is relatively small, whether the non-uniform rectangular traction or the non-uniform X-shaped traction. However, under the use of CTC and CTC-RBF, the mean error between the tracking trajectory and the traction trajectory of each joint of the manipulator are relatively large. The error bar graph can also reflect the consistency between the tracking trajectory and the traction trajectory, that is, the smaller the mean error and variance, the more consistent the tracking trajectory and the traction trajectory will be, and the stronger traction ability the control algorithm will exhibit.

As revealed from the error bar graph presented in Fig.9, the mean error and variance illustrated in the yellow histogram are smaller than those shown in the blue and orange histogram, demonstrating that ADRSMC outperforms CTC and CTC-RBF in the aspect of non-uniform traction.

V. CONCLUSION

The present study proposes an active disturbance rejection sliding mode traction control system to deal with the sudden change of joint angular acceleration and the jitter attributed to non-uniform traction during terminal traction of teleoperation manipulator by HCI devices. In such a system, the effect of non-uniform traction on the robot joints is considered disturbance, and then an active disturbance rejection sliding mode controller (ADRSMC) is designed for each joint, and the effect disturbance is eliminated online by the anti-interference ability of ADRSMC. The stability of ADRSMC is proved by the Lyapunov method. To prove the terminal traction ability of ADRSMC, a series of terminal traction control experiments were carried out on the UR5 manipulator. The experimentally achieved results show that the active disturbance rejection sliding mode traction control system proposed in the present study can achieve fast, stable and high-precision terminal traction control of the manipulator by human-computer interaction device.

In the present study, we use 2D HCI devices to achieve the terminal traction control of the teleoperation manipulator.
Future work will explore the terminal traction control by 3D HCI devices (e.g., the Wii gamepad). Besides, to efficiently estimate system uncertainties and unknown external disturbances in finite-time, more robust finite-time differentiators and disturbance estimators are to be explored, including the sliding mode differentiator and the disturbance compensator designed in [45]. Moreover, the study will be conducted on other sliding mode control algorithms capable of reducing chattering or achieving faster convergence based on the ADRC framework (e.g., high-order sliding mode control or non-singular fast terminal sliding mode control).

REFERENCES

[1] P. Ji, A. Song, P. Xiong, P. Yi, X. Xu, and H. Li, “Egocentric-vision based hand gesture posture control system for reconnaissance robots,” J. Intell. Robotic Syst., vol. 87, nos. 3–4, pp. 583–599, Sep. 2017.
[2] G. Wu, Z. Xie, C. Guo, and H. Liu, “Kalman smoothing with soft inequality constraints for space robot teleoperation,” Int. J. Adv. Robotic Syst., vol. 15, no. 1, 2018, Art. no. 25006390, doi: 10.1177/1729881417746024.
[3] W. Wang and S. Cong, “Nonlinear computed torque control for a high-speed planar parallel manipulator,” Mechantronics, vol. 19, no. 6, pp. 987–992, Sep. 2009, doi: 10.1016/j.mechatronics.2009.04.002.
[4] M. H. Sangdani, A. R. Tavakolpour-Saleh, and A. Lotfavar, “Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment,” Eng. Appl. Artif. Intell., vol. 67, pp. 24–38, Jan. 2018, doi: 10.1016/j.engappai.2017.09.014.
[5] X. Wang and B. Hou, “Trajectory tracking control of a 2-DOF manipulator using computed torque control combined with an implicit Lyapunov function method,” J. Mech. Sci. Technol., vol. 32, no. 6, pp. 2803–2816, Jun. 2018, doi: 10.1007/s12206-018-03537-6.
[6] W. Peng, Z. Lin, and J. Su, “Computed torque control-based composite nonlinear feedback controller for robot manipulators with bounded torques,” IET Control Theory Appl., vol. 3, no. 6, pp. 701–711, Jun. 2009, doi: 10.1049/ietcta.2008.0259.
[7] Y. Chen, G. Ma, S. Lin, and J. Gao, “Adaptive fuzzy computed-torque control for robot manipulator with uncertain dynamics,” Int. J. Adv. Robotic Syst., vol. 9, no. 6, p. 237, Dec. 2012, doi: 10.5772/52463.
[8] C. Chen, C. Zhang, T. Hu, H. Ni, and W. Luo, “Model-assisted extended state observer-based computed torque control for trajectory tracking of uncertain robotic manipulator systems,” Int. J. Adv. Robotic Syst., vol. 15, no. 5, 2018, Art. no. 25590819, doi: 10.1177/1729881418801738.
[9] J. Nubert, J. Kohler, V. Berenz, F. Allgower, and S. Trimpe, “Safe and fast trajectory tracking control of robot manipulators,” IEEE/ASME Trans. Mechatronics, vol. 23, no. 1, pp. 8701–8712, 2019, doi: 10.1109/TMECH.2018.286222.
[10] H. Liu, J. Sun, J. Nie, and L. Zou, “Observer-based adaptive second-order non-singular fast terminal sliding mode controller for robotic manipulators,” Asian J. Control, pp. 1–10, Jul. 2020, doi: 10.1002/asjc.2369.
[11] V.-C. Nguyen, A.-T. Vo, and H.-J. Kang, “A non-singular fast terminal sliding mode control based on third-order sliding mode observer for a class of second-order uncertain nonlinear systems and its application to robot manipulators,” IEEE Access, vol. 8, pp. 78109–78120, 2020, doi: 10.1109/ACCESS.2020.2989613.
[12] Y.-J. Liu, L. Ma, L. Liu, S. Tong, and C. L. P. Chen, “Adaptive neural network learning controller design for a class of nonlinear systems with time-varying state constraints,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 1, pp. 66–75, Jan. 2020, doi: 10.1109/TNNLS.2019.2975597.
[13] S. Zhang, Y. Dong, Y. Ouyang, Z. Yin, and K. Peng, “Adaptive neural control for robotic manipulators with output constraints and uncertainties,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 11, pp. 5554–5564, Nov. 2018, doi: 10.1109/TNNLS.2018.2830827.
[14] X. Yu, S. Zhang, Q. Fu, C. Xue, and W. Sun, “Fuzzy logic control of an uncertain manipulator with full-state constraints and disturbance observer,” IEEE Access, vol. 8, pp. 24284–24295, 2020, doi: 10.1109/ACCESS.2020.2992255.
[15] F. Wang, Z.-Q. Chao, L.-B. Huang, H.-Y. Li, and C.-Q. Zhang, “Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode,” Cluster Comput., vol. 22, no. 53, pp. 5799–5809, May 2019, doi: 10.1007/s10586-017-1538-4.
[16] M.-D. Tran and H.-J. Kang, “A novel adaptive fast-time tracking control for robotic manipulators using nonsingular terminal sliding mode and RBF neural networks,” Int. J. Precis. Eng. Manuf., vol. 17, no. 7, pp. 863–870, Jul. 2016, doi: 10.1007/s12541-016-0105-x.
[17] C. Liu, Z. Zhao, and G. Wen, “Adaptive neural network control with optimal number of hidden nodes for trajectory tracking of robot manipulators,” Neurocomputing, vol. 350, pp. 136–145, Jul. 2019, doi: 10.1016/j.neucom.2019.03.043.
[18] Z. Bungil and O. Karahan, “A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control,” Expert Syst. Appl., vol. 38, no. 1, pp. 1017–1031, Jan. 2011, doi: 10.1016/j.eswa.2010.07.131.
H. Zhang and W. Li, “Tracking control using hierarchical fuzzy-regulated optimal approach for robotic manipulators,” Proc. Inst. Mech. Eng. C, J. Mech. Eng. Sci., vol. 233, no. 2, pp. 635–648, Jan. 2019, doi: 10.1177/0954406218769937.

J. Han, “From PID to active disturbance rejection control,” IEEE Trans. Ind. Electron., vol. 56, no. 3, pp. 900–906, Mar. 2009, doi: 10.1109/TIE.2008.2011621.

D. Liu, Q. Gao, Z. Chen, and Z. Liu, “Linear active disturbance rejection control of a two-degrees-of-freedom manipulator,” Math. Problems Eng., vol. 2020, pp. 1–19, May 2020, doi: 10.1155/2020/669207.

X. Cheng, X. Tu, Y. Zhou, and R. Zhou, “Active disturbance rejection control of multi-joint industrial robots based on dynamic feedback,” Electronics, vol. 8, no. 5, p. 591, May 2019, doi: 10.3390/electronics8050591.

L. A. Castañeda, A. Luviano-Juárez, G. Ochoa-Ortega, and I. Chairez, “Tracking control of uncertain time delay systems: An ADRC approach,” Control Eng. Pract., vol. 78, pp. 97–104, Sep. 2018, doi: 10.1016/j.conengprac.2018.06.015.

J. Yao and W. Deng, “Active disturbance rejection adaptive control of uncertain nonlinear systems: Theory and application,” Nonlinear Dyn., vol. 89, no. 3, pp. 1611–1624, Aug. 2017, doi: 10.1007/s11071-017-3538-6.

R. Fareh, M. Al-Shabi, M. Bettayeb, and J. Ghommem, “Robust active disturbance rejection control for flexible link manipulator,” Robotica, vol. 38, no. 1, pp. 118–135, Jan. 2020, doi: 10.1017/S026357471900050X.

L. Zhao, H. Cheng, and T. Wang, “Sliding mode control for a two-joint coupling nonlinear system based on extended state observer,” ISA Trans., vol. 73, pp. 130–140, Feb. 2018, doi: 10.1016/j.isatra.2017.12.027.

H. Pan and W. Sun, “Nonlinear output feedback finite-time control for vehicle active suspension systems,” IEEE Trans. Ind. Informat., vol. 15, no. 4, pp. 2073–2082, Apr. 2019, doi: 10.1109/TII.2018.2866518.

P. Van Cuong and W. Y. Nan, “Adaptive trajectory tracking neural network control with robust compensator for robot manipulators,” Neural Comput. Appl., vol. 27, no. 2, pp. 525–536, Feb. 2016, doi: 10.1007/s00521-015-1873-4.