



## OPEN Tracking control of humanoid manipulator using sliding mode with neural network and disturbance observer

Yina Wang<sup>1</sup>, Yanjun Yu<sup>1✉</sup>, Ziteng Wang<sup>1</sup>, Liyao Ji<sup>1</sup>, Junyou Yang<sup>1</sup> & Shuoyu Wang<sup>2</sup>

The nursing robot, equipped with a 6-degree-of-freedom (6-DOF) humanoid manipulator, has been applied in elderly and disabled care to execute complex and random nursing tasks with its advantages in automation and intelligence. Especially, when the nursing robot performs daily care tasks such as serving tea and pouring water, the good trajectory tracking performance of its manipulator is a crucial capability. However, nonlinear coupling, model uncertainty, joint friction, unknown external disturbances, and particularly the fact that manipulator does not satisfy Pieper criterion—are the main challenges, which degrade control performance. Few existing studies have simultaneously addressed all these issues to improve the control accuracy of the manipulator. Therefore, to achieve the good tracking performance for manipulator, a robust control method combining sliding mode control (SMC), radial basis function neural network (RBFNN), and nonlinear disturbance observer (NDO) is proposed. An improved Jacobian-based gradient descent method solves inverse kinematics, with the improved gradient descent driven inverse kinematics (IGDIK) module ensuring accuracy; RBFNN compensates for model uncertainty; NDO handles disturbances and friction. Simulations and experiments demonstrate enhanced trajectory tracking accuracy and stability, validating its effectiveness for the target manipulator.

**Keywords** 6-DOF humanoid manipulator, RBFNN, Sliding mode control, Disturbance observer, Dynamic model, MATLAB/Simulink, Robot operating system (ROS)

### Abbreviations

$q$	Joint angles
$\tau$	Joint torque vector
$d$	External disturbances
$F(q)$	Joint friction
$e$	Position tracking error
$pv_{t-1}$	The upper limit of the time gradient
$\alpha$	Learning rate
$p$	Attenuation factor
$r$	Sliding variables
$h_j(x)$	The nonlinear activation function of the $j$ th hidden node
$b_j$	Base width
$c_j$	The center vector of the Gaussian function of the $j$ -th node
$M(q)$	The symmetric positive definite inertia matrix
$C(q, \dot{q})$	The Coriolis matrix
$G$	The gravity force
$F(\dot{q})$	The joint friction
$\rho$	Compound disturbance
$m$	The number of input
$k$	The number of hidden layer nodes

<sup>1</sup>School of Electrical Engineering, Shenyang University of Technology, Shenyang 110870, China. <sup>2</sup>Department of Intelligent Mechanical Systems Engineering, Kochi University of Technology, Kami 7828502, Japan. ✉email: yu.yanjun@smail.sut.edu.cn

$\phi(x)$	The Gaussian function
$L(q, \dot{q})$	The gain matrix
$\hat{w}$	The weight of the neural network
$\hat{\rho}$	Composite interference

Population aging is intensifying elderly care pressure. Nursing robots alleviate this pressure by performing complex, random tasks—such as feeding and medication delivery—in homes and hospitals. Our laboratory has developed a specific nursing robot (as shown in Fig. 1), with its 6-DOF humanoid manipulator as the core for task execution. When performing tasks, the manipulator needs high-precision trajectory tracking. Thus, designing a high-precision trajectory tracking controller is key to ensuring the robot's reliable operation.

However, the humanoid manipulator faces multiple challenges in practical applications, directly limiting improvements in trajectory tracking accuracy. On one hand, its humanoid structure violates the Pieper criterion, making inverse kinematics difficult to solve and hindering the rapid generation of accurate joint control commands. On the other hand, issues like dynamic model uncertainty, unknown external disturbances, and joint friction in nursing scenarios further increase control difficulty. These problems are mutually coupled, and conventional control methods struggle to address them simultaneously. Therefore, it is necessary to develop an appropriate controller to simultaneously address all these issues to improve the control accuracy of the manipulator.

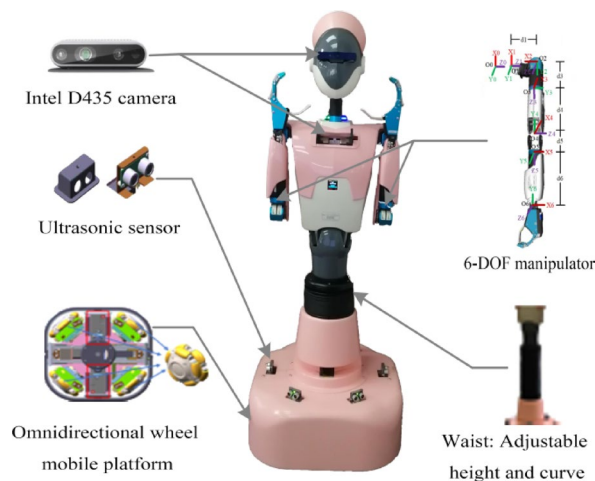
## Literature review

For the problem of inverse kinematics solving in manipulators that do not satisfy the Pieper criterion, relevant research has been conducted. Dereli et al.<sup>1</sup> employed the Firefly algorithm for 7-DOF manipulators. Bai et al.<sup>2</sup> proposed a hybrid FABRIK-ANN method. Zhou et al.<sup>3</sup> reduced 6-DOF calculation time. Ames et al.<sup>4</sup> used deep learning. Gao et al.<sup>5</sup> optimized BP neural networks. However, these methods overlook its structural nonlinearity, limiting practicality. This study thus proposes an improved gradient descent method for such non-Pieper systems, validated for high accuracy, efficiency, and resistance to local optima.

SMC is valued for its simplicity, strong robustness, and fast response, making it relevant for manipulator control. For example, Yin et al.<sup>6</sup> developed an adaptive fuzzy SMC. Tran et al.<sup>7</sup> integrated RBFNN with SMC for 3-DOF manipulators. Yin et al.<sup>8</sup> achieved kinematic control of humanoid upper-body robots via virtual flexible joint dynamics and a quasi-sliding mode observer. Xiao et al.<sup>9</sup> validated SMC on a six-axis BU-RASL manipulator. Dachang et al.<sup>10</sup> proposed an adaptive back-stepping control and validated it on a planar two-link. Su et al.<sup>11</sup> proposed a non-singular terminal SMC with an integral sliding surface. However, Tran et al.<sup>7</sup> treated model uncertainty as a disturbance, failing to address it in practical nursing tasks, highlighting the need for improved strategies. To address this, Song et al.<sup>12</sup> designed a nonsingular fast terminal sliding mode controller with a new nonlinear disturbance observer to enhance control accuracy and anti-interference ability. Wei et al.<sup>13</sup> introduced a disturbance observer.

Beyond SMC, RBFNN excel in dynamic model approximation for nonlinearities. Liu et al.<sup>14</sup> showed RBFNN outperforms BP networks in generalization. Stoffel et al.<sup>15</sup> noted its superiority over FFNN and DCNN. Narayan et al.<sup>16</sup> used RBFNN for uncertain dynamics. Xin et al.<sup>17</sup> combined it with fuzzy SMC. Sun et al.<sup>18</sup> reduced chattering via T-S fuzzy switching. Notably, these studies<sup>16–18</sup> overlook unforeseen external disturbances, limiting applicability in care environments.

To address external disturbances, researchers have optimized SMC. Li et al.<sup>19</sup> and Nguyen et al.<sup>20</sup> optimized SMC for convergence and robustness. Sun et al.<sup>21</sup> used full-drive models. Chen et al.<sup>22</sup> integrated RBFNN with disturbance observers for better control. Narayan et al.<sup>23</sup> and Liu et al.<sup>24</sup> combined RBFNN with fuzzy control. However, these methods remain unvalidated for 6-DOF humanoid manipulators, which face unique,



**Fig. 1.** Seed Noid R7F nursing service robot.

unaddressed challenges including non-Pieper criterion models, model uncertainty, joint friction, and external disturbances.

In this paper, a sliding mode robust control strategy integrating RBFNN and NDO, combined with an improved Jacobian-based gradient descent method, is proposed. Contributions include: (1) An improved Jacobian-based gradient descent method, effectively solving inverse kinematics for 6-DOF non-Pieper humanoid manipulators with higher accuracy and smoothness. (2) A novel RBFNN + NDO control strategy: RBFNN approximates the dynamic model with online parameter adjustment via weight adaptive law; NDO compensates composite disturbances (friction + external disturbances) in real time.

## System description

This paper focuses on the 6-DOF humanoid manipulator of the Seed Noid R7F nursing service robot, which is the core research object. The Seed Noid R7F is a humanoid robot with a life-size structural design, as illustrated in Fig. 1. It has a height of 141–170 cm (with a prismatic joint in the legs), a weight of approximately 80 kg, and two identical 6-DOF robotic manipulators. In addition, the Seed Noid R7F robot is equipped with an industrial motherboard, a Linux system controller based on a mini PC, an Android touchscreen tablet, four omnidirectional wheels under the base layer, and six ultrasonic sensors (0.02–3 m) on the base layer. Furthermore, it is equipped with two 3D D435 depth cameras (one on the head and one on the chest), two angle sensors (0.06–4 m,  $-120^\circ \sim 120^\circ$ ), two remote control boards for control and communication, and two rechargeable DC 12 V/22Ah lead batteries, mainly used to drive the motors and operate the circuit board.

## Dynamics and properties

The equations of the dynamic model of the robot are obtained using the Lagrange equation. The proposed control system of the humanoid robotic manipulator considers the effects of joint friction and unknown external disturbances. Thus, the complete dynamic model<sup>25</sup> is as follows:

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) + \mathbf{F}(\dot{q}) = \tau + d \quad (1)$$

where,  $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$  are the angle, the angular velocity, and the angular acceleration of the joints, respectively,  $\mathbf{M}(q) \in \mathbb{R}^{n \times n}$  is the symmetric positive definite inertia matrix,  $\mathbf{C}(q, \dot{q}) \in \mathbb{R}^{n \times n}$  is the Coriolis matrix,  $\mathbf{G}(q) \in \mathbb{R}^n$  is the gravity force, and  $\mathbf{F}(\dot{q}) \in \mathbb{R}^n$  is the joint friction.  $d \in \mathbb{R}^n$  is the external disturbance with an upper bound  $\|d\| \leq d_M$ , with  $d_M$  denoting a well-known positive constant,  $\|\cdot\|$  is the standard Euclidean norm, and  $\tau \in \mathbb{R}^n$  represents the joint torque vector. This dynamic model has the following properties<sup>26</sup>.

**Property 1** The inertia matrix  $\mathbf{M}(q)$  is a symmetric positive definite matrix. There exist positive constants  $m_{\min}, m_{\max}$  that satisfy the following inequality holds:

$$m_{\min} \|x\|^2 \leq x^T \mathbf{M}(q) x \leq m_{\max} \|x\|^2 \quad (2)$$

**Property 2**  $\dot{\mathbf{M}}(q) - 2\mathbf{C}(q, \dot{q})$  is a skew-symmetric matrix.

$$x^T [\dot{\mathbf{M}}(q) - 2\mathbf{C}(q, \dot{q})] x = 0, \forall x \in \mathbb{R}^n \quad (3)$$

**Assumption 1** The mass and length of the target gripped by the humanoid robotic manipulator are limited. Therefore, the inertia matrix  $\mathbf{M}(q)$ , the Coriolis matrix  $\mathbf{C}(q, \dot{q})$ , and the gravity force  $\mathbf{G}(q)$  are bounded.

$$\begin{cases} M_m \leq \|\mathbf{M}(q)\| \leq M_M, & \forall q \in \mathbb{R}^n \\ \|\mathbf{C}(q, \dot{q})\| \leq C_M \|\dot{q}\|, & \forall q, \dot{q} \in \mathbb{R}^n \\ \|\mathbf{G}(q)\| \leq G_M, & \forall q \in \mathbb{R}^n \end{cases} \quad (4)$$

where,  $M_m, M_M, C_M$  and  $G_M$  are some well-known positive constants.

**Assumption 2** Treat joint friction  $\mathbf{F}(\dot{q}) \in \mathbb{R}^n$  and external disturbance  $d \in \mathbb{R}^n$  are considered compound disturbance  $\rho$ . It is Assumed that the derivative of the compound disturbance is unknown but in a bounded interval:

$$\rho = d - \mathbf{F}(\dot{q}) \quad (5)$$

By multiplying both sides of Eq. (1) by  $\mathbf{M}^{-1}(q)$  and solving for  $\ddot{q}$ , one obtains:

$$\ddot{q} = \mathbf{M}^{-1}(q) [\tau + \rho - \mathbf{C}(q, \dot{q})\dot{q} - \mathbf{G}(q)] \quad (6)$$

For manipulator models with compound disturbances and parametric uncertainties, we propose a sliding-mode robust controller using RBFNNs. This approach combines global approximation and a NDO to ensure asymptotic convergence of sliding surfaces and tracking errors to zero. The position tracking error  $e(t) \in \mathbb{R}^n$  represents the difference between the actual joint angle  $q$  and the desired joint angle  $q_d$  (i.e., the deviation between actual and ideal motion).

$$e = q - q_d \quad (7)$$

Given the desired joint position  $q_d$ , few studies have addressed trajectory tracking control for multi-degree-of-freedom (DOF) robotic manipulators. However, their industrial applications have grown rapidly in recent years. As the deployment of these systems expands, solving trajectory tracking optimization and tracking error convergence challenges becomes critical. Therefore, developing advanced control strategies to enhance trajectory-tracking performance is essential.

### Adaptive iterative optimization for kinematic solution

The kinematic model of humanoid robotic manipulators is inherently highly nonlinear and coupled. As illustrated in Fig. 2, these manipulators replicate human physiological structures. Due to their anthropomorphic design, they often violate the criterion of Pieper, making traditional geometric, algebraic, and iterative algorithms ineffective for solving inverse kinematics.

To address this, an improved gradient descent method is proposed, with its core iterative formula derived as follows:

The objective cost function is defined as the difference between the desired and actual poses:

The objective cost function is defined as the difference between the desired and actual poses:

$$\begin{aligned} F &= \frac{1}{2} (X_{target} - X)^T (X_{target} - X) \\ &= \frac{1}{2} (X_{target} - f(\theta))^T (X_{target} - f(\theta)) \end{aligned} \quad (8)$$

where,  $X_{target}$  and  $X = f(\theta)$  are the desired and actual poses, respectively. The Jacobian matrix represents the mapping relationship between joint and terminal pose velocity:

$$\dot{X} = J(q)\dot{q} \quad (9)$$

Based on the target cost function in Eq. (8) and the Jacobian matrix in Eq. (9), the basic iterative formula of inverse kinematics is derived:

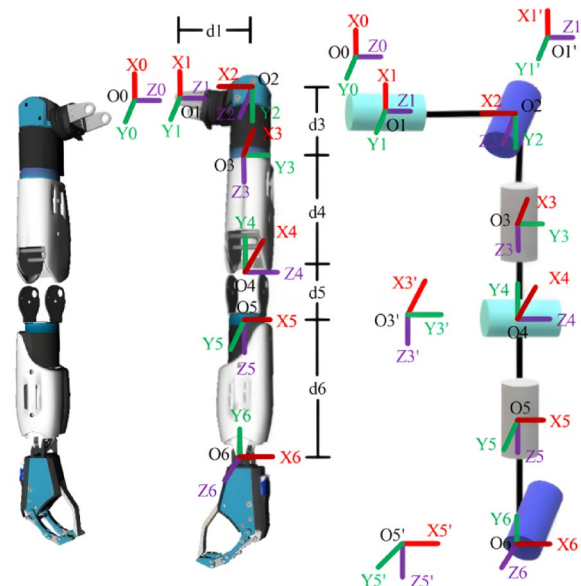
$$\begin{aligned} \Delta\theta &= \alpha ((X_{target} - X)^T \frac{\partial f(\theta)}{\partial \theta})^T \\ &= \alpha J^T(\theta) (X_{target} - X) \end{aligned} \quad (10)$$

The standard gradient descent method exhibits limitations such as tortuous optimization paths, slow convergence rates, and susceptibility to local minima. To overcome these, a momentum term is introduced into the framework:

$$\Delta\theta = pv_{t-1} + \alpha J^T(\theta) \Delta X \quad (11)$$

where,  $pv_{t-1}$  is the upper time gradient,  $\alpha$  is the learning rate, and  $p$  is the attenuation factor.

To further improve calculation accuracy, a second-order momentum term is introduced:



$$r_t = r_{t-1} + (\mathbf{J}^T(\theta) \Delta X)^2 \quad (12)$$

where,  $r_t$  is the second-order momentum (exponentially weighted moving average of squared gradients). Incorporating this into the iterative update rule, the proposed IGDIK control module is formulated as:

$$\Delta\theta = pv_{t-1} + \frac{a}{\sqrt{r_t + \sigma}} \mathbf{J}^T(\theta) \Delta X \quad (13)$$

Among them,  $\sigma$  is a small constant to stabilize numerical calculations and avoid division by zero.

### IGDIK control module

Equation (13) integrates first-order and second-order momentum terms to optimize the inverse kinematics solution process, with its core mechanism as follows:

**First-order momentum ( $pv_{t-1}$ ):** This term introduces historical gradient information, forming a weighted sum of past update directions. As illustrated in Fig. 3, it enables the algorithm to maintain “inertia” during iteration, overcoming local optima by continuing the search in regions where the objective function value may decrease, even when trapped in suboptimal points. In areas with consistent historical gradient directions, it accelerates convergence, reducing the time required for inverse kinematics solving.

**Second-order momentum ( $\sqrt{r_t + \sigma}$ ):** This term quantifies the scale of historical gradient magnitudes, dynamically adjusting the effective learning rate  $a$ . When gradients exhibit high volatility (large  $r_t$ ), the denominator increases, reducing the step size to facilitate fine-grained exploration of the solution space. Conversely, when gradients are stable (small  $r_t$ ), the denominator decreases, increasing the step size to accelerate convergence toward the global optimum.

The improved gradient descent algorithm flow is shown in Fig. 4. This mechanism ensures the algorithm balances solution accuracy and convergence speed, effectively addressing the limitations of traditional gradient descent methods and enabling efficient inverse kinematics solving for non-Pieper humanoid manipulators.

To validate the effectiveness of the improved gradient descent method, it is compared with an advanced gradient descent method based on gradient normalization<sup>27</sup>. Table 1; Fig. 5 present the root mean square error (RMS) and trajectory error curves of the two methods in the inverse kinematics solution of a 6-DOF humanoid manipulator.

In Fig. 5a, the trajectory error of the improved gradient descent method remains consistently at the  $10^{-10}$  m order of magnitude, with a fluctuation range smaller than  $3.3799 \times 10^{-10}$  m, demonstrating excellent convergence stability and no local optimum trap issue. In Fig. 5b, although the gradient normalization method alleviates gradient oscillation through gradient norm scaling, the absence of a second-order momentum adaptive mechanism leads to error fluctuations reaching the  $10^{-8}$  m order of magnitude.

## Design of the control strategy

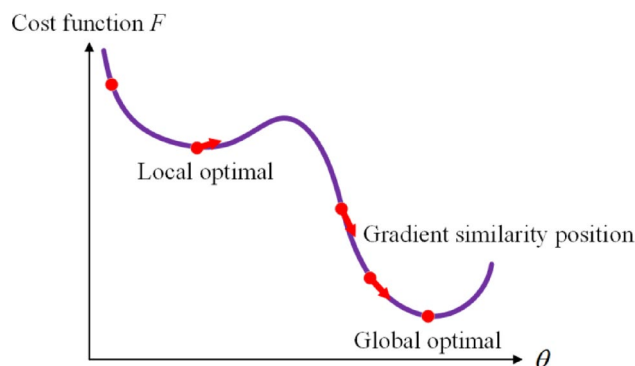
### RBFNN design for SMC

In practical applications, the dynamic model of the humanoid robotic manipulator varies concurrently with payload variations, which inevitably affects the performance of the control strategy. To surmount this challenge, this paper employs the RBFNN<sup>28</sup> to estimate the parameters of the robotic manipulator model, as depicted in Fig. 6. Considering the dynamic model in Eq. (1) and the tracking error equation in Eq. (7), the following sliding variables are defined:

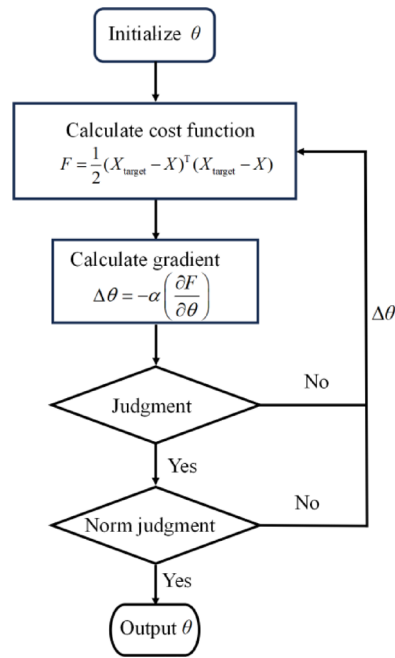
$$r = \dot{e} + \mathbf{A}e \quad (14)$$

where,  $\mathbf{A} = [\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_n]^T \in \mathbb{R}^n$ . Its selection needs to satisfy that  $s^{n-1} + \lambda_{n-1}s^{n-2} + \cdots + \lambda_1$  is Hurwitz polynomial (when  $r \rightarrow 0$ , then  $e \rightarrow 0$ ). From Eq. (14), we obtain:

$$\dot{q} = -r + \dot{q}_d + \mathbf{A}e \quad (15)$$



**Fig. 3.** Momentum gradient descent method.



**Fig. 4.** Gradient descent method flowchart.

Solving algorithm	ex(m)	ey(m)	ez(m)
Improved gradient descent method	$5.1768 \times 10^{-10}$	$2.5413 \times 10^{-10}$	$3.3799 \times 10^{-10}$
Advanced gradient descent method	$3.6539 \times 10^{-8}$	$1.8437 \times 10^{-8}$	$2.3838 \times 10^{-8}$

**Table 1.** Comparison table of RMS of position Error.

and

$$\begin{aligned}
 \mathbf{M}\dot{\mathbf{r}} &= \mathbf{M}(\ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} + \Lambda\dot{\mathbf{e}}) \\
 &= \mathbf{M}(\ddot{\mathbf{q}}_d + \Lambda\dot{\mathbf{e}}) - \mathbf{M}\ddot{\mathbf{q}} \\
 &= \mathbf{M}(\ddot{\mathbf{q}}_d + \Lambda\dot{\mathbf{e}}) + \mathbf{C}\dot{\mathbf{q}} + \mathbf{G} + \mathbf{F} + \tau_d - \tau \\
 &= \mathbf{M}(\ddot{\mathbf{q}}_d + \Lambda\dot{\mathbf{e}}) - \mathbf{C}\mathbf{r} + \mathbf{C}(\dot{\mathbf{q}}_d + \Lambda\mathbf{e}) + \mathbf{G} + \mathbf{F} + \tau_d - \tau \\
 &= -\mathbf{C}\mathbf{r} - \tau + f(x) + \tau_d
 \end{aligned} \tag{16}$$

Among them,  $f(x) = \mathbf{M}(\ddot{\mathbf{q}}_d + \Lambda\dot{\mathbf{e}}) + \mathbf{C}(\dot{\mathbf{q}}_d + \Lambda\mathbf{e}) + \mathbf{G} + \mathbf{F}$  is the dynamic model that needs to be approximated,  $\mathbf{M}$ ,  $\mathbf{C}$  and  $\mathbf{G}$  are abbreviations for matrices  $\mathbf{M}(q)$ ,  $\mathbf{C}(q, \dot{q})$ , and  $\mathbf{G}(q)$ , respectively.

In practical applications, the model uncertainty term  $f(x)$  is unknown, so RBFNN needs to approximate  $f(x)$ , and its RBFNN structure is presented in Fig. 6.

In Fig. 7,  $x = [e^T, \dot{e}^T, q_d^T, \dot{q}_d^T, \ddot{q}_d^T]^T$  is the neural network input vector,  $m$ , and  $k$  are the number of input and hidden layer nodes, respectively. The radial basis function in RBFNN uses the Gaussian basis function, expressed as follows:

$$h_j(x) = \exp\left[-\frac{\|x - c_j\|^2}{2b_j^2}\right], j = 1, 2, \dots, k \tag{17}$$

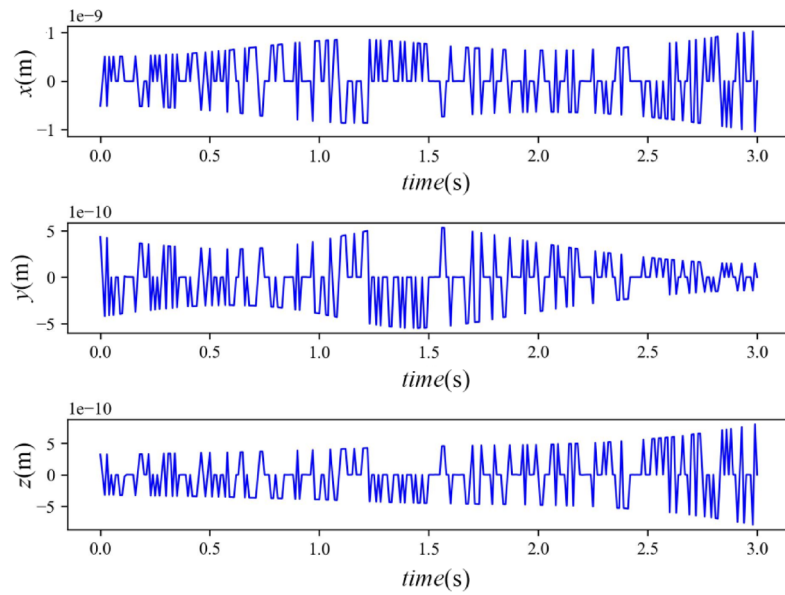
where,  $h_j(x)$  is the nonlinear activation function of the  $j$ th hidden node,  $b_j$  is the basis width of the  $j$ th Gaussian basis function,  $c_j$  is the Gaussian basis function center vector of the  $j$ th node, and  $\|x - c_j\|^2$  is the Euclidean distance between  $x$  and  $c_j$ .

The RBFNN output is defined as follows:

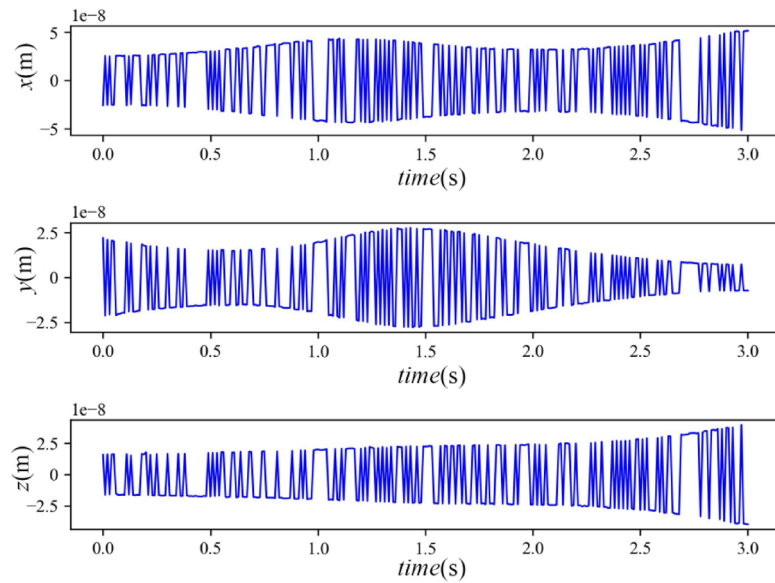
$$\hat{f}(x) = \tilde{\mathbf{W}}^T \varphi(x) \tag{18}$$

where,  $\tilde{\mathbf{W}} = \mathbf{W} - \hat{\mathbf{W}}$ ,  $\|\mathbf{W}\|_F \leq W_{\max}$ .

Then, the designed control law is:



(a) Improved gradient descent method error



(b) Advanced gradient descent method error

**Fig. 5.** Position error comparison analysis. (a) Improved gradient descent method error, (b) advanced gradient descent method error.

$$\tau = \hat{f}(x) + K_v r \quad (19)$$

where,  $\hat{f}(x)$  is the estimated value of  $f(x)$  using RBFNN.

Inserting the control law in Eq. (19) into Eq. (16) provides:

$$\begin{aligned} M\dot{r} &= -Cr - \hat{f} - K_v r + f + \tau_d \\ &= -(K_v + C)r + \tilde{f} + \tau_d \end{aligned} \quad (20)$$

The Lyapunov function is established as:

$$L = \frac{1}{2} r^T M r \quad (21)$$



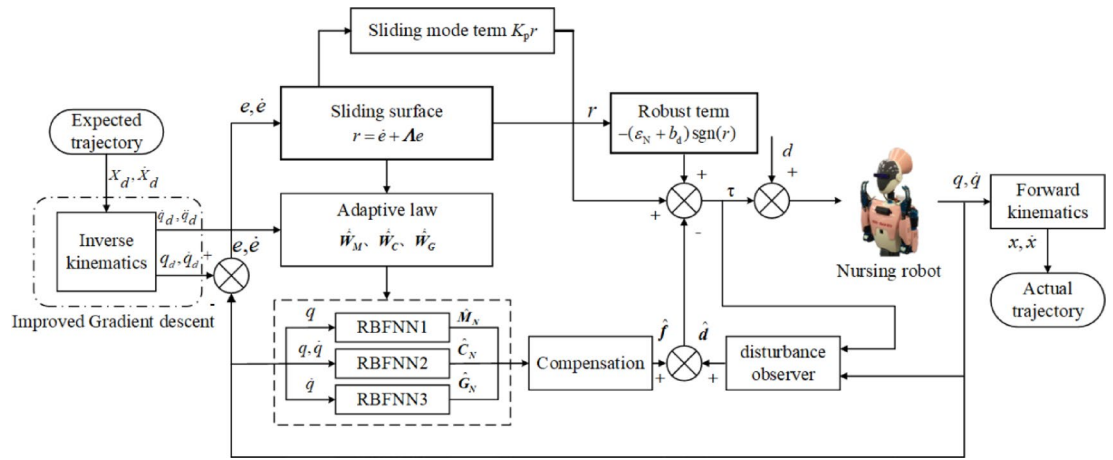


Fig. 6. Overall control strategy structure.

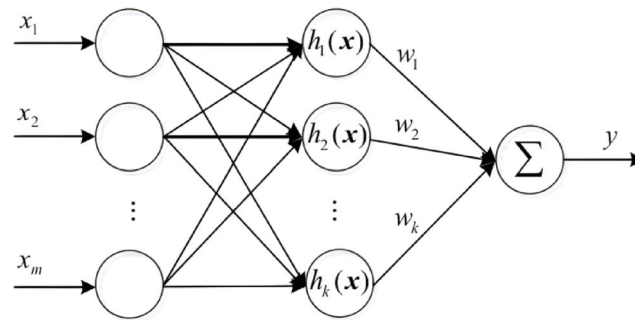


Fig. 7. RBFNN structure diagram.

then,

$$\begin{aligned} \dot{\mathbf{L}} &= \mathbf{r}^T \mathbf{M} \dot{\mathbf{r}} + \frac{1}{2} \mathbf{r}^T \dot{\mathbf{M}} \mathbf{r} \\ &= -\mathbf{r}^T \mathbf{K}_v \mathbf{r} + \frac{1}{2} \mathbf{r}^T (\dot{\mathbf{M}} - 2\mathbf{C}) \mathbf{r} + \mathbf{r}^T (\tilde{\mathbf{f}} + \tau_d) \end{aligned} \quad (22)$$

Therefore, under the condition that  $K_v$  is a constant value, the stability of the control system depends on the approximation accuracy of  $\tilde{\mathbf{f}}$  to  $\mathbf{f}$  and the size of external disturbance  $\tau_d$ . Since the external disturbance  $\tau_d$  is uncertain, the control law must be further improved to ensure the stability of the system.

### RBFNN design for sliding mode robust control

This section proposes a robust sliding mode controller that operates without requiring precise model knowledge. The ideal RBFNN output is defined as:

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}^T \boldsymbol{\varphi}(\mathbf{x}) + \varepsilon \quad (23)$$

where,  $\mathbf{W}^T$  is the weight of the neural network,  $\boldsymbol{\varphi}(\mathbf{x})$  is the Gaussian function.

For the dynamic model in Eq. (1), the control law is designed as:

$$\begin{aligned} \tau &= \hat{\mathbf{f}}(\mathbf{x}) + \mathbf{K}_v \mathbf{r} - \mathbf{v} \\ &= \hat{\mathbf{W}}^T \boldsymbol{\varphi}(\mathbf{x}) + \mathbf{K}_v \mathbf{r} - \mathbf{v} \end{aligned} \quad (24)$$

where,  $\mathbf{v} = -(\varepsilon_N + b_d) \text{sgn}(\mathbf{r})$  is a robust term used to overcome approximation errors in neural networks<sup>29–32</sup>.

Placing the control law of Eq. (24) into Eq. (16), we have:

$$\begin{aligned} \mathbf{M} \dot{\mathbf{r}} &= -(\mathbf{K}_v + \mathbf{C}) \mathbf{r} + \tilde{\mathbf{W}}^T \boldsymbol{\varphi}(\mathbf{x}) + (\varepsilon + \tau_d) + \mathbf{v} \\ &= -(\mathbf{K}_v + \mathbf{C}) \mathbf{r} + \zeta_1 \end{aligned} \quad (25)$$



where,  $\zeta_1 = \tilde{W}^T \varphi(x) + (\varepsilon + \tau_d) + v$ .

Define Lyapunov function:

$$\mathbf{L} = \frac{1}{2} r^T \mathbf{M} r + \frac{1}{2} \text{tr}(\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \tilde{\mathbf{W}}) \quad (26)$$

where,  $\text{tr}(\cdot)$  is the trace of the matrix. The trace of a matrix is the sum of its eigenvalues, by seeking differentiation, we can obtain:

$$\dot{\mathbf{L}} = \dot{r}^T \mathbf{M} \dot{r} + \frac{1}{2} r^T \dot{\mathbf{M}} r + \text{tr}(\tilde{\mathbf{W}}^T \mathbf{F}^{-1} \dot{\tilde{\mathbf{W}}}) \quad (27)$$

By substituting Eq. (25) into the above equation, we can obtain:

$$\dot{\mathbf{L}} = -r^T K_v r + \frac{1}{2} r^T (\dot{\mathbf{M}} - 2\mathbf{C}) r + \text{tr}(\tilde{\mathbf{W}}^T (\mathbf{F}^{-1} \dot{\tilde{\mathbf{W}}} + \varphi r^T)) + r^T (\varepsilon + \tau_d + v) \quad (28)$$

According to Eq. (28) above, in order to make the control system asymptotically stable ( $\dot{V} \leq 0$ ), it is necessary to take:

$$\dot{\tilde{\mathbf{W}}}^T = -\mathbf{F} \varphi r^T \quad (29)$$

The adaptive law for updating weights in RBFNN is:

$$\dot{\tilde{\mathbf{W}}}^T = \mathbf{F} \varphi r^T \quad (30)$$

We can obtain

$$\dot{\mathbf{L}} = -r^T K_v r + r^T (\varepsilon + \tau_d + v) \quad (31)$$

By substituting the robust term  $v = -(\varepsilon_N + b_d) \text{sign}(r)$  into Eq. (31), we can obtain:

$$\dot{\mathbf{L}} = r^T (\varepsilon + \tau_d + v) = r^T (\varepsilon + \tau_d) + r^T v = r^T (\varepsilon + \tau_d) - \|r\| (\varepsilon_N + b_d) \leq 0 \quad (32)$$

Among them,  $\|\varepsilon\| \leq \varepsilon_N, \|\tau_d\| \leq \tau_{dN}$ . Therefore, if  $\dot{V} \leq 0$  can be ensured, the control system will gradually stabilize. Equation (25) infers that the model uncertainty term  $f(x)$  can be described as:

$$f(x) = \mathbf{M}(q) \zeta_1(t) + \mathbf{C}(q, \dot{q}) \zeta_2(t) + \mathbf{G}(q) + \mathbf{F}(\dot{q}) \quad (33)$$

where,  $\xi_1(t) = \ddot{q}_d + \Lambda \dot{e}$ ,  $\xi_2(t) = \dot{q}_d + \Lambda e$ .

RBFNN can be used to approximate and calculate the terms in  $f(x)$  separately:

$$\begin{aligned} \hat{\mathbf{M}}(q) &= \mathbf{W}_M^T \varphi_M(q) \\ \hat{\mathbf{C}}(q, \dot{q}) &= \mathbf{W}_V^T \varphi_V(q, \dot{q}) \\ \hat{\mathbf{G}}(q) &= \mathbf{W}_G^T \varphi_G(q) \\ \hat{\mathbf{F}}(\dot{q}) &= \mathbf{W}_F^T \varphi_F(\dot{q}) \end{aligned} \quad (34)$$

then,

$$\hat{f}(x) = \begin{bmatrix} \mathbf{W}_M^T \xi_1(t) & \mathbf{W}_V^T \xi_2(t) & \mathbf{W}_G^T & \mathbf{W}_F^T \end{bmatrix} \begin{bmatrix} \varphi_M \\ \varphi_V \\ \varphi_G \\ \varphi_F \end{bmatrix} \quad (35)$$

$$\text{where, } \varphi_0(x) = \begin{bmatrix} \varphi_M \\ \varphi_V \\ \varphi_G \\ \varphi_F \end{bmatrix}, \mathbf{W}^T = \begin{bmatrix} \mathbf{W}_M^T & \mathbf{W}_V^T & \mathbf{W}_G^T \end{bmatrix}.$$

The adaptive law is defined as:

$$\begin{aligned} \dot{\hat{\mathbf{W}}}_M &= \mathbf{F}_M \varphi_M \mathbf{r}^T - k_M \mathbf{F}_M \|\mathbf{r}\| \hat{\mathbf{W}}_M \\ \dot{\hat{\mathbf{W}}}_V &= \mathbf{F}_V \varphi_V \mathbf{r}^T - k_V \mathbf{F}_V \|\mathbf{r}\| \hat{\mathbf{W}}_V \\ \dot{\hat{\mathbf{W}}}_G &= \mathbf{F}_G \varphi_G \mathbf{r}^T - k_G \mathbf{F}_G \|\mathbf{r}\| \hat{\mathbf{W}}_G \\ \dot{\hat{\mathbf{W}}}_F &= \mathbf{F}_F \varphi_F \mathbf{r}^T - k_F \mathbf{F}_F \|\mathbf{r}\| \hat{\mathbf{W}}_F \end{aligned} \quad (36)$$

where,  $k_M > 0, k_V > 0, k_G > 0, k_F > 0$ .

Considering the adaptive law, the Lyapunov function is redefined as:

$$\begin{aligned} \mathbf{L} = & \frac{1}{2} \mathbf{r}^T \mathbf{M} \mathbf{r} + \frac{1}{2} \text{tr}(\tilde{\mathbf{W}}_M^T F_M^{-1} \tilde{\mathbf{W}}_M) + \frac{1}{2} \text{tr}(\tilde{\mathbf{W}}_V^T F_V^{-1} \tilde{\mathbf{W}}_V) \\ & + \frac{1}{2} \text{tr}(\tilde{\mathbf{W}}_G^T F_G^{-1} \tilde{\mathbf{W}}_G) \end{aligned} \quad (37)$$

then,

$$\begin{aligned} \dot{\mathbf{L}} = & \mathbf{r}^T \dot{\mathbf{M}} \mathbf{r} + \frac{1}{2} \mathbf{r}^T \dot{\mathbf{M}} \mathbf{r} + \text{tr}(\tilde{\mathbf{W}}_M^T F_M^{-1} \dot{\tilde{\mathbf{W}}}_M) + \text{tr}(\tilde{\mathbf{W}}_V^T F_V^{-1} \dot{\tilde{\mathbf{W}}}_V) \\ & + \text{tr}(\tilde{\mathbf{W}}_G^T F_G^{-1} \dot{\tilde{\mathbf{W}}}_G) + \text{tr}(\tilde{\mathbf{W}}_F^T F_F^{-1} \dot{\tilde{\mathbf{W}}}_F) \end{aligned} \quad (38)$$

Placing the control law of Eq. (24) in Eq. (38) provides:

$$\begin{aligned} \dot{\mathbf{L}} = & -\mathbf{r}^T K_v \mathbf{r} + \frac{1}{2} \mathbf{r}^T (\dot{\mathbf{M}} - 2\mathbf{C}) \mathbf{r} + \mathbf{r}^T (\varepsilon + \tau_d) + \mathbf{r}^T \mathbf{v} \\ & + \text{tr} \tilde{\mathbf{W}}_M^T (F_M^{-1} \dot{\tilde{\mathbf{W}}}_M + \varphi_M \mathbf{r}^T) + \text{tr} \tilde{\mathbf{W}}_V^T (F_V^{-1} \dot{\tilde{\mathbf{W}}}_V + \varphi_V \mathbf{r}^T) \\ & + \text{tr} \tilde{\mathbf{W}}_G^T (F_G^{-1} \dot{\tilde{\mathbf{W}}}_G + \varphi_G \mathbf{r}^T) + \text{tr} \tilde{\mathbf{W}}_F^T (F_F^{-1} \dot{\tilde{\mathbf{W}}}_F + \varphi_F \mathbf{r}^T) \end{aligned} \quad (39)$$

Considering the characteristics of the humanoid robotic manipulator and substituting the adaptive law of Eq. (36) into Eq. (39), we get:

$$\begin{aligned} \dot{\mathbf{L}} = & -\mathbf{r}^T K_v \mathbf{r} + \mathbf{r}^T (\varepsilon + \tau_d) + \mathbf{r}^T \mathbf{v} \\ & + k_M \|\mathbf{r}\| \text{tr} \tilde{\mathbf{W}}_M^T (\mathbf{W}_M - \tilde{\mathbf{W}}_M) \\ & + k_V \|\mathbf{r}\| \text{tr} \tilde{\mathbf{W}}_V^T (\mathbf{W}_V - \tilde{\mathbf{W}}_V) \\ & + k_G \|\mathbf{r}\| \text{tr} \tilde{\mathbf{W}}_G^T (\mathbf{W}_G - \tilde{\mathbf{W}}_G) \\ & + k_F \|\mathbf{r}\| \text{tr} \tilde{\mathbf{W}}_F^T (\mathbf{W}_F - \tilde{\mathbf{W}}_F) \end{aligned} \quad (40)$$

Because of

$$\text{tr} \tilde{\mathbf{W}}^T (\mathbf{W} - \tilde{\mathbf{W}}) = (\tilde{\mathbf{W}}, \mathbf{W})_F - \|\tilde{\mathbf{W}}\|_F^2 \leq \|\tilde{\mathbf{W}}\|_F \|\mathbf{W}\|_F - \|\tilde{\mathbf{W}}\|_F^2 \quad (41)$$

Considering the robust term  $v = -(\varepsilon_N + b_d) \text{sign}(s)$ , we get:

$$\begin{aligned} \dot{\mathbf{L}} \leq & -K_{v\min} \|\mathbf{r}\|^2 + k_M \|\mathbf{r}\| \|\tilde{\mathbf{W}}_M\|_F (\mathbf{W}_{M\max} - \|\tilde{\mathbf{W}}_M\|_F) \\ & + k_V \|\mathbf{r}\| \|\tilde{\mathbf{W}}_V\|_F (\mathbf{W}_{V\max} - \|\tilde{\mathbf{W}}_V\|_F) \\ & + k_G \|\mathbf{r}\| \|\tilde{\mathbf{W}}_G\|_F (\mathbf{W}_{G\max} - \|\tilde{\mathbf{W}}_G\|_F) \\ = & -\|\mathbf{r}\| [K_{v\min} \|\mathbf{r}\| + k_M \|\tilde{\mathbf{W}}_M\|_F (\|\tilde{\mathbf{W}}_M\|_F - \mathbf{W}_{M\max}) \\ & + k_V \|\tilde{\mathbf{W}}_V\|_F (\|\tilde{\mathbf{W}}_V\|_F - \mathbf{W}_{V\max}) \\ & + k_G \|\tilde{\mathbf{W}}_G\|_F (\|\tilde{\mathbf{W}}_G\|_F - \mathbf{W}_{G\max}) \\ & + k_F \|\tilde{\mathbf{W}}_F\|_F (\|\tilde{\mathbf{W}}_F\|_F - \mathbf{W}_{F\max})] \end{aligned} \quad (42)$$

Because of

$$\begin{aligned} & k \|\tilde{\mathbf{W}}\|_F (\|\tilde{\mathbf{W}}\|_F - \mathbf{W}_{\max}) \\ = & k \|\tilde{\mathbf{W}}\|_F (\|\tilde{\mathbf{W}}\|_F - \mathbf{W}_{\max}/2)^2 - k \mathbf{W}_{\max}^2/4 \end{aligned} \quad (43)$$

For  $\dot{\mathbf{L}} < 0$ , we need

$$\|\mathbf{r}\| > \frac{k_M \mathbf{W}_{M\max}^2/4 + k_V \mathbf{W}_{V\max}^2/4 + k_G \mathbf{W}_{G\max}^2/4 + k_F \mathbf{W}_{F\max}^2/4}{K_{v\min}} \quad (44)$$

or  $\|\tilde{\mathbf{W}}_M\|_F > \mathbf{W}_{M\max}$ ,  $\|\tilde{\mathbf{W}}_V\|_F > \mathbf{W}_{V\max}$  and  $\|\tilde{\mathbf{W}}_G\|_F > \mathbf{W}_{G\max}$ .

When faced with unknown dynamics and compound model uncertainties, RBFNN is employed to approximate uncertain system dynamics. Leveraging SMC principles, this paper proposes a robust controller that operates independently of precise model information. The SMC law is designed to compensate for both approximation errors and external disturbances through adaptive control gains, ensuring closed-loop stability under Lyapunov analysis.

### Nonlinear disturbance observer

The control system of the Seed Noid R7F humanoid robotic manipulator must consider unknown external disturbances and the effect of joint friction, which, according to Eq. (5), are regarded as compound disturbances. By using a NDO<sup>4</sup> to estimate compound disturbances, the calculated values are fed back to the input to adjust the control torque, reducing the impact of compound disturbances on the humanoid robotic manipulator control system and demonstrating anti-disturbance solid performance. The kinetic equation for the compound disturbance  $\rho$  is described as follows:

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) = \tau + \rho \quad (45)$$

According to Assumption 2 in Sect. 2, the compound disturbance Eq. (5) is bounded, and thus, the NDO equation is written as:

$$\begin{aligned} \dot{\hat{\rho}} &= \mathbf{L}(q, \dot{q}) (\rho - \hat{\rho}) \\ &= \mathbf{L}(q, \dot{q}) [\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) - \tau] - \mathbf{L}(q, \dot{q})\hat{\rho} \end{aligned} \quad (46)$$

where,  $\hat{\rho}$  is the estimated compound disturbance from the NDO and  $\mathbf{L}(q, \dot{q})$  is the gain matrix. Since it is challenging to obtain the joint angular acceleration signal in practical applications, this signal should not be used when designing a NDO and the auxiliary parameter vector must be defined as:

$$z = \hat{\rho} - p(q, \dot{q}) \quad (47)$$

where,  $p(q, \dot{q})$  is the function vector that must be designed. Then,  $p(q, \dot{q})$  and  $\mathbf{L}(q, \dot{q})$  must satisfy the following conditions:

$$\frac{dp(q, \dot{q})}{dt} = \mathbf{L}(q, \dot{q}) \mathbf{M}(q)\ddot{q} \quad (48)$$

Since  $\mathbf{M}(q)$  is a positive definite matrix, the matrix  $\mathbf{L}(q, \dot{q})$  can be made positive definite by designing an appropriate  $p(q, \dot{q})$ . Taking the derivative of  $z$  provides:

$$\begin{aligned} \dot{z} &= \dot{\hat{\rho}} - \dot{p}(q, \dot{q}) \\ &= \mathbf{L}(q, \dot{q}) [\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) - \tau] \\ &\quad - \mathbf{L}(q, \dot{q})\hat{\rho} - \mathbf{L}(q, \dot{q})\mathbf{M}(q)\ddot{q} \\ &= \mathbf{L}(q, \dot{q}) [\mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) - \tau - p(q, \dot{q})] - \mathbf{L}(q, \dot{q})z \end{aligned} \quad (49)$$

To sum up, the NDO is written as:

$$\begin{aligned} \dot{z} &= \mathbf{L}(q, \dot{q}) [\mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) - \tau - p(q, \dot{q})] - \mathbf{L}(q, \dot{q})z \\ \dot{\hat{\rho}} &= z + p(q, \dot{q}) \end{aligned} \quad (50)$$

then, the general control equation of the humanoid robotic manipulator system can be written as follows:

$$\tau = \hat{f}(x) + K_v r - \nu - \hat{\rho} \quad (51)$$

Let the estimation error of the NDO be  $e = \rho - \hat{\rho}$ . Assuming that the change of the compound disturbance is slow, i.e.,  $\dot{\rho} = 0$ , we have:

$$\dot{e}_\rho = \dot{\rho} - \dot{\hat{\rho}} = -\dot{\hat{\rho}} = -\mathbf{L}(q, \dot{q})e_\rho \quad (52)$$

The Lyapunov function is defined as  $V_\rho = \frac{1}{2}e_\rho^T e_\rho$ , and thus:

$$\dot{V}_\rho = e_\rho^T \dot{e} = -e_\rho^T \mathbf{L}(q, \dot{q})e_\rho \quad (53)$$

Since  $\mathbf{L}(q, \dot{q})$  is a positive definite matrix,  $\dot{V}_\rho \leq 0$ , and the observation error of the NDO converges asymptotically.

A NDO is designed to estimate and attenuate unknown external disturbances in the dynamic model of a nursing robot, specifically addressing nonlinear joint friction effects. By integrating the NDO output with the SMC input, a closed-loop control architecture is constructed to compensate for both parametric uncertainties and external perturbations, thereby ensuring robust tracking performance under Lyapunov stability theory.

This study presents an integrated control framework combining RBFNN, sliding-mode robust control, and NDO. The RBFNN provides global approximation of parametric uncertainties for adaptive parameter estimation, while sliding-mode control ensures robustness through sliding variable design. The NDO compensates for external disturbances and unmodeled dynamics, guaranteeing closed-loop stability under Lyapunov analysis. This synergetic architecture effectively addresses dynamic variations in 6-DOF humanoid manipulators operating in complex environments, offering a robust control solution for trajectory tracking applications.

J	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	0°	0	0.05978	$\theta_1$
2	− 90°	0.028046	0	$\theta_2$
3	− 90°	0.01	0.2895	$\theta_3$
4	− 90°	− 0.035	0	$\theta_4$
5	90°	0.035	0.2655	$\theta_5$
6	− 90°	0	0	$\theta_6$

**Table 2.** D-H parameter table for seed noid R7F.

J	Mass (kg)	Inertia (kg•m <sup>2</sup> )	Length (m)
1	0.242	0.000263	0.05978
2	0.744	0.000623	0.028046
3	0.574	0.000214	0.2895
4	0.711	0.000178	0.035
5	0.132	0.0007	0.2655
6	0.005	0.0001	0

**Table 3.** Physical parameters of seed noid R7F.

Simulation and result analysis

An evaluation of the proposed sliding-mode robust control strategy, which integrates a RBFNN with global approximation capabilities and a NDO, is conducted on a 6-DOF humanoid robotic manipulator, specifically the Seed Noid R7F nursing service robot. This robotic manipulator consists of six rotating joints, offering a greater number of joint configurations during movement, thereby enhancing its flexibility and trajectory diversity.

In this study, the right arm of the robot is selected as the modeling object. The D-H parameters are presented in Table 2, and the comprehensive physical parameters of the humanoid robotic manipulator are listed in Table 3.

The control strategy for the humanoid robotic manipulator system is developed in MATLAB/Simulink. Spiral trajectories are used as the reference paths for the robot in Cartesian space. The pose angle of the robot is represented by Euler angles following the Z-Y-X sequence, which correspond to the rotation angles about the Z, Y, and X axes, respectively.

The spiral lines are  $x_d = 0.02\cos(\pi t) + 0.1$ ,  $y_d = 0.02\sin(\pi t) + 0.1$ ,  $z_d = 0.05t + 0.1$ . The pose angle is a constant value, where roll, pitch, and yaw are 45°, 60°, and 30°, respectively. The sliding mode surface parameter is  $\Lambda = 15$ . The basis function center is  $c = [-1 - 0.8 - 0.6 - 0.4 - 0.2 \ 0 \ 0.2 \ 0.4 \ 0.6 \ 0.8 \ 1]$  and the basis function width is  $b = 20$ . For the weight adaptive law coefficient matrix,  $F_M$ ,  $F_C$ , and  $F_G$  are assigned a value of 100. In response to the influence of unknown external disturbances and joint friction in the control system of the 6-DOF humanoid robotic manipulator, this study considers the unknown external disturbances and joint friction as composite disturbances. Simulated experiment applying composite interference. Joint friction is the sum of Coulomb friction and viscous friction  $F(\dot{q}) = 2\text{sign}(\dot{q}) + 0.8\dot{q}$ . The external disturbance is given to six joints  $\tau_d = 0.8\sin(t)$ , respectively.

The robust parameters of this article are selected as 0.5, 1.5, 2.1, and 3, and the position error comparison chart is shown in Fig. 8.

The number of hidden layer neurons in RBFNN directly impacts dynamic model robustness. We evaluated 5, 8, 11, and 14 hidden units for robustness analysis, with results presented in Fig. 9.

The design parameters of the nonlinear disturbance observer are  $p(q, \dot{q}) = cc[\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]^T$  and  $L(q, \dot{q}) = ccM^{-1}$ , where  $cc$  is selected as 10, 20, 60, 100, and 120. The disturbance errors under different parameters are compared, as shown in Fig. 10.

Finally, the design parameters of the NDO are:  $p(q, \dot{q}) = 60[\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]^T$  and  $L(q, \dot{q}) = 60M^{-1}$ . The RBFNN involves 11 hidden layer nodes.  $\varepsilon_N = 2$  and  $b_d = 0.1$  in robust term.

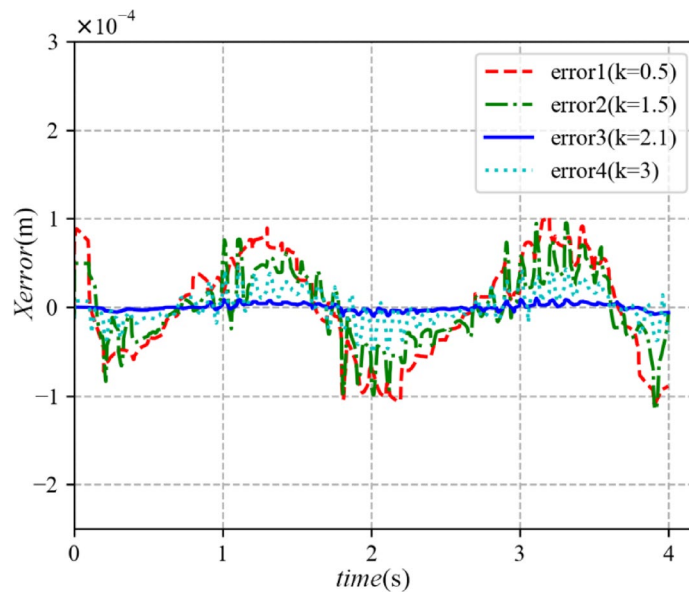
The inverse kinematics solution maps Cartesian space coordinates to joint space, generating desired joint angle trajectories. The proposed control strategy is then applied to achieve trajectory tracking in joint space. Finally, forward kinematics computations reconstruct the Cartesian space trajectory from the tracked joint angles.

Digital simulation results

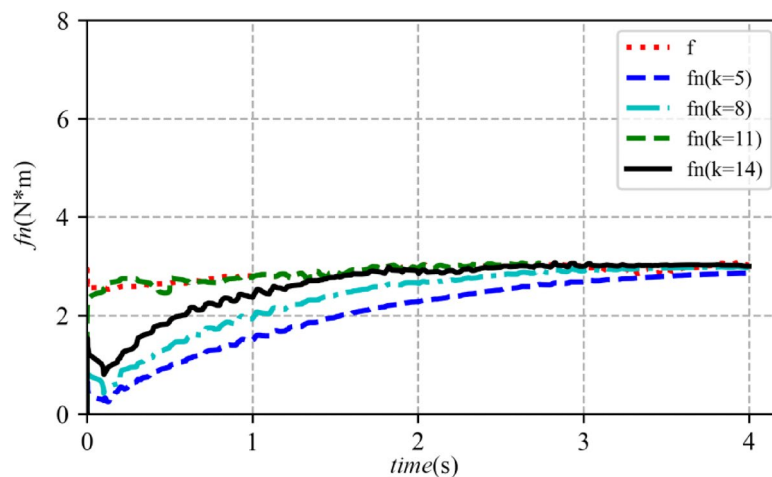
Figures 11 and 12 present Cartesian space trajectory tracking results of the proposed RBFNN- SMC strategy, demonstrating high-precision convergence between actual and desired trajectories.

The inverse kinematics solution maps Cartesian space trajectories to joint space, generating expected joint angle trajectories. Figure 13 presents attitude angle results in Cartesian space, while Figs. 14 and 15 illustrate joint position and velocity profiles in joint space. These results demonstrate that the proposed control strategy achieves high-precision tracking of spiral trajectories, validating its reliability for complex trajectory tasks.

The fitting results of RBFNN based on the overall approximation of the model are illustrated in Fig. 16.



**Fig. 8.** Comparison of position Error results under different robust parameter.



**Fig. 9.** Comparison of dynamic model fitting results with different numbers of hidden neurons.

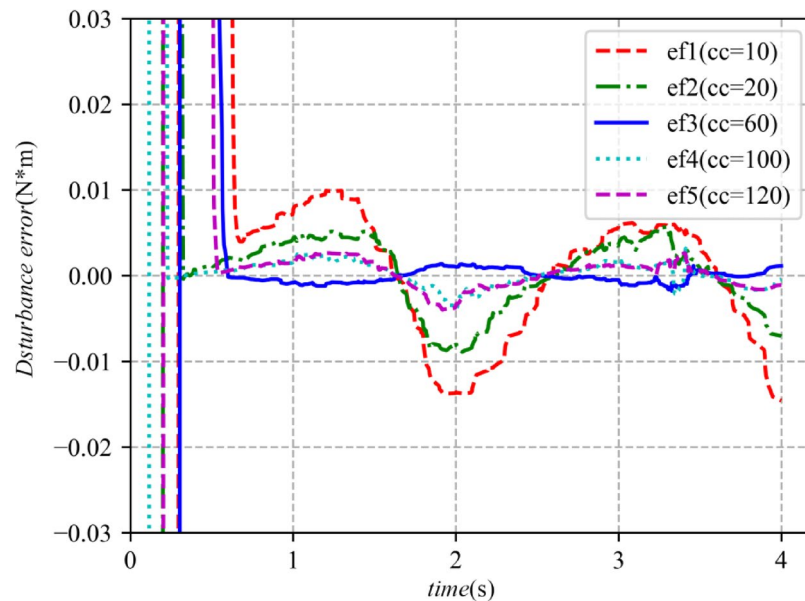
Take the uncertain terms of the model as:  $\bar{M} = M + 0.5M$ ;  $\bar{C} = C + 0.5C$ ;  $\bar{G} = G + 0.5G$ . With the introduction of model uncertainty, the RBFNN-based approximation converges to the actual dynamics, demonstrating robust control performance under parametric variations. Figure 17 compares the dynamic model fitting results before and after uncertainty injection, verifying the effectiveness of the RBFNN in approximating uncertain system dynamics.

Figure 18 shows the joint torque diagram, indicating that after effectively handling the uncertain influencing factors of the control system, the torque jitter phenomenon is significantly reduced.

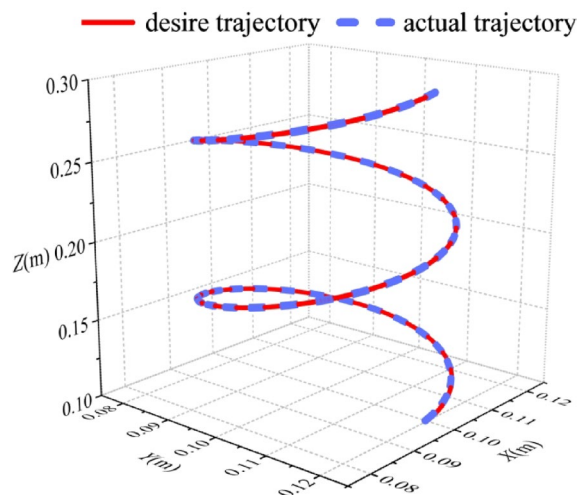
The composite interference results are estimated using a disturbance observer, as depicted in Fig. 19. It reveals that after the response, the matrix norm of the fitted model  $f_N(x)$  is almost identical to the actual model  $f(x)$ , and the error is within an acceptable range.

At  $t = 2s$ , a step interference signal with amplitude of 10 N is applied to the end effector. As depicted in Fig. 20, the NDO demonstrates the ability to precisely estimate step interference signals.

RBFNN have demonstrated superior performance in dynamic modeling applications. Specifically, RBFNN exhibits robust fitting capabilities for nonlinear system dynamics, accurately capturing complex model characteristics. Under parametric uncertainty, RBFNN-based approximations asymptotically converge to true system dynamics, preserving control robustness and ensuring stable operation in uncertain environments. When subjected to composite disturbances, the RBFNN-identified model maintains close approximation to actual dynamics, showcasing excellent disturbance rejection properties. Furthermore, the proposed framework



**Fig. 10.** Comparison of interference error fitting results with different interference observer parameters.



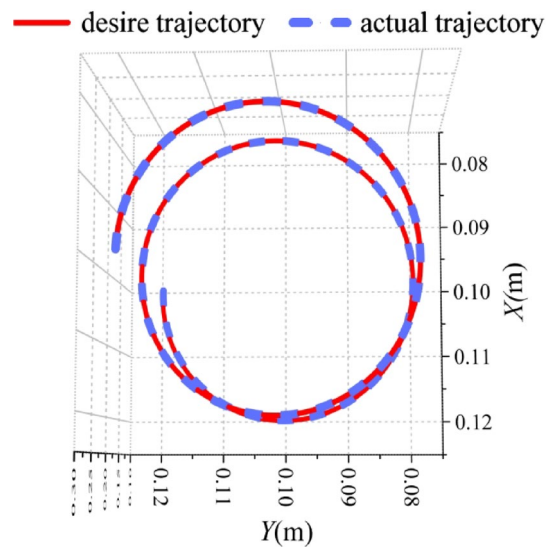
**Fig. 11.** Cartesian space high precision trajectory tracking results.

demonstrates favorable control performance under step disturbances, effectively mitigating perturbations and validating system robustness.

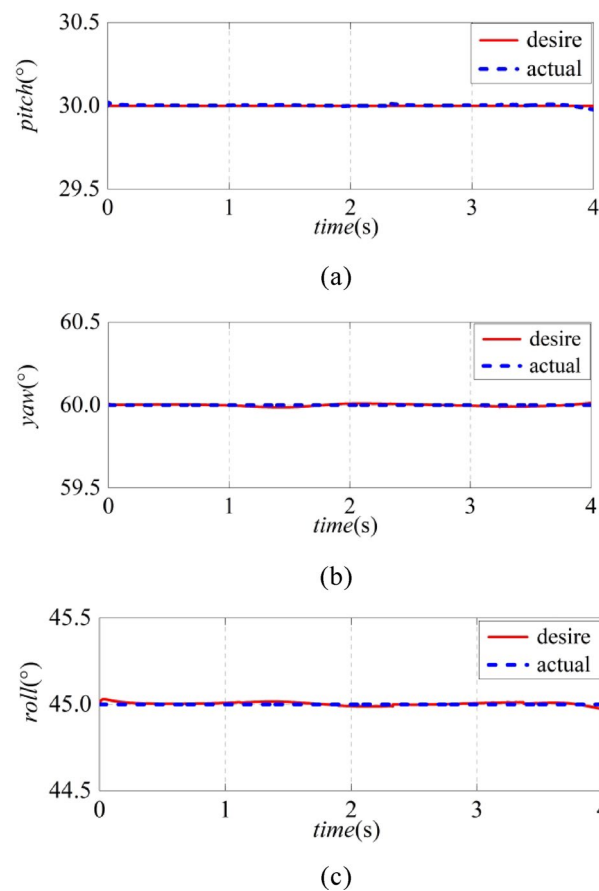
### Comparative analysis of the results

Comparative analysis between the proposed control framework and alternative strategies is presented in Figs. 21 and 22. For the 6-DOF humanoid manipulator trajectory tracking task, Cartesian space position errors are compared between actual and target trajectories. Quantitative results demonstrate statistically significant improvements in tracking performance through adaptive parameter tuning. Specifically, Figs. 21 and 22 reveal that the RBF + NDO and FA-SMC strategies outperform other methods in both control accuracy and disturbance rejection. While FA-SMC exhibits larger Z-axis errors at initial transient phases, PD + NDO demonstrates lower tracking accuracy, and SMC + NDO experiences error oscillations under uncertainty, indicating reduced robustness. Key performance metrics are summarized in Table 4, including RMSE and maximum deviation values.

$$E = \frac{\sum e_i^2}{n} \quad (54)$$



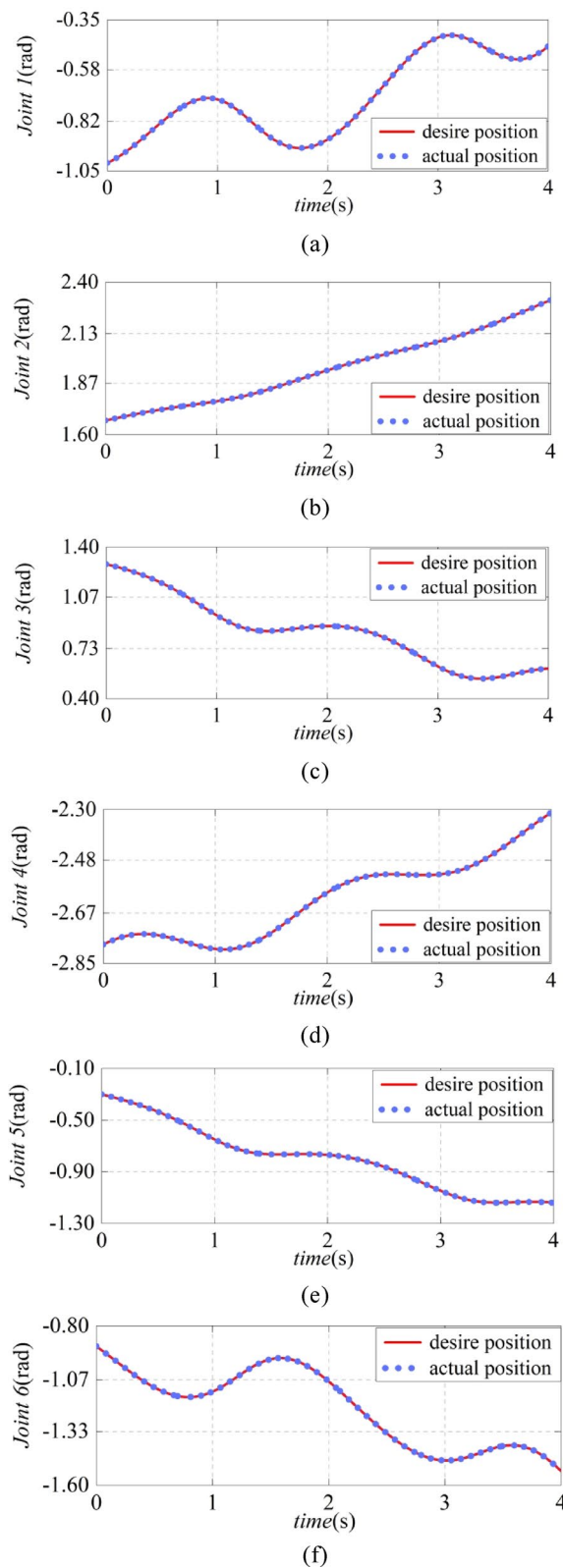
**Fig. 12.** Cartesian space high precision trajectory tracking results.



**Fig. 13.** Attitude angle tracking.

As manifested in Table 4, the devised sliding mode robust control strategy, which employs the RBFNN predicated on model global approximation and a NDO, demonstrates excellent performance in the trajectory tracking control of the humanoid robotic manipulator. In contrast to conventional control strategies, this control approach attains enhanced anti-interference capabilities and superior control accuracy.

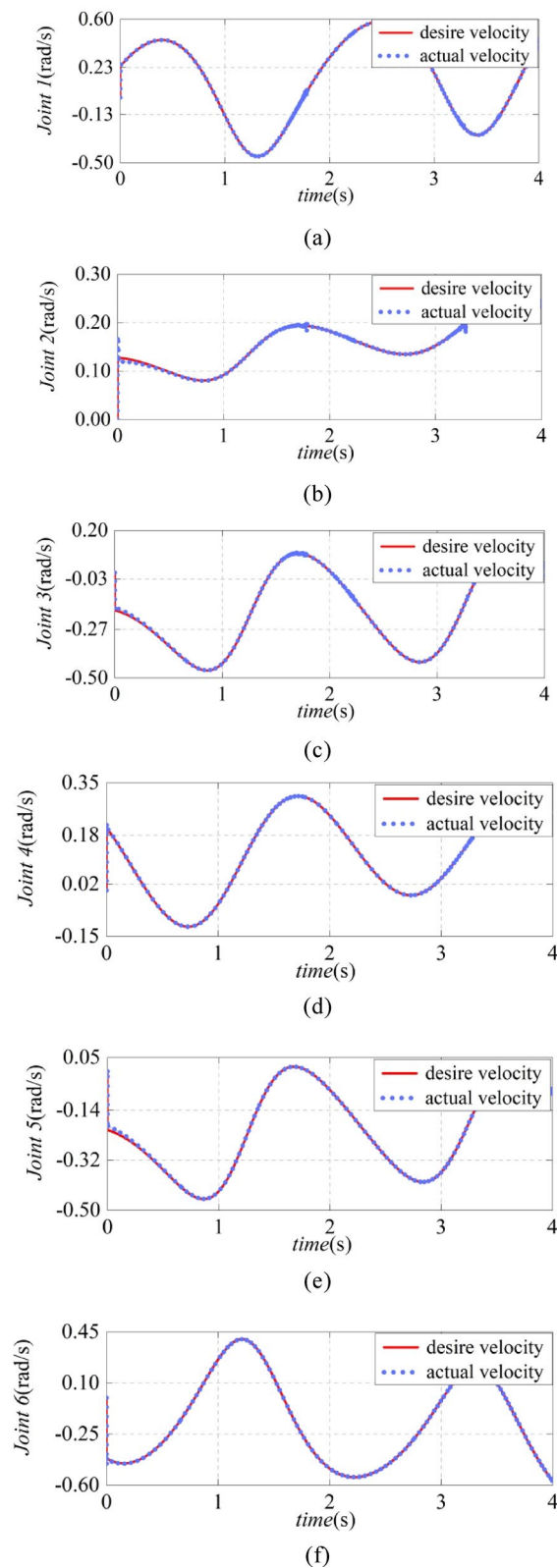




**Fig. 14.** Joint position tracking.

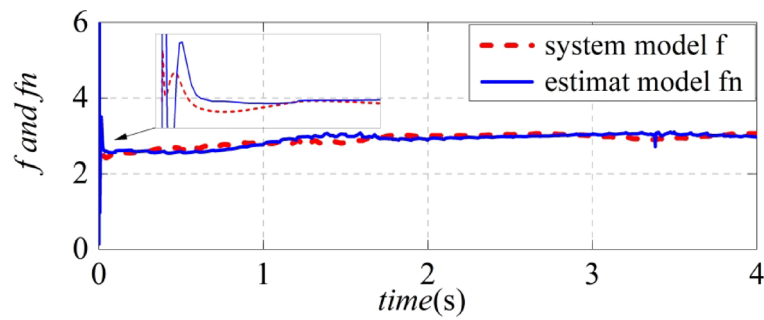
### Virtual simulation results

The Linux/Ubuntu20.04 system uses the Noetic version of the ROS simulation platform, which is combined with the Moveit library for the experiments. First, one must obtain the URDF file officially provided by the robot and then import the Seed Noid R7F humanoid robotic manipulator into the system for trajectory tracking

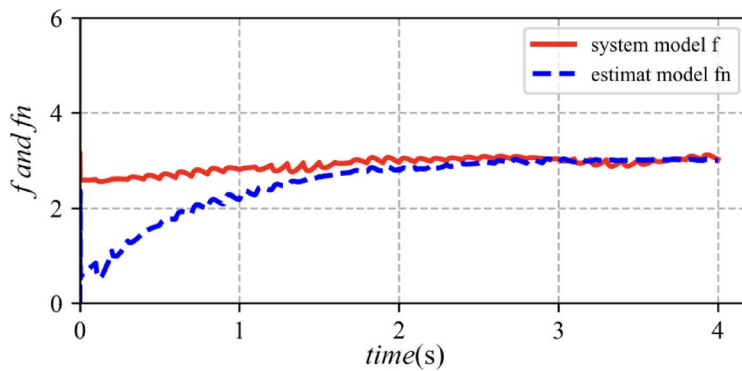


**Fig. 15.** Joint velocity tracking.

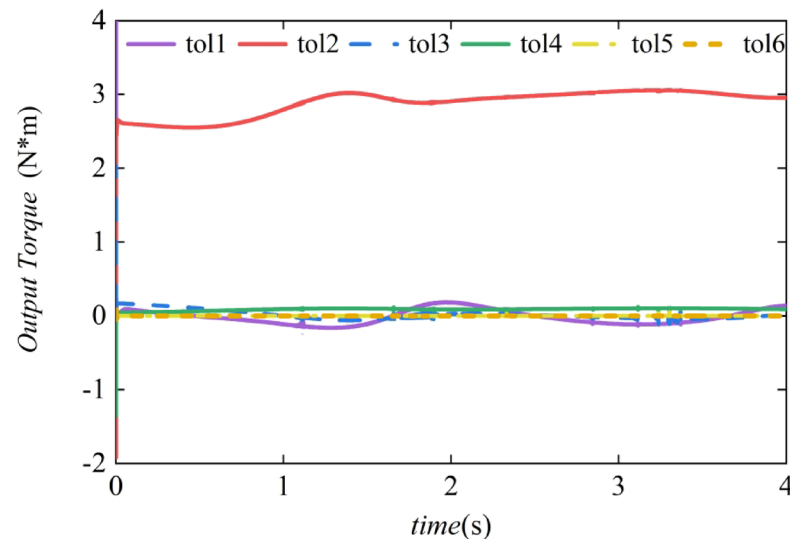
experiments. The path trajectories are automatically generated by Moveit, and the tracking experiments are carried out through the employment of the robust RBFNN-SMC founded on global approximation. With the initial position of the box being (0, 3, -0.2, 0, 0.775), the robotic manipulator is set to transition from its initial state to the initial position of the crate and subsequently to the target position of the box, which is (0, 5, -0.4, 1,



**Fig. 16.** Figure of fitting results of the dynamic model.



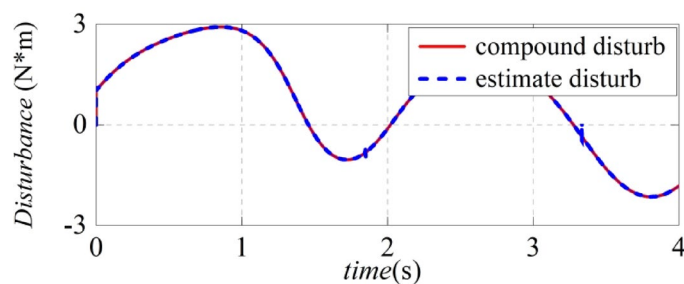
**Fig. 17.** Figure of fitting results of the dynamic model.



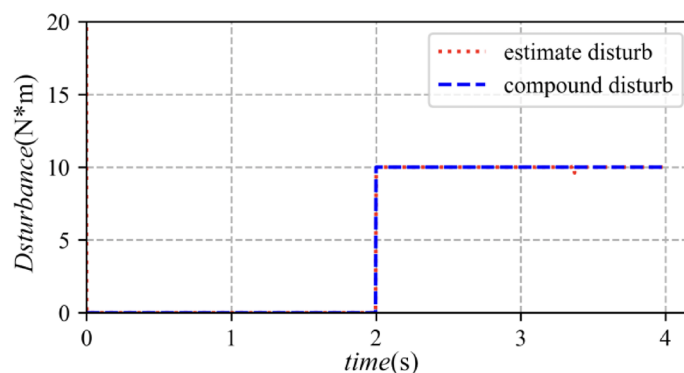
**Fig. 18.** Controller output torque.

1.055). The manner in which the robotic manipulator is moved from the initial state to the initial position of the box is illustrated in Fig. 23. The movement state of the trajectory of the robotic manipulator from the initial point of the box to its target point is portrayed in Fig. 24. The final state of the robotic manipulator is presented in Fig. 25.

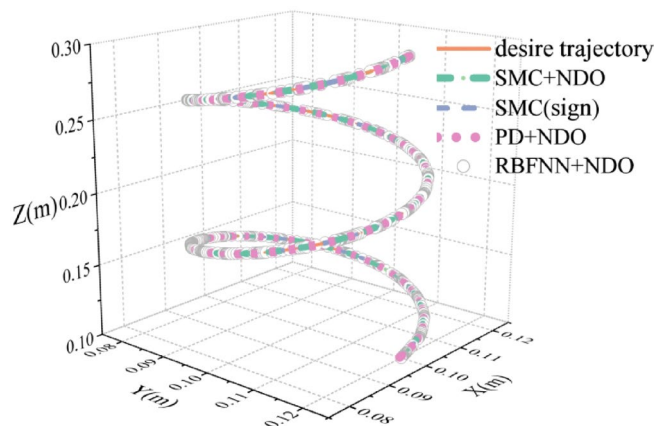
Figures 24 and 25 demonstrate smooth, stable, and precise target object grasping at desired positions using the proposed RBFNN-SMC strategy. Figures 26, 27 and 28, and 29 reveal that the robot can smoothly, stably, and accurately grasp the target object at the desired position in the ROS simulation platform using the proposed control strategy. During experiments, extensive sensor data were collected for key parameters including joint



**Fig. 19.** Composite interference and estimation.



**Fig. 20.** Composite interference and estimation.



**Fig. 21.** Control strategy comparison.

angles, accelerations, torques, and end-effector trajectory states. Data analysis reveals that the control strategy maintains excellent tracking performance and robust disturbance rejection under compound uncertainties (model uncertainty, external unknown disturbances, and joint friction), ensuring closed-loop stability and precision control.

### Experimental process and result analysis

The experiment uses the Seed Noid R7F robot manipulator with six rotary joints. The maximum load capacity of the manipulator is 2 kg, the loaded manipulator contains three fingers, and the maximum holding force is 150 N, capable of performing various complex movements. The robot is controlled in real-time on the PC side of the ROS operating system.

The experimental platform uses the KDL (Kinematics and Dynamics Library) solver. The target object is a box of milk. Upon the generation of the trajectory via KDL, the trajectory tracking experiment is carried out by employing the sliding mode robust control strategy, which incorporates the RBFNN founded on the proposed global approximation and NDO.

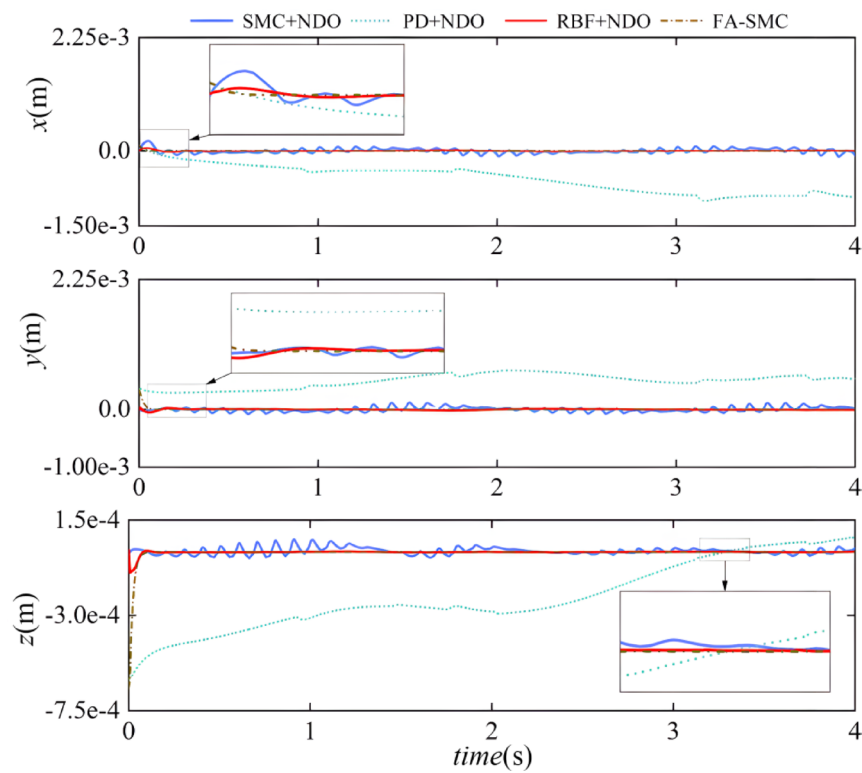


Fig. 22. Position error comparison chart.

Controller	x(m)	y(m)	z(m)
RBF + NDO	$5.01 \times 10^{-5}$	$4.45 \times 10^{-5}$	$1.39 \times 10^{-5}$
SMC + NDO	$1.71 \times 10^{-4}$	$2.45 \times 10^{-4}$	$2.38 \times 10^{-5}$
SMC(sign)	$2.46 \times 10^{-5}$	$3.88 \times 10^{-5}$	$1.93 \times 10^{-4}$
PD + NDO	$2.84 \times 10^{-4}$	$1.18 \times 10^{-4}$	$1.71 \times 10^{-4}$

Table 4. Mean square deviation comparison. Subsequently, by computing the mean square error (MSE), a quantitative metric is established to further assess the performance of the tracking control strategy. The MSE is defined as follows:

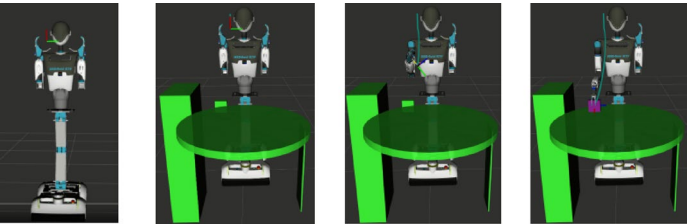
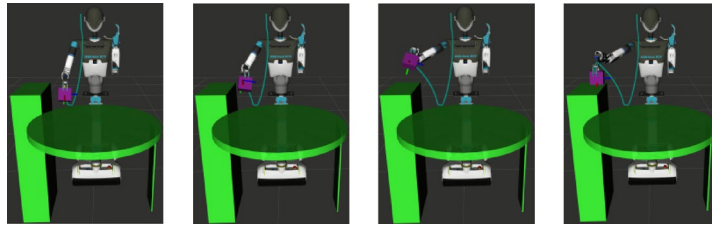
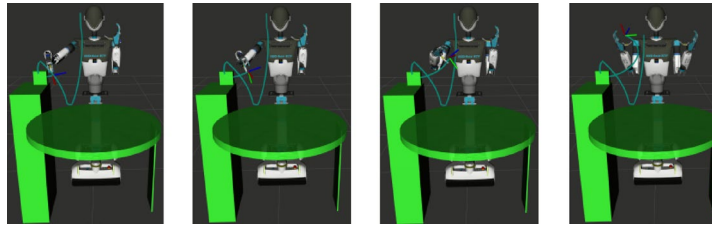


Fig. 23. Initial state of the Seed Noid R7F robot.

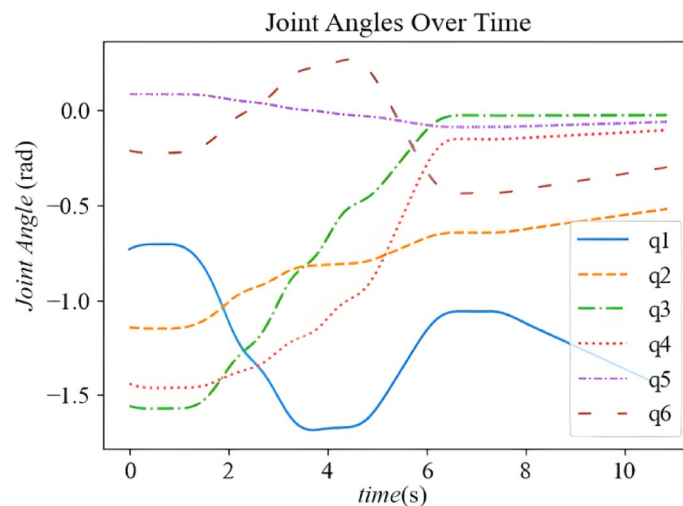
Figure 30 presents the experimental status of the Seed Noid R7F robot. In the phases illustrated by (a)-(d), the robot executes a displacement from the initial pose to the pose of the box. Subsequently, during the intervals depicted in (e)-(h), the robot undertakes the task of grasping the box and transporting it to the target pose. It is a long process for the Seed Noid R7F robot to perform the above gripping tasks and requires much computing time. Therefore, due to the substantial computational overhead of the trajectory tracking and the complex trajectories, the execution time is extended. Nevertheless, the experimental outcomes indicate that the Seed Noid R7F robot is capable of accomplishing the grasping action successfully with the employment of the proposed controller. Although friction interference and hardware circuit response delay the grasping process of



**Fig. 24.** Seed Noid R7F robot motion status.



**Fig. 25.** Seed Noid R7F robot final state.

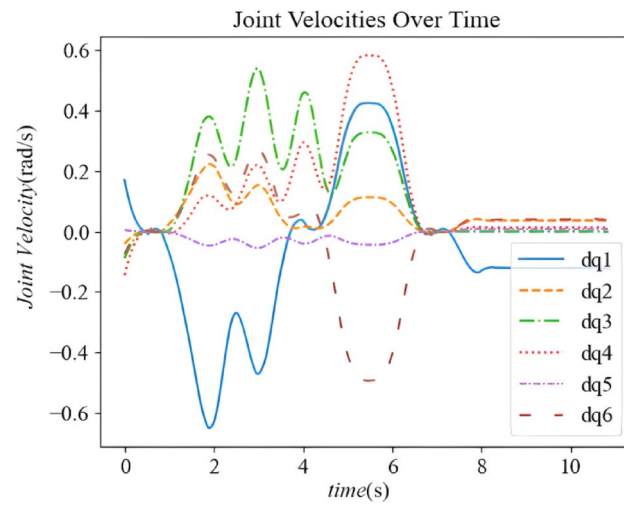


**Fig. 26.** Six joint angles of the robot.

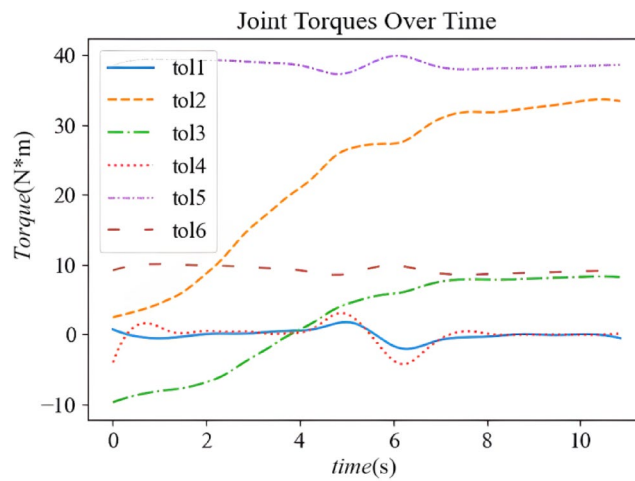
the robotic manipulator, the grasping trajectory action is accurate, as illustrated in Fig. 31. This control strategy has good control performance and strong anti-interference ability.

## Conclusion

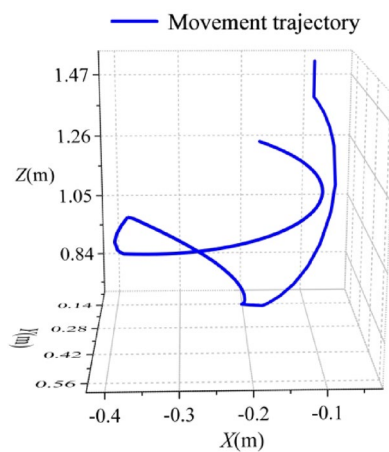
The present paper tackles the problems concerning model uncertainty, joint friction, and unknown external disturbances that are associated with the 6-DOF humanoid robotic manipulator. Specifically, this study proposes a sliding mode robust control strategy with the RBFNN based on model global approximation and NDO by using the RBFNN to approximate the whole dynamic model and applying the weight adaptive law to adjust the dynamic model parameters online. The developed strategy can effectively reconstruct the dynamic model. Additionally, the NDO conducts real-time monitoring of the unknown external disturbances and joint friction, and further compensates within the control strategy to mitigate the influence exerted by these factors. Simulation and experimental results in MATLAB/Simulink and ROS environments demonstrate that compared with control approaches such as SMC+NDO and PD+NDO, the proposed controller exhibits remarkably smaller fluctuations across all axes, maintaining closer adherence to the ideal trajectory. Mean square deviation data further reveals that the RBF + NDO controller achieves the minimum values for the x, y, and z axes  $-5.01 \times 10^{-5}$  m,  $4.45 \times 10^{-5}$  m, and  $1.39 \times 10^{-5}$  m, respectively. Notably, the x-axis value is merely 29.3% of the SMC+NDO method ( $1.71 \times 10^{-5}$  m). These quantitative comparisons and trajectory analyses collectively validate the prominent advantages of the RBF + NDO method in trajectory tracking accuracy and stability.



**Fig. 27.** Six joint velocities of the robot.

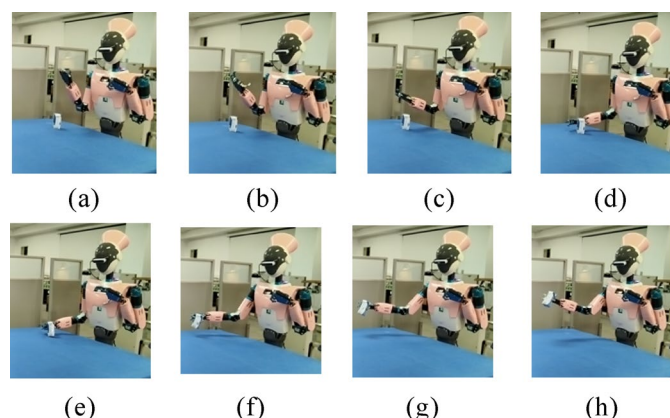


**Fig. 28.** Six joint torques of the robot.

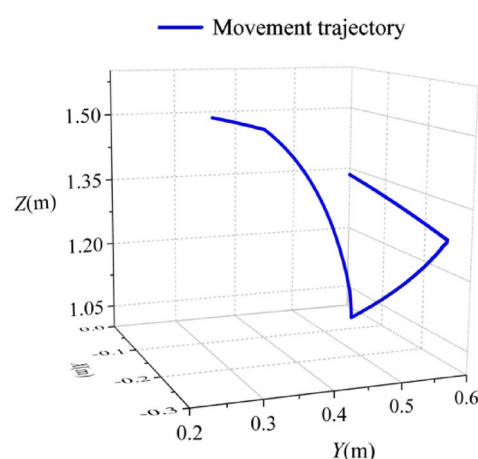


**Fig. 29.** Robot end trajectory in ROS.





**Fig. 30.** Seed Noid R7F robot experimental status.



**Fig. 31.** Robot end trajectory in the experiment.

Nevertheless, in practical implementation, the 6-DOF humanoid robotic manipulator is expected to accommodate various scenarios. Owing to the structural constraints of the robotic manipulator, the current study was confined to conducting trajectory tracking control under a fixed attitude. Future research efforts should be directed towards trajectory tracking control investigations that involve dynamically adjusting the attitude angles in accordance with task demands.

### Data availability

The datasets used and analyzed during the current study available from the corresponding author on reasonable request. Correspondence and requests for materials should be addressed to Y. Y.

Received: 3 January 2025; Accepted: 1 October 2025

Published online: 06 November 2025

### References

- Dereli, S. & Köker, R. A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: quantum behaved particle swarm algorithm. *Artif. Intell. Rev.* **53**, 949–964 (2020).
- Bai, Y. & Hsieh, S. J. A hybrid method using FABRIK and custom ANN in solving inverse kinematic for generic serial robot manipulator. *Int. J. Adv. Manuf. Technol.* **130**, 4883–4904 (2024).
- Zhou, X. et al. An improved inverse kinematics solution for 6-DOF robot manipulators with offset wrists. Published online by Cambridge University Press: 14 January (2022).
- Ames, B., Morgan, J. & Konidakis, G. Ikflow: generating diverse inverse kinematics solutions. *IEEE Rob. Autom. Lett.* **7**(3) (2022).
- Gao, R. Inverse kinematics solution of Robotics based on neural network algorithms. *J. Ambient Intell. Hum. Comput.* **11**, 6199–6209 (2021).
- Yin, X., Pan, L. & Cai, S. Robust adaptive fuzzy sliding mode trajectory tracking control for serial robotic manipulators. *Robot. Comput. Integr. Manuf.* **72**, 101884 (2021).
- Tran, T. & Ahn, K. K. Adaptive nonsingular fast terminal sliding mode control of robotic manipulator based neural network approach. *Int. J. Precis. Eng. Manuf.* **22**, 417–429 (2021).

8. Yin, H. et al. Kinematic control of humanoid upper body robot using virtual flexible joint dynamics primitive and Quasi-Sliding mode Observer[J]. *IEEE Trans. Industrial Electron.* (2025).
9. Xiao, B., Cao, L., Xu, S. & Liu, L. Robust tracking control of robot manipulators with actuator faults and joint velocity measurement uncertainty. *IEEE/ASME Trans. Mechatron.* **25**, 1354–1365 (2020).
10. Zhu, D., Du, B., Zhu, P. & Wu, W. Adaptive backstepping sliding mode control of trajectory tracking for robotic manipulators. *Complexity* **2020**, 1–11 (2020).
11. Su, Y. & Zheng, C. A new nonsingular integral terminal sliding mode control for robot manipulators. *Int. J. Syst. Sci.* **51**, 1418–1428 (2020).
12. Song, T., Fang, L. & Zhang, Y. *Nonsingular Fast Terminal Sliding Mode Control Based on a Novel Nonlinear Disturbance Observer for Robotic Systems [J]* (Asian Journal of Control, 2025).
13. He, W., Huang, H. & Ge, S. Adaptive neural network control of a robotic manipulator with Time-Varying output constraints. *IEEE Trans. Cybernetics.* **47**, 3136–3147 (2017).
14. Liu, Q. et al. Adaptive bias RBF neural network control for a robotic manipulator. *Neurocomputing* **447**, 213–223 (2021).
15. Stoffel, M., Gulakala, R., Bamer, F. & Markert, B. Artificial neural networks in structural dynamics: A new modular radial basis function approach vs. convolutional and feedforward topologies. *Comput. Methods Appl. Mech. Eng.* **364** (Issue), 112989–112989 (2020).
16. Narayan, J. & Dwivedy, S. K. Adaptive Control of a Pediatric Gait Exoskeleton: Integrating RBF Neural Network with Non-Singular Fast Terminal Sliding Mode Scheme. In *10th International Conference on Control, Decision and Information Technologies (CoDIT)*. (2024).
17. Xin, Z. & Ying, Q. Fuzzy sliding mode control of manipulator based on disturbance observer and RBF neural network. *Autom. Control Comput. Sci.* **57**, 123–134 (2023).
18. Sun, X., Zhang, L. & Gu, J. Neural-network based adaptive sliding mode control for Takagi-Sugeno fuzzy systems. *Inf. Sci.* **628**, 240–253 (2023).
19. Li, H., Hu, X., Zhang, X., Chen, H. & Li, Y. Adaptive radial basis function neural network sliding mode control of robot manipulator based on improved genetic algorithm. *Int. J. Comput. Integr. Manuf.* **37** (8), 1025–1039 (2024).
20. Viet-Thanh & Nguyen Bao-Long Pham, Thi-Van-Anh Nguyen, Ngoc-Tam Bui, Quy-Thinh Dao. Sliding mode control of antagonistically coupled pneumatic artificial muscles using radial basis neural network function. *SN Appl. Sci.* **5**, 246 (2023).
21. Sun, H., Huang, L. & He, L. Research on the trajectory tracking control of a 6-DOF manipulator based on Fully-Actuated system models. *J. Syst. Sci. Complexity.* **35**, 641–659 (2022).
22. Chen, Z., Zhang, Y., Nie, Y., Tang, J. & Zhu, S. Adaptive sliding mode control design for nonlinear unmanned surface vessel using RBFNN and Disturbance-Observer. *IEEE Access.* **8**, 45457–45467 (2020).
23. Narayan, J., Abbas, M., Patel, B. & Dwivedy, S. K. Adaptive RBF neural network-computed torque control for a pediatric gait exoskeleton system. *Intell. Serv. Robot.* **16**, 549–564 (2023).
24. Liu, Y. et al. A novel trajectory tracking control approach for uncertain 6-DOF manipulators based on fuzzy sliding mode of radial basis function neural network. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **237**(6), 1032–1044 (2023).
25. Liu, Z., Wang, Y., Yang, J. & Shi, J. In *Robotics and Rehabilitation Intelligence: First International Conference, ICRR I Fushun, China, September 9–11, 2020, Proceedings, Part I* 1. 64–82 (Springer). (2020).
26. Yu, J. Y. *Trajectory Tracking Based on Neural Network Sliding Mode controller[J]* (Authorea Preprints, 2023).
27. Roy, S. & Roy, S. B. Kar, I. N. Adaptive-robust control of Euler-Lagrange systems with linearly parametrizable uncertainty bound. *IEEE Trans. Control Syst. Technol.* **26**, 1842–1850 (2017).
28. Jian, Z., Hanqing, S., Jinzhu, Z., Tao, W. & Qingxue, H. Solving the inverse kinematics of A series robotic arm based on gradient descent. *Modular Mach. Tool Automatic Manuf. Technique* **11**, 78–82 (2024).
29. Zhang, Y., Qiu, M. & Gao, H. In *IJCAI Thirty-Second International Joint Conference on Artificial Intelligence*. 4602–4610 (2023).
30. Yu, J., Wu, M., Ji, J. C. & Yang, W. Neural Network-Based region tracking control for a Flexible-Joint robot manipulator. *J. Comput. Nonlinear Dyn.* **19**, 1–12 (2024).
31. Yang, J., Zhou, Z. & Ji, J. Nonlinear integral sliding mode control with adaptive extreme learning machine and robust control term for Anti-External disturbance robotic manipulator. *Arab. J. Sci. Eng.* **48**, 2375–2397 (2023).
32. Xi, R., Xiao, X., Ma, T. & Yang, Z. X. Adaptive sliding mode disturbance observer based robust control for robot manipulators towards assembly assistance. *IEEE Rob. Autom. Lett.* **7**, 1–1 (2022).

## Author contributions

The conceptualization was carried out by Y.W., Y.Y. and L.J. The methodology was developed by Y.W., Y.Y., and J.Y. The journals were reviewed by Z.W., S.W. and L.J. The software was designed and implemented by Y.Y., Z.W., and L.J. The verification process was conducted by Y.Y. and L.J. The overall supervision was overseen by Y.W. All authors have thoroughly read and consented to the final published version of the manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to Y.Y.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025