Research Article

Model-Free Adaptive Sliding Mode Robust Control with Neural Network Estimator for the Multi-Degree-of-Freedom Robotic Exoskeleton

Xiangjian Chen,1 Di Li,2 Pingxin Wang,3 Xibei Yang,1 and Hongmei Li1

1Jiangsu University of Science and Technology, School of Computer, Zhenjiang, China
2China Shipbuilding Industry Corporation 723, Yangzhou, China
3Jiangsu University of Science and Technology, School of Science, Zhenjiang, China

Correspondence should be addressed to Xiangjian Chen; sandra831209@163.com

Received 22 May 2019; Accepted 2 January 2020; Published 17 March 2020

Academic Editor: Eulalia Martínez

Copyright © 2020 Xiangjian Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new model-free adaptive robust control method has been proposed for the robotic exoskeleton, and the proposed control scheme depends only on the input and output data, which is different from model-based control algorithms that require exact dynamic model knowledge of the robotic exoskeleton. The dependence of the control algorithm on the prior knowledge of the robotic exoskeleton dynamics model is reduced, and the influence of the system uncertainties are compensated by using the model-free adaptive sliding mode controller based on data-driven methodology and neural network estimator, which improves the robustness of the system. Finally, real-time experimental results show that the control scheme proposed in this paper achieves better control performances with good robustness with respect to system uncertainties and external wind disturbances compared with the model-free adaptive control scheme and model-free sliding mode adaptive control scheme.

1. Introduction

Exoskeleton robots can significantly improve the motor ability and quality of life of people with reduced limb function. With the acceleration of the aging process, the number of patients with stroke, Parkinson’s disease, spinal injury, and lower extremity paraplegia will continue to increase. Even after professional rehabilitation or treatment, the body functions of this group of people can be restored to a certain degree, but it is difficult to return to their original state. Most people’s physical control ability becomes lower, and walking will become a very difficult task. Activities and quality seriously affect their daily lives. From the point of view of population aging trend and the growth of the potential beneficiary population, the study of full-body-assisted exoskeleton robot for the elderly and disabled can not only improve the motor ability, but also promote the social harmonious development. However, existing exoskeleton robots are generally larger in size and heavier in weight. Adaptation to the ground and flexibility of movement still need to be further improved. The working mode of most exoskeleton systems is that the wearer is carried by the robot, and the gait of different users is not fully coordinated. Although there are several products that can meet the requirements of exoskeleton robot control, the coupling performance of human exoskeleton robot used to help the elderly and disabled is still very insufficient.

One therapist Exoskeleton robots in this paper are designed to be worn to provide rehabilitation therapy for the stroke patients [1–4]. The effective control strategies are so important for the exoskeleton to operate coordinately with the human upper limb. General control methods [5, 6] are usually used for the control of manipulators.

In order to help the stroke patient, one therapist Exoskeleton robots were designed in this paper [1–4]; this robot is so complicated to be operated control to work with human
upper limb, so the general control methods [5, 6] are designed, and the other methods used the gravity and friction compensation controller also gained effective control performances [7–9]. Recently, researchers began to use more advanced control methods. But most of the control methods are designed based on the model of robot. Since the interacting model of this robot is time-varying with uncertainty, which makes the controller based on model schemes not enough effective. With the development of the technology, the data-driven method now is an alternative effective way, which does not need the model of interacting technology, the data-driven method now is an alternative effective way, which does not need the model of interacting model of exoskeleton, and the information can be obtained directly form the recorded process [10].

Data-driven Control (DDC) method does not contain the mathematical model information of the controlled process in designing process explicitly or implicitly, which only uses online or offline process data to design the controller. DDC has characteristics of convergence, stability, and robust under certain assumed conditions. In literature [11], they stated the necessity of the data-driven control theory in detail from three aspects, which are the control theory, the application of control theory, and the development of control theory. Meanwhile, many artificial intelligent methods [12–15] are used as supplementary methods for Data-driven control technology.

As a kind of data-driven control technology, Model-free Adaptive Control (MFAC) has been popular among a lot of scholars due to its complete theoretical analysis. The general idea of this algorithm can be summarized as follows. A new pseudopartial derivative concept is introduced under a certain assumed condition. At each working point of the discrete system, an equivalent virtual dynamic linearization model is established, and then the dynamic linearization model is used to design the controller and the adaptive law, to analyze stability, etc. The existence of pseudopartial derivatives is guaranteed by the Cauchy mean value theorem and the solution of some numerical equations, which is insensitive to those deterministic factors, such as the time-varying parameters, the time-varying structures, and the change of the system rank. Therefore, model-free adaptive control avoids the influence of unmodelled dynamics, and its uncertainty is mainly caused by disturbances and incomplete data. The model-free adaptive control was applied to the linear motor control, which was compared with PID control in terms of control accuracy in literature [16]. In literature [17], model-free adaptive control was combined with neural network control, and the parameters of the controller are adjusted online real-time through neural network, and then this algorithm was applied in the experiment of three capacities water tank system. In literature [18], the author proposed the data-based model-free adaptive sliding mode control for the multi degree-of-freedom robotic exoskeleton, but the control concise and control speed still can be influenced by the errors, so this paper introduced the neural network to deal with this problem.

The innovations of this paper are shown as follows: (1) the controller is designed only using the input and output data and without using the dynamic model of the system, which avoids the influence of model dynamics and system parameter uncertainty; (2) transverse cyclic pitch, longitudinal cyclic pitch, propeller collective pitch are used as control inputs, which avoids the use of unmeasured measurement of rotor flapping angle. It is necessary to achieve flight control; (3) the new controller can effectively enhance the anti-interference capability of the system; (4) the algorithm proposed has the characteristics of reliability and is verified in the comparison with other two algorithms.

2. Dynamic Model and Dynamic Transformation

In dynamic modeling, the dynamics of such a robotic system can be approximated by the following form:

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

where \( \dot{q} \in R^n \) denotes the vector of joint variables, \( \tau \in R^n \) denotes the inputs, \( M(k) \in R^{n \times n} \) represents the inertia, \( C(k), k \in R^{n \times n} \) denotes the centrifugal and Coriolis state, and \( G(k) \in R^n \) denotes the gravitational torques. The generalized force \( d \in R^n \) represents the disturbances, if the input can be represented as \( u(k) = \tau(k) \), and the output then can be rewritten as \( y(k) = \dot{q}(k) \), and the \( k_{th} \) moment discrete-time form of this model is

\[ y(k + 1) = \left( I - \frac{TC(k)}{M(k)} \right) y(k) + \frac{T}{M(k)} u(k) - \frac{T(G(k) + d(k))}{M(k)} \]

For the dynamic system equation (1) with \( \| \Delta y(k), \Delta u(k) \|^2 \neq 0 \) for each fixed \( k \), existing \( \Phi(k) \) can form a dynamic linearization data model:

\[ \Delta y(k + 1) = \Phi(k) \Delta y(k), \Delta u(k) \in G(k)^T, \]

where \( \Phi(k) = [\phi_1(k), \phi_2(k)], \phi_1(k), \phi_2(k) \in R^{n \times n} \) and \( \| \Phi(k) \| < b, b \) is a positive constant. For each \( k \), considering the following equation:

\[ \psi(k) = H(k) [\Delta y(k), \Delta u(k)]^T, \]

\( H(k) \in R^{n \times n} \) is one numerical matrix. If the condition \( \| \Delta y(k), \Delta u(k) \|^2 \neq 0 \) to be satisfied, then the above equation must have at least one solution \( H(k) \); in fact, it must have infinite number of solutions for each \( k \).

\[ \Phi(k) = H^T(k) + \left[ \frac{\partial y(k + 1)}{\partial y(k)} \right] \frac{\partial y(k + 1)}{\partial u(k)} \]

Now the dynamic system can be rewritten as

\[ y(k + 1) = \phi_1(k) \Delta y(k) + \phi_2(k) \Delta u(k) + y(k), \]

where \( \phi_1(k), \phi_2(k) \) are dynamic changed.

3. Control Design

The controller design is divided into three parts: first, to apply the model-free adaptive control theory to designing the controller; second, to enhance the robustness of the system and compensate the influence of unknown external disturbance, this paper combines the mode-free adaptive
control with both the sliding mode control and neural network estimator to compose the final controller; and finally, to prove that the convergence of the control error and the stability of the closed loop system with theoretical ways.

3.1. Designing Model-Free Adaptive Controller. There is the general nonlinear discrete-time MIMO system as follows:

\[ y(k+1) = f(y(k), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)) \]

(7)

where \( u(k), y(k) \in \mathbb{R}^n \) are, respectively, the input and output vectors of the system, \( f(\ldots) \in \mathbb{R}^n \) is smooth nonlinear function, and \( n_y, n_u \) are the order of system.

According to Theorem 1, the system equation (5) can be rewritten into

\[ \Delta y(k+1) = \Phi(k) \Delta u(k), \]

(8)

where

\[ \Phi(k) = [\varphi_1, \varphi_2, \ldots, \varphi_n]. \]

(9)

Note 1. In equation (9), \( \Phi(k) \) represents a time-varying dynamic linearization parameter. It can be computed by estimation algorithm such as least square algorithm, projection algorithm. In this paper, the estimation algorithm in paper [17] is used to calculate the value of the parameters.

Considering the system described by equation (9) and combining the definition Theorem 1, the controlled system can be rewritten as

\[
\begin{align*}
&\begin{bmatrix}
y_1(k+1) \\
y_2(k+1) \\
\vdots \\
y_n(k+1)
\end{bmatrix} = \\
&\begin{bmatrix}
y_1(k) \\
y_2(k) \\
\vdots \\
y_n(k)
\end{bmatrix} + \\
&\begin{bmatrix}
\varphi_{11}(k) \\
\varphi_{22}(k) \\
\vdots \\
\varphi_{nn}(k)
\end{bmatrix} \begin{bmatrix}
\Delta u_1(k) \\
\Delta u_2(k) \\
\vdots \\
\Delta u_n(k)
\end{bmatrix} + \\
&\begin{bmatrix}
f_{NL1}(k) \\
f_{NL2}(k) \\
\vdots \\
f_{NLn}(k)
\end{bmatrix},
\end{align*}
\]

(10)

The coupling effect between systems is considered as a disturbance, and it can be compensated estimating by neural network.

First, the pseudomatrix \( \Phi(k) \) of the Jacobi matrix is calculated by the method given in [17].

\[
\hat{\Phi}^T(k) = \Phi^T(k-1) + \frac{\eta \Delta u(k-1) u^T}{\|u\| \|\Delta u(k-1)\|^2}
\]

\[
\times \left[ \Delta y^T(k) - \Delta u(k-1) \hat{\Phi}^T(k-1) \right].
\]

(12)

Taking \( \delta_{\hat{\Phi}}(k) = \varphi_i(k) \hat{\varphi}_i(k), i = 1, 2, \ldots, n, \) it represents the error of parameter estimation.

Combining equations (11) and (12), we can get

\[
\begin{bmatrix}
y_1(k+1) \\
y_2(k+1) \\
\vdots \\
y_n(k+1)
\end{bmatrix} = \\
\begin{bmatrix}
y_1(k) \\
y_2(k) \\
\vdots \\
y_n(k)
\end{bmatrix} + \\
\begin{bmatrix}
\varphi_{11}(k) \\
\varphi_{22}(k) \\
\vdots \\
\varphi_{nn}(k)
\end{bmatrix} \begin{bmatrix}
\Delta u_1(k) \\
\Delta u_2(k) \\
\vdots \\
\Delta u_n(k)
\end{bmatrix} + \\
\begin{bmatrix}
f_{NL1}(k) \\
f_{NL2}(k) \\
\vdots \\
f_{NLn}(k)
\end{bmatrix},
\]

(13)

From this, we can know that \( f_{NLi}(k) \) includes unmodeled dynamics, and the coupling effect between systems includes time-varying parameter estimation error. It can be regarded as generalized disturbance signal and estimated by the neural network.

3.2. Design of Sliding Mode Controller. Define the \( i_{th} \) loop tracking error:

\[ e_i(k+1) = y_{ri}(k+1) - y_i(k+1), \quad i = 1, 2, \ldots, n. \]

(14)

Define the \( i_{th} \) planar function of loop sliding mode controller:

\[ s_i(k) = c_i^T E_i(k). \]

(15)

where \( c_i^T = [1, c_{i1}, c_{i2}], c_{i1} \) is the constant greater than 0. \( E_i(k)T = [e_i(k), e_i(k-1)], y_{ri}(k) \) is the bounded reference input.

The discrete convergence law is

\[ s_i(k+1) = (1 - q_i) s_i(k) - \varepsilon_i \text{sgn}(s_i(k)), \quad \varepsilon_i > 0, 0 < q_i < 1. \]

(16)

Equivalent control can be obtained by

\[ s_i(k+1) = (1 - q_i) s_i(k) - \varepsilon_i \text{sgn}(s_i(k)) = c_i^T E_i(k+1). \]

(17)

From equations (16)–(19), we can get

\[ c_i^T E_i(k+1) = e_i(k+1) + c_{q_i} e_i(k) = (1 - q_i) s_i(k) - \varepsilon_i \text{sgn}(s_i(k)), \]

(18)

\[ e_i(k+1) = (1 - q_i) s_i(k) - \varepsilon_i \text{sgn}(s_i(k)) - c_{q_i} e_i(k), \]

which is
\[ y_{ri}(k + 1) - y_i(k + 1) = (1 - q_i)s_i(k) - \varepsilon \text{sgn}(s_i(k)) - c_{i0}e_i(k), \]
\[ \Phi_i(k)u_i(k) = y_{ri}(k + 1) - y_i(k) - (1 - q_i)s_i(k) + \varepsilon \text{sgn}(s_i(k)) + c_{i0}e_i(k) - f_{NL}(k), \]
\[ \bar{u}_i(k) = \frac{1}{\left( \Phi_i(k) + \sigma \right)} \left( y_{ri}(k + 1) - y_i(k) - c_i^T E_i(k) - f_{NL}(k) + \varepsilon \text{sgn}(c_i^T E_i(k)) \right), \]

where \( \tau_i^T = [1 - q_i - c_{i0}, (1 - q_i)c_{i0}] \).

**Note 2.** When is very small, the control input equation (12) may becomes very large or even unbounded. In order to avoid this phenomenon, we use a very small positive number.

**Note 3.** The control law equation (12) contains the unknown term \( f_{NL}(k) \). In this paper, the radial basis neural network is used to estimate this part.

### 3.3. Neural Network Estimator

In this paper, we used the radial basis function (RBF) neural network, which uses neural network approximation to predict disturbance and other uncertainties, and then combined with adaptive sliding mode control to realize the control of disturbed discrete nonlinear systems. The introduction of neural network estimator is designed to improve the dynamic performance of the system. The structure diagram of the control system is shown in Figure 1 (take the case of two inputs/two outputs as an example).

\( \Delta u, \Delta u, \Delta y_c \) are system control signals, output signals, and expected signals. \( f_{NN} \) is the output of the \( i \)th neural network predictor, that is, the estimate of the \( f_{NL} \).

The neural network structure and algorithm used in this paper are the same as reference [17], so it is omitted.

### 3.4. Stability Analysis

Suppose 1: The system has a globally asymptotically stable zero dynamics. Based on the approximation property of neural network knowledge, after properly selecting network structure and its network parameters, \( f_{NN} \) can approximate the generalized disturbance signal \( f_{NL} \) with arbitrary precision, so there is an arbitrarily small positive number \( c_i \) and the prediction error is always less than \( \zeta_i \), which is the output \( y_{NN} \) of neural network. Meet the following conditions:

\[ |f_{NN}(k) - f_{NL}(k)| \leq c_i, \quad i = 1, 2, \ldots, n. \]  

**Definition of positive definite function \( \nu_i(k) \) is**

\[ V_i(k) = \left[ c_i^T E_i(k) \right]^2 + \frac{1}{\sum_{j=1}^{k+1} \xi_j^2(j)} - \frac{1}{\sum_{j=1}^{k+1} \xi_j^2(j)}, \quad i = 1, 2, \ldots, n. \]  

According to the definition of \( V_i(k) \), we can know \( V_i(k) > 0 \). And

\[ V_i(k + 1) = V_i(k + 1) - V_i(k) = \left[ c_i^T E_i(k + 1) \right]^2 + \frac{1}{\sum_{j=1}^{k+1} \xi_j^2(j)} - \frac{1}{\sum_{j=1}^{k+1} \xi_j^2(j)} - \left[ c_i^T E_i(k) \right]^2 \]
\[ = \left( (1 - q_i)c_i^T E_i(k) - \varepsilon \text{sgn}c_i^T E_i(k) \right)^2 - \left[ c_i^T E_i(k) \right]^2 + \frac{\sum_{j=1}^{k+1} \xi_j^2(j) - \sum_{j=1}^{k+1} \xi_j^2(j)}{\sum_{j=1}^{k+1} \xi_j^2(j)} \]
\[ = -2q_i \left[ c_i^T E_i(k) \right]^2 + q_i^2 \left[ c_i^T E_i(k) \right]^2 - 2\varepsilon_i(1 - q_i)c_i^T E_i(k)\text{sgn}c_i^T E_i(k) + \varepsilon_i^2 - \frac{\sum_{j=1}^{k+1} \xi_j^2(j)}{\sum_{j=1}^{k+1} \xi_j^2(j)} \sum_{j=1}^{k+1} \xi_j^2(j) \]
\[ = -q_i(2 - q_i) \left[ c_i^T E_i(k) \right]^2 - 2\varepsilon_i(1 - q_i)c_i^T E_i(k)\text{sgn}c_i^T E_i(k) - \frac{\sum_{j=1}^{k+1} \xi_j^2(j)}{\sum_{j=1}^{k+1} \xi_j^2(j)} \sum_{j=1}^{k+1} \xi_j^2(j) + \varepsilon_i^2. \]

**Definition of tracking error:**

\[ e_i(k) = y_{ni}(k) - y_i(k). \]

As shown in equation (22), the first 3 items on the right side are less than 0. So if selecting \( \varepsilon_i \) to be small enough, make sure that \( \Delta V_i(k + 1) < 0 \) is set up. It is known that \( V_i(0) \) is bounded by the previous definition and the system initial value, which means that the system is stable under the control law, and the tracking error of the system converges to the neighborhood of zero.
It is also known from the previous proof that \( \{ y_i(k) \}, i = 1, 2, \ldots, n \) is bounded. Then by equation (16), we know that the system has a global asymptotically stable zero dynamics. Therefore, the existence of constants \( M_1, M_2, \) and \( k_0 \) satisfies the following condition:

\[
|u_i(k-1)| \leq M_1 \max_{0 \leq r \leq k} |y_i(\tau)| + M_2, \quad i = 1, 2, \ldots, n.
\]

(24)

Obviously, \( \{ u_i(k) \}, i = 1, 2, \ldots, n \), which is also a bounded sequence.

### 4. Simulation and Results

The mechanical structure of this robotic exoskeleton has been shown in Figure 2.

In this simulation experiment, the wrist two joints keep unchanged, and the other three joints will be controlled by the different controllers, which include model-free adaptive sliding mode control combined with neural network (MFASMCNN), model-free adaptive sliding mode control (MFASMC), and model-free adaptive control (MFAC). The simulation results will demonstrate the superiority of the proposed algorithm. In the last simulation part, the five joints of the robot were all under control of MFASMCNN and MFASMC approaches to verify the good performance of MFASMCNN when the simulation time is 10\(s\) and the discrete-time sample period \(T = 0.001\)s.

#### 4.1. Controlling Three Joints of Robot

The desired angle velocities \([1, 2]\) of the robot have been set as follows:

\[
y_{d1} = \begin{cases} 
-0.4, & y_{d1} < -0.4, \\
\frac{2\pi}{10} \cos\left(\frac{2\pi}{10} t\right), & -0.4 \leq y_{d1} \leq 0.4, \\
0.4, & y_{d1} > 0.4,
\end{cases}
\]

\[
y_{d2} = \begin{cases} 
-0.6, & y_{d2} < -0.6, \\
\frac{2\pi}{7} \cos\left(\frac{2\pi}{7} t\right), & -0.6 \leq y_{d2} \leq 0.6, \\
0.6, & y_{d2} > 0.6,
\end{cases}
\]

\[
y_{d3} = \begin{cases} 
-0.8, & y_{d3} < -0.8, \\
\frac{2\pi}{5} \cos\left(\frac{2\pi}{5} t\right), & -0.8 \leq y_{d3} \leq 0.8, \\
0.8, & y_{d3} > 0.8.
\end{cases}
\]
4.2. Performance of Model-Free Adaptive Control.

Figure 3 demonstrates the simulation results used by model-free adaptive control scheme, where Figures 3(a)–3(c) show the performances of outputs $y_1(k)$, $y_2(k)$, $y_3(k)$. Figure 3(d) shows the control inputs of $u_1(k)$, $u_2(k)$, and $u_3(k)$ respectively. From Figures 3(a)–3(c), we can see that control inputs have some fluctuation.

4.3. Performance of Model-Free Adaptive Sliding Mode Control.

Figure 4 shows the control performance using the model-free adaptive sliding mode control method, where Figures 4(a)–4(c) show control performances of outputs of $y_1(k)$, $y_2(k)$, $y_3(k)$, form which we can see that the convergence of the method is slow. Figure 4(d) shows control inputs of $u_1(k)$, $u_2(k)$, $u_3(k)$, respectively, from which we can see that the control performance of the MFASMC on the robotic exoskeleton is tracking well, but have a big fluctuation at first for control inputs shown in Figure 4(d).

4.4. Performance of Model-Free Adaptive Sliding Mode Combined with Neural Network Control.

This part shows the control performance of model-free sliding mode combined with neural network control method, the initial values shown as
Figure 4: The model-free adaptive sliding mode control simulation results of three joints under control. (a) Tracking performance of $y_1(k)$. (b) Tracking performance of $y_2(k)$. (c) Tracking performance of $y_3(k)$. (d) Control inputs of $u_1(k)$, $u_2(k)$ and $u_3(k)$.

Figure 5: Performance of model-free adaptive sliding model control combined with neural network control of three joints. (a) Tracking performance of $y_1(k)$. (b) Tracking performance of $y_2(k)$. (c) Tracking performance of $y_3(k)$. (d) Control inputs of $u_1(k)$, $u_2(k)$ and $u_3(k)$. 
Table 1: RMS values of errors by MFAC.

| | (0.2) | (2.4) | (4.6) | (6.8) | (8.10) |
|---|---|---|---|---|---|
| e1 | 0.0227 | 0.0090 | 0.0160 | 0.0155 | 0.0154 |
| e2 | 0.0232 | 0.0204 | 0.0167 | 0.0179 | 0.0072 |
| e3 | 0.0097 | 0.0103 | 0.0075 | 0.0058 | 0.0032 |

Table 2: RMS values of errors by MFASMC.

| | (0.2) | (2.4) | (4.6) | (6.8) | (8.10) |
|---|---|---|---|---|---|
| e1 | 0.0292 | 0.0028 | 0.0019 | 0.0009 | 0.0021 |
| e2 | 0.0411 | 0.0045 | 0.0031 | 0.0023 | 0.0015 |
| e3 | 0.0230 | 0.0018 | 0.0016 | 0.0019 | 0.0019 |

Table 3: RMS values of errors by MFASMCNN.

| | (0.2) | (2.4) | (4.6) | (6.8) | (8.10) |
|---|---|---|---|---|---|
| e1 | 0.0186 | 0.0014 | 0.0013 | 0.0007 | 0.0014 |
| e2 | 0.0261 | 0.0035 | 0.0021 | 0.0016 | 0.0009 |
| e3 | 0.0138 | 0.0012 | 0.0009 | 0.0011 | 0.0011 |

\[
y(k - 1) = [0, 0, 0]^T u(k - 1) = [0, 0, 0]^T, \\
y(k - 2) = [0, 0, 0]^T u(k - 2) = [0, 0, 0]^T, \\
\varphi_1 = \text{diag}([1, 1, 1]), \\
\varphi_2 = \text{diag}([1, 1, 1]).
\]

The controller parameters are \( \eta = \rho = \lambda = \mu = 1 \), \( q = 800 \), \( \varepsilon = 10 \), \( \Gamma = \text{diag}([4, 4, 1]) \).

Figure 5 shows the control performance of model-free adaptive sliding mode combined with neural network method, where Figures 5(a)–5(c) show the performances of outputs \( y_1, y_2, y_3 \), from which we can see that the tracking results perform quite satisfactory. Figure 5(d) presents the control inputs of \( u_1, u_2, u_3 \), from which we can also see that control inputs are quite stable.

4.5. Comparison Analysis. In this part, comparison has been done to demonstrate the control performance of the proposed method; for this, the root mean square (RMS) values of errors are created by the three kinds of controller, and the results are shown in Tables 1–3, respectively, from comparison results of the RMS values, we can conclude that the convergence performance and speed of the proposed method is better than two other controllers although the proposed controller has a little overshoot at first time.

4.6. Five Joints under Control. The desired angle velocities of robot are set as follows:

\[
y_{d1} = \begin{cases} 
-0.4, & y_{d1} < -0.4, \\
2\pi/10 \cos \left(\frac{2\pi}{10}t\right), & -0.4 \leq y_{d1} \leq 0.4, \\
0.4, & y_{d1} > 0.4,
\end{cases}
\]

\[
y_{d2} = \begin{cases} 
-0.5, & y_{d2} < -0.5, \\
2\pi/8 \cos \left(\frac{2\pi}{8}t\right), & -0.5 \leq y_{d2} \leq 0.5, \\
0.5, & y_{d2} > 0.5,
\end{cases}
\]

\[
y_{d3} = \begin{cases} 
-0.6, & y_{d3} < -0.6, \\
2\pi/7 \cos \left(\frac{2\pi}{7}t\right), & -0.6 \leq y_{d3} \leq 0.6, \\
0.6, & y_{d3} > 0.6,
\end{cases}
\]

\[
y_{d4} = \begin{cases} 
-0.7, & y_{d4} < -0.7, \\
2\pi/6 \cos \left(\frac{2\pi}{6}t\right), & -0.7 \leq y_{d4} \leq 0.7, \\
0.7, & y_{d4} > 0.7,
\end{cases}
\]

\[
y_{d5} = \begin{cases} 
-0.8, & y_{d5} < -0.8, \\
2\pi/5 \cos \left(\frac{2\pi}{5}t\right), & -0.8 \leq y_{d5} \leq 0.8, \\
0.8, & y_{d5} > 0.8.
\end{cases}
\]

The initial values of the proposed method are

\[
y(k - 1) = [0, 0, 0, 0, 0]^T u(k - 1) = [0, 0, 0, 0, 0]^T, \\
y(k - 2) = [0, 0, 0, 0, 0]^T u(k - 2) = [0, 0, 0, 0, 0]^T, \\
\varphi_1 = \text{diag}([1, 1, 1, 1, 1]), \\
\varphi_2 = \text{diag}([1, 1, 1, 1, 1]).
\]

The controller parameters are \( \eta = \rho = \lambda = \mu = 1, \varepsilon = 10, q = 800, \Gamma = \text{diag}([4, 4, 4, 0.2, 0.2]) \).

Figure 6 shows the control performance based on MAFSMC and MFASMCNN, where Figures 6(a)–6(e) show the performances of outputs \( y_1, y_2, y_3, y_4, y_5 \), in which we can see that the convergence of the proposed method is quite well, and Figure 6(f) shows the control inputs of \( u_1, u_2, u_3, u_4, u_5 \) respectively, from which we can see that the control is stable.

From the above simulation results in Figures 3 and 4, we can see that the MFAC, MFASMC, and MFASMCNN methods are satisfactory under the parameter time-
Figure 6: Comparison results with five joints under control ("11" represents MFSMC, "22" represents MFASMCNN). (a) Tracking performance of \(y_1(k)\). (b) Tracking performance of \(y_2(k)\). (c) Tracking performance of \(y_3(k)\). (d) Tracking performance of \(y_4(k)\). (e) Tracking performance of \(y_5(k)\). (f) Control inputs of \(u_1(k), u_2(k), u_3(k), u_4(k)\) and \(u_5(k)\) with MFSMC. (g) Control inputs of \(u_1(k), u_2(k), u_3(k), u_4(k)\) and \(u_5(k)\) with MFASMCNN.
invariable system and also the control inputs of MFASMCNN method are more stable under the big fluctuation.

5. Conclusions

In this paper, we analyzed the transformation and linearization of the robotic exoskeleton dynamics, and a model-free adaptive sliding mode controller with neural network estimator based on data-driven methodology was designed. Then the convergence of control error and the stability of the closed loop system are proved. The experimental results show that the control design proposed in this paper performs well and has good robustness in system uncertainties and external disturbances compared with MFAC and MFASMC.

Data Availability

The data supporting this paper are from the references reported studies and datasets, which have been cited.

Conflicts of Interest

The author(s) declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This paper was supported by National Natural Science Found of China (Nos. 61502211, 61572242, and 61702234).

References

[1] N. N. Gheidari, “Effects of robot-assisted therapy on stroke rehabilitation in upper limbs: systematic review and meta-analysis of the literature,” Journal of Rehabilitation Research & Development, vol. 49, pp. 479–495, 2012.

[2] H.-B. Kang and J.-H. Wang, “Adaptive robust control of 5 DOF Upper-limb exoskeleton robot,” International Journal of Control, Automation and Systems, vol. 13, no. 3, pp. 733–741, 2015.

[3] Q. Li, D. Wang, Z. Du, and L. Sun, “A novel rehabilitation system for upper limbs,” in Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005, IEEE-EMBS 2005, pp. 6840–6843, IEEE, Shanghai, China, January 2006.

[4] H. S. Lo and S. Q. Xie, “Exoskeleton robots for upper-limb rehabilitation: state of the art and future prospects,” Medical Engineering & Physics, vol. 34, no. 3, pp. 261–268, 2012.

[5] C. Carignan, J. Tang, and S. Roderick, “Development of an exoskeleton haptic interface for virtual task training,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, IROS 2009, pp. 3697–3702, IEEE, St. Louis, MO, USA, October 2009.

[6] R. A. R. C. Gopura, K. Kiguchi, and Y. Li, “Sueful-7: a 7DOF upper-limb exoskeleton robot with muscle-model-oriented EMG-based control,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, IROS 2009, pp. 1126–1131, IEEE, St. Louis, MO, USA, October 2009.

[7] T. Nef, M. Guidali, V. Klamroth-Marganska, and R. Rien, “ARMin—exoskeleton robot for stroke rehabilitation,” IFMBE Proceedings, vol. 6, pp. 127–130, 2009.

[8] A. H. Stienen, E. E. Hekman, F. C. Van der Helm et al., “Dampace: dynamic force-coordination trainer for the upper extremities,” in Proceedings of the IEEE 10th International Conference on Rehabilitation Robotics, 2007. ICORR 2007, pp. 820–826, IEEE, Noordwijk, Netherlands, June 2007.

[9] R. Vertechy, A. Frisoli, A. Dettori, M. Solazzi, and M. Bergamasco, “Development of a new exoskeleton for upper limb rehabilitation,” in Proceedings of the IEEE International Conference on Rehabilitation Robotics, 2009, ICORR 2009, pp. 188–193, IEEE, Kyoto, Japan, June 2009.

[10] S. Yin, S. X. Ding, X. Xie, and H. Luo, “A review on basic data-driven approaches for industrial process monitoring,” IEEE Transactions on Industrial Electronics, vol. 61, no. 11, pp. 6418–6428, 2014.

[11] Z. S. Hou and S. T. Jin, Model-free Adaptive Control: Theory and Applications, Science Press, Beijing, China, 2013.

[12] X. Chen, D. Li, X. Wang, X. Yang, and H. Li, “Rough intuitionistic type-2 fuzzy c-means clustering algorithm for MR image segmentation,” IET Image Processing, vol. 13, no. 4, pp. 607–614, 2019.

[13] G. Wang and Z. Huang, “Data-driven fault-tolerant control design for wind turbines with robust residual generator,” IET Control Theory & Applications, vol. 9, no. 7, pp. 1173–1179, 2015.

[14] P. X. Wang and Y. Yao, “CE3: a three-way clustering method based on mathematical morphology,” Knowledge-based Systems, vol. 155, pp. 55–65, 2018.

[15] X. J. Chen, D. Li, X. Wang, and X. B. Yang, “A deep convolutional neural network with fuzzy rough sets for fer,” IEEE Access, vol. 8, 2019.

[16] Y. Zhou and Z. S. Hou, “Data-driven MFAC for a class of discrete-time nonlinear systems with RBFNN,” IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 5, pp. 1013–1020, 2014.

[17] Y. Hou, D. Wang, and D. Xu, “Enhanced data driven model-free adaptive yaw control of unmanned-aerial-vehicle helicopter,” International Journal of Multimedia and Ubiquitous Engineering, vol. 11, no. 2, pp. 207–218, 2016.

[18] X. F. Wang, L. A. Xing, J. H. Wang, X. K. Fang, and X. F. Zhu, “Data-driven model-free adaptive sliding mode control for the multi degree-of-freedom robotic exoskeleton,” Information Sciences, vol. 16, no. 7, pp. 1–12, 2015.