Hindawi
Complexity
Volume 2018, Article ID 4034320, 13 pages
https://doi.org/10.1155/2018/4034320

Research Article

An RBFNN-Based Direct Inverse Controller for PMSM with Disturbances

Shengquan Li,[1,2] Juan Li,[1] and Yanqiu Shi[1]

[1] *School of Hydraulic, Energy and Power Engineering, Yangzhou University, Yangzhou 225127, China*
[2] *Jiangsu Engineering Research Center on Meteorological Energy Using and Control, Nanjing University of Information Science and Technology, Nanjing 210044, China*

Correspondence should be addressed to Shengquan Li; sqli@yzu.edu.cn

Received 30 March 2018; Accepted 9 May 2018; Published 13 June 2018

Academic Editor: Jing Na

Considering the system uncertainties, such as parameter changes, modeling error, and external uncertainties, a radial basis function neural network (RBFNN) controller using the direct inverse method with the satisfactory stability for improving universal function approximation ability, convergence, and disturbance attenuation capability is advanced in this paper. The weight adaptation rule of the RBFNN is obtained online by Lyapunov stability analysis method to guarantee the identification and tracking performances. The simulation example for the position tracking control of PMSM is studied to illustrate the effectiveness and the applicability of the proposed RBFNN-based direct inverse control method.

1. Introduction

In engineering systems, various uncertainties exist including parameter uncertainties, unmodeled dynamics, and unknown external disturbances, which often bring adverse effects to stability and performance of the whole control systems. With growing interests of high-precision control systems, how to develop efficient control approaches to counteract the adverse effects, caused by various uncertainties, is an active topic in both the control theory and application [1], for example, robust control [2], sliding model control [3, 4], adaptive control [5, 6], backstepping control [7, 8], and disturbance estimation-based compound control [8–12]. The effectiveness of these control methods has been proved by their applications in some industrial areas. Unfortunately, due to the need of exact knowledge of these control systems, the improvement of performance is limited. Moreover, some of these control methods lack online learning mechanism. The control performance cannot be guaranteed while the industrial systems are subjected to drastic internal and external disturbances. Considering the limitation of the kinds of disturbances estimate methods, some researches proposed many other disturbances estimation methods [13, 14]. In [13], an approximation-free funnel function is proposed to guarantee the transient and asymptotic behavior of the tracking performance. In order to approximate unknown nonlinearities and to dramatically diminish the computational costs, a novel high-order neural network with only a scalar weight is introduced in [14].

On the basis of approachability, neural networks have been used to control unknown nonlinear dynamic systems, since it can be proved that a neural network can be trained to approximate any nonlinear function with the any given accuracy under certain condition [15–19]. The use of neural network learning ability avoids complex mathematical analysis in solving the control problem of plant dynamics with high complexity and nonlinearity. It is commonly known that this ability of neural network is the obvious advantage compared with traditional control methods.

However, effectively handling of the presence of disturbances is not well developed within the adaptive neural network control method. So reinforcing adaptive neural network controllers with disturbance attenuation capabilities still remains a challenging task in enormous practical applications. An initial approach is provided in [20], since then, many works have been emerged in [21–27]. Recently, the output feedback control scheme combining a model-based controller with a neural network feed-forward compensator to model the unknown system dynamics is proposed in [23, 24]. For the purpose of enhancing the stability, an additional robust controller is needed to be introduced to solve the problems arising from approximation errors of the neural network. The RBFNN-based disturbance observer is proposed in [21, 22] to estimate the lumped disturbances, that is, external disturbance from uncertain external condition and internal disturbance caused by parameter variations or modeling errors. A method of indirect adaptive neural network control is presented in [23] to identify high-order nonlinear continuous plant. Moreover, control parameters will be updated with the identified model information to increase the control performance.

Generally, weight matrix parameters of neural networks are adjusted with gradient method; however, there is currently no systematic way of ensuring when these methods will be successful. And analysis becomes very complicated when learning and control are attempted simultaneously, even the simplest control situation, such as a linear, time-invariant process, and a linear feedback control law, becomes a high-dimensional, coupled, nonlinear problem with the addition of online tuning of the neural network controller parameters [28]. And for the purpose of stable and efficient online control, the sufficiently accurate identified information system by using the gradient method is a necessary prerequisite. That is to say, the off-line training which is time-consuming is needed to provide a good starting point for the online adaptive control. Application of gradient optimization methods contains instability mechanisms, since there exists parameter variations and internal and external disturbances. Some adaptive neural network control strategies employ the enhanced gradient algorithm or avoid the gradient method to obtain high performance, since the neural network trained by gradient algorithm may not exactly reconstruct a certain required nonlinear function. From the reliability point of view, adding more components to a system will involve a higher probability of malfunction. Motivated by the fact that the existence of a robust control Lyapunov function is a necessary and sufficient condition for robust stabilization via a suitable control law. In this paper, an adaptive neural network-based direct inverse controller (NBIC) with a RBFNN with guaranteed stability, convergence, and disturbance attenuation capabilities is investigated for the lumped disturbances, that is, the unknown nonlinear system with parameter variations, unmodeled dynamic response, and bounded external disturbances. The scheme does not need to design extra controllers but only using one RBFNN which acts not only on the feedback controller but also can compensate for the external disturbances. Thus, the structure of the control system is less complicated. The weights of RBFNN is tuned online based on the Lyapunov theory which will not only guarantee the given performance for this system but will also illuminate the relationships between performance and the parameters of the NBIC. Moreover, the control scheme not only guarantees the stabilities of the closed-loop system but also the tracking error will astringents to a small neighborhood of the origin. And the neural network controller can obtain the benefits of model-based control without a priori knowledge of system dynamics or without the computational burden of classical dynamic. Also, under the circumstance without disturbances, the proposed control strategy can ensure the uniform ultimate boundedness property if tracking error with respect to a compact set around the origin of any small area.

The structure of this article is as follows. In Section 2, some theoretical preliminaries are addressed, including mathematical notations, the description of the unknown nonlinear system under research, control objective, and the description of the RBFNN. In Section 3, the design procedure of adaptive neural network using a direct inverse controller is introduced and stability analysis is also given in this section. The application on the high-precision position tracking of PMSM servo system and the simulation comparisons and analysis with the model-based inverse controller are presented in Section 4. Finally, some conclusions are described in Section 5.

2. Preliminaries

Let R and R^n be the real number set and the n-dimensional vector space, respectively. Define $R^{m\times n}$ and let $m \times n$ be the real matrix space. The parameters of $\lambda_{\min}(P)$ and $\lambda_{\max}(P)$ are the minimum and maximum eigenvalue of matrix P, respectively. The symbol $tr(\bullet)$ indicates the trajectory of the matrix $[\bullet]$, $\|\bullet\|$ denotes the Euclidean norm, $\|W\|_F$ is the Frobenius norm (F norm). According to the particularity of F norm, there are $\|W\|_F^2 = \sum_{i,j}|w_{i,j}|^2 = tr(WW^T) = tr(W^TW)$ and $tr(WW^T) = tr(\tilde{W}^T(W - \tilde{W})) \leq \|\tilde{W}^T\|\|W\|_F - \|\tilde{W}\|_F^2$.

2.1. The Statement of the Problem. A class of single-input single-output (SISO) nonlinear system with unknown disturbances can be described with the following Brunovsky form [29]:

$$\begin{aligned}\dot{x}_1 &= x_2,\\ &\vdots\\ \dot{x}_{n-1} &= x_n,\\ \dot{x}_n &= f(x) + g(x)u + d,\\ y &= x_1,\end{aligned} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)^T \in R^n$ means the state vector of this system, $u \in R$ and $y \in R$ are the input and output, $f(x) \in R^n$ and $g(x) \in R^n$ R^n are the unknown continuous functions including internal uncertainties, and d denotes external disturbances. In practice, many systems such as chemical reactions, PMSMs, and robots are essentially nonlinear, whose input variables may enter in the systems nonlinearly as described by the above general form. In terms of nonlinear control literatures, these systems are feedback linearizable and have a relative degree equal to note. The smooth function satisfies $g(x) \neq 0$ with $\forall x \in R^n$, and it implies that the function $g(x)$ is bounded away from zero with strictly positive or negative value. The control goal addressed here is to find a suitable control law u, so that the system output y can track a bounded reference trajectory y_d with a satisfactory accuracy in the presence of internal disturbances caused by parameter uncertainties and external disturbances d, while all involved variables, such as u, y, x, and d, should be bounded.

Assumption 1. In the compact sets $S \subset R$, $g(x)$ has inverse function and boundary, as $|g(x)| \leq a < \infty$, and a is arbitrary nonnegative constant.

Assumption 2. The system state variable x can be observable.

Assumption 3. The system external disturbance d is defined by a known constant $d_0 > 0$, that is $|d| \leq d_0$.

Assumption 4. The reference trajectory x_d is continuous and bounded known function of time with bounded known derivatives up to the nth order.

The vectors x_d and the tracking error $e \in R^n$ are defined as the following two equations.

$$\mathbf{x}_d = \left[x_d \quad \dot{x}_d \quad \cdots \quad x_d^{n-1} \right] \in R^n,$$
$$\mathbf{e} = \mathbf{x} - \mathbf{x}_d = \left[e_1 \quad e_2 \quad \cdots \quad e_n \right]^T = \left[e_1 \quad \dot{e}_1 \quad \cdots \quad e_1^{n-1} \right]^T. \tag{2}$$

If the exact knowledge of the system dynamics and the external disturbances can be obtained precisely, that is, functions $f(x)$, d, and $g(x)$ are known exactly, the following model-based inverse controller (MBIC) can be obtained as follows:

$$u = g^{-1}(x)(r - f(x) - d), \tag{3}$$

$$r = x_d^n - k_n e_1 - k_{n-1} e_2 - \cdots - k_1 e_n, \tag{4}$$

where $k_1, \ldots, k_n$ is the system coefficients related to transient performance of the closed-loop system. Therefore, the closed-loop system constituted by (1), (2), (3), and (4) can be redescribed as follows:

$$e_1^n = K^T e, \tag{5}$$

where $K = (k_n, k_{n-1}, \ldots, k_1)^T$ is the coefficient matrix. And this matrix should be reasonably chosen so that the roots of the Hurwitz polynomial $p(s) = s^n + k_1 s^{n-1} + \cdots + k_n$ are all in the open left-half plane. Furthermore, (5) can be overwritten as

$$\dot{\mathbf{e}} = A\mathbf{e}, \tag{6}$$

where $A = \begin{bmatrix} 0 & 1 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \\ -k_n & -k_{n-1} & -k_{n-2} & \cdots & -k_1 \end{bmatrix}$ is a stable matrix. By choosing Lyapunov, it represents as $V_e = 1/2\mathbf{e}^T P\mathbf{e}$ and assumes that the positive-definite matrices $P = P^T > 0$ and $Q = Q^T > 0$ satisfy the following:

$$A^T P + PA = -Q. \tag{7}$$

Then, taking the derivative of V_e along the trajectory of (6), the following can be obtained:

$$\begin{aligned} \dot{V}_e &= \frac{1}{2}\left(\mathbf{e}^T P\dot{\mathbf{e}} + \dot{\mathbf{e}}^T P\mathbf{e}\right) \\ &= \frac{1}{2}\mathbf{e}^T \left(PA + A^T P\right)\mathbf{e} \\ &= -\frac{1}{2}\mathbf{e}^T Q\mathbf{e} \\ &\leq -\frac{1}{2}\lambda_{\min}(Q)\|\mathbf{e}\|^2 \leq 0. \end{aligned} \tag{8}$$

On the basis of the Lyapunov theorem, the stability of this closed-loop system can be ensured with control law (3). And the tracking error e also can be astringented to zero. That is to say, the system state variable x will asymptotically approximate the desired trajectory x_d from any initial conditions, that is, $\lim_{t\to\infty} |\mathbf{e}| = 0$.

The control law of MBIC in (3) depends highly on the exact knowledge of the nonlinear functions $f(x)$ and $g(x)$ and external disturbance d of the nonlinear dynamics systems. So, precise parameters in the dynamic model in (1) have to be known. However, in many practical engineering projects, the perfect model of the system is difficult to obtain and external disturbances are impossible to ignore or impossible to measure directly or to obtain a precise mechanism model for them. In an effort to solve the problem of unknown nonlinearly parameterized $f(x)$ and $g(x)$, adaptive control schemes employing function approximation techniques have been studied in [15, 18, 21, 27]. In these approaches, the nonlinear functions $f(x)$ and $g(x)$, related to the system dynamics, are usually approximated by estimated function $\widehat{f}(\mathbf{x}, \widehat{W})$ and $\widehat{g}(\mathbf{x}, \widehat{W})$ with neural networks or fuzzy systems, respectively. The parameter $\widehat{W}$ denotes the estimated weights. Therefore, additional precautions are necessary to be made for avoiding possible singularities of the control action, that is, $\widehat{g}(\mathbf{x}, \widehat{W}) \neq 0$. In order to solve this problem, the initial values of the NN weights are chosen sufficiently close to the ideal values in [27]. Hence, offline training phases are needed before the controller is put into operation.

Remark 1. The MBIC method is unattractive to industry applications since the exact knowledge of system dynamics and lumped disturbances is hard to be obtained. But the MBIC method could not be neglected in designing the controller for uncertain nonlinear systems, as it is readily understood with good performance. An MBIC method with an appropriate compensative controller, proposed in [24], is valid for controlling uncertain nonlinear MIMO system with uncertainties and disturbances.

2.2. Description of RBFN. The radial basis function neural network (RBFNN) with Gaussian activation functions is the

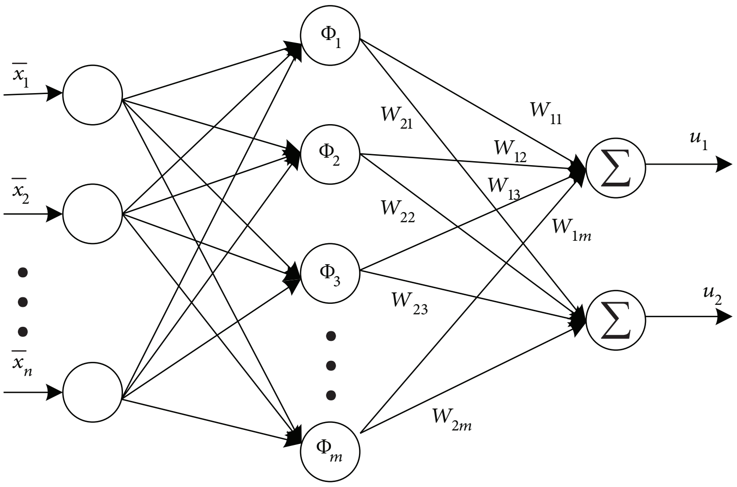


Figure 1: The structure of a radial basis function network in three-layers.

most popular type of artificial neural network architectures. The RBFNN with desirable features of local adjustment of the weight and mathematical tractability has been successfully applied to various issues [17, 21, 22, 30–32]. And RBFNN has been proven that, for any given real continuous function, there exists an RBFNN that can uniformly approximate over a compact set with arbitrary accuracy. A schematic diagram of a simple type of RBFNN with three-layer is described in Figure 1.

The RBFNN is a kind of feed-forward neural networks, which forms mappings from an input vector x to an output vector u. From Figure 1, the structure of the employed RBFNN, with n inputs, two outputs, and m hidden units, can be described by

$$\Phi_i(\bar{x}) = \exp\left(-\frac{\|\bar{x}-c_i\|^2}{\sigma_i^2}\right), \quad i = 1, 2, \ldots, m, \qquad (9)$$

where $\bar{\mathbf{x}} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n]^T \in R^n$ is the input vector, $\mathbf{\Phi} = [\Phi_1, \Phi_2, \ldots, \Phi_m]^T$ represents Gaussian activation function of the hidden layer, and $\mathbf{u} = [u_1, u_2]^T$ is the RBFNN controller output. The parameter $W_{i,j}$ is the weight, which connects the jth output unit with the ith radial basis function, c^i is defined as the central values of the ith hidden layer node, and $\sigma_i > 0$ means the radius of Gaussian function, respectively. It can be seen that each hidden node in the RBFNN computes an output that depends on a radially symmetric function, and the sum of the hidden layer outputs export to three-layer with a linear weighted. It is common that stronger output can be obtained when the input is nearer at the centroid of the node.

It has been already proved that, under mild assumptions, the RBFNN can approximate any continuous function over a compact set to any degree of accuracy.

Assumption 5. The output of the neural network $\hat{u}(x, W)$ is continuous with respect to its arguments for all finite (x, W).

Assumption 6. For an arbitrarily small positive number ε_0, there is an optimal neural network output $\hat{u}(x, W)$ that makes max $\|\hat{u}(x, W^*) - u\| \leq \varepsilon_0$.

Theoretical and numerical studies show that performance of RBFNN is highly dependent on the locations of centers. The structure of the RBFNN will be simpler with the less number of second layer. Unfortunately, satisfactory performance would not be easily obtained. On the contrary side, the identification precision is higher with the bigger number of hidden nodes, but the structure of RBFNN will be very complicated. So, we choose the nearest neighbor clustering algorithm to calculate the hidden nodes to fix the structure of the neural network, which is ignored here, similar to the authors' the previous studies [17, 21, 22]. The detailed process of this learning algorithm is also proposed in these researches.

3. Design of the RBFNN-Based Direct Inverse Control Scheme

Following the above conclusions in Section 2, the inverse controller based on (3) cannot be acted, since the parameter variation and external disturbances is hard to get directly. According to the superiorities of RBFNN, (3) can be realized online and adaptively by the RBFNN, which can be represented as (10). The structure of the novel neural network using direct inverse control system is shown in Figure 2.

$$\begin{aligned}
u &= g^{-1}(\mathbf{x})(r - f(\mathbf{x}) - d) \\
&= g^{-1}(\mathbf{x})(r - f(\mathbf{x})) - g^{-1}(\mathbf{x})d \\
&= u_f - u_c \\
&= \hat{u}_f(\bar{x}, W_1) - \hat{u}_c(\bar{x}, W_2) \\
&= \sum_{i=1}^{m} W_{1i}\Phi_i(\bar{x}) - \sum_{i=1}^{m} W_{2i}\Phi_i(\bar{x}) \\
&= \widehat{W}_1^T \Phi(\bar{x}) - \widehat{W}_2^T \Phi(\bar{x}).
\end{aligned} \qquad (10)$$

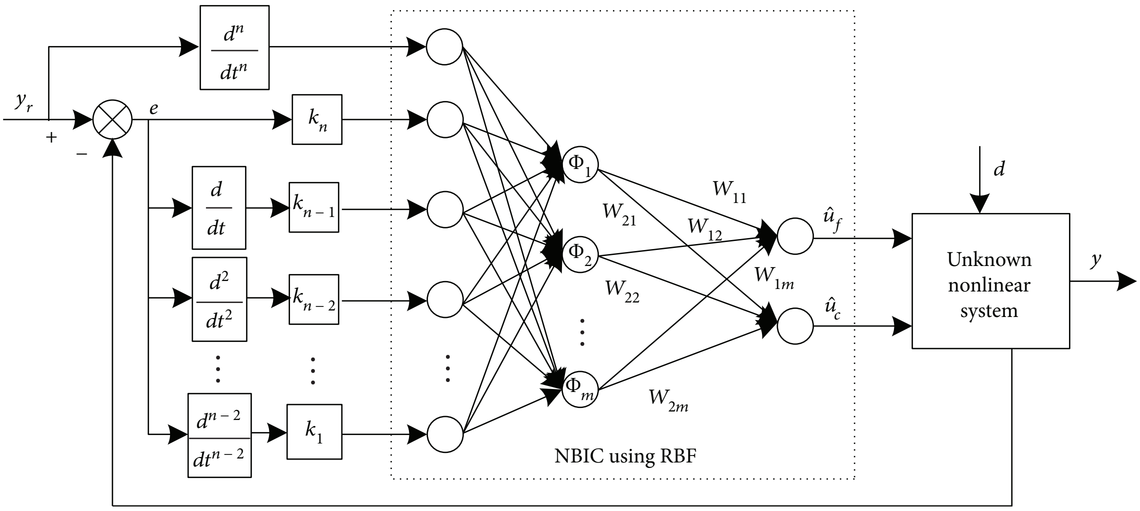


Figure 2: The structure of neural network-based direct inverse controller using RBF.

From (10) and Figure 2, we can see that the unknown nonlinear function u_f and u_c are parameterized by an RBFNN with output $\hat{u}_f(\bar{x}, W_1)$ and $\hat{u}_c(\bar{x}, W_2)$, respectively. And the matrices of adjustable weights are described by the parameters W_1 and W_2. By calling up the NN theory and Assumption 6, on a compact set $M_x \subset R^n$, for every $\varepsilon > 0$, there exists a Gaussian basis function $\Phi(\bar{x})$ and two weights vectors $\widehat{W}_1 \subset \Omega_1$ and $\widehat{W}_2 \subset \Omega_2$ such that

$$\left|u_f - \widehat{W}_1^T \Phi(\bar{x})\right| < \varepsilon, \left|u_c - \widehat{W}_2^T \Phi(\bar{x})\right| < \varepsilon. \qquad (11)$$

The compact sets of $\Omega_1 \in \{\widehat{W}_1 : \|\widehat{W}_1\| \le M\}$ and $\Omega_2 \in \{\widehat{W}_2 : \|\widehat{W}_2\| \le M\}$ are the known subsets of R^m. Therefore, the optimal weights W_1^* and W_2^*, which can minimize the functions $|u_f - \widehat{W}_1^T \Phi(\bar{x})|$ and $|u_c - \widehat{W}_2^T \Phi(\bar{x})|$, can be defined as

$$\begin{aligned} W_1^* &= \arg \min_{\widehat{W}_1 \in \Omega_1} \left[\sup_{\bar{x} \in M_x} \left| u_f - \hat{u}_f\left(\bar{x}, \widehat{W}_1\right) \right| \right], \\ W_2^* &= \arg \min_{\widehat{W}_2 \in \Omega 2} \left[\sup_{\bar{x} \in M_x} \left| u_c - \hat{u}_c\left(\bar{x}, \widehat{W}_2\right) \right| \right]. \end{aligned} \qquad (12)$$

The parameters of weight coefficient matrix $\widehat{W}_1$ and $\widehat{W}_2$ are estimated by W_1^* and W_2^*. M_x and M represent a compact set of system states and a design parameter, respectively. The parameters W_1^* and W_2^* are constrained by $\|W_1^*\| \le W_{1\ \max}$ and $\|W_2^*\| \le W_{2\ \max}$.

The parameter η is described as the optimal estimation error of neural network, that is,

$$\eta = u - \hat{u}\pi(\bar{x}, W^*). \qquad (13)$$

The estimation error η is bounded by a finite constant η_0, that is, $\eta_0 = \sup \|u - \hat{u}(\bar{x}, W^*)\|$. The error between (10) realize by the RBFNN and the ideal control law (3) can be described as

$$\begin{aligned} u - \hat{u}(\bar{x}, W) &= u_f - u_c - \hat{u}_f\left(\bar{x}, \widehat{W}_1\right) + \hat{u}_c\left(\bar{x}, \widehat{W}_2\right) \\ &= u_f - \hat{u}_f(\bar{x}, W_1^*) + \hat{u}_f(\bar{x}, W_1^*) - \hat{u}_f\left(\bar{x}, \widehat{W}_1\right) \\ &\quad - (u_c - \hat{u}_c(\bar{x}, W_2^*)) + \hat{u}_c(\bar{x}, W_2^*) - \hat{u}_c\left(\bar{x}, \widehat{W}_2\right) \\ &= \eta_1 + \widetilde{W}_1^T \Phi(\bar{x}) - \eta_2 - \widetilde{W}_2^T \Phi(\bar{x}) \\ &= \eta + \widetilde{W}_1^T \Phi(\bar{x}) - \widetilde{W}_2^T \Phi(\bar{x}), \end{aligned} \qquad (14)$$

where $\widetilde{W}_1 = W_1^* - \widehat{W}_1$, $\widetilde{W}_2 = W_2^* - \widehat{W}_2$, and $\eta = \eta_1 - \eta_2$. So, the designed control law of the proposed NBIC can be obtained as

$$u - \hat{u}(\bar{x}, W_1, W_2) = u + \eta + \widetilde{W}_1^T \Phi(\bar{x}) - \widetilde{W}_2^T \Phi(\bar{x}). \qquad (15)$$

The output of control law (15) is applied to (1), so the closed-loop system can be described as

$$\begin{aligned} \dot{x}_n &= f(x) + g(x)\left(u + \eta + \widetilde{W}_1^T \Phi(\bar{x}) - \widetilde{W}_2^T \Phi(\bar{x})\right) + d \\ &= f(x) + g(x) \frac{r - f(x) - d}{g(x)} \\ &\quad + g(x)\left(\eta + \widetilde{W}_1^T \Phi(\bar{x}) - \widetilde{W}_2^T \Phi(\bar{x})\right) + d \\ &= r + g(x)\left(\eta + \widetilde{W}_1^T \Phi(\bar{x}) - \widetilde{W}_2^T \Phi(\bar{x})\right). \end{aligned} \qquad (16)$$

Let $B = [0 \quad 0 \quad \cdots \quad 1]^T$. The closed-loop system (16) can be rewritten as follows combining with (4), (6), and (16):

$$\dot{\mathbf{e}} = A\mathbf{e} + B\left[g(x)\left(\eta + \widetilde{W}_1^T \Phi(\bar{x}) - \widetilde{W}_2^T \Phi(\bar{x})\right)\right]. \qquad (17)$$

Although the imperfect neural network controller will generally lead to degradation of the tracking performance, the system can possess acceptable performance. If the neural network controller can approximate the ideal controller of the system, that is, the controller estimation $\eta \to 0$ and the parameter estimation error $\tilde{W}_1 \to 0$ and $\tilde{W}_2 \to 0$, the tracking error can be achieved to zero, that is, $e \to 0$ as $t \to \infty$.

3.1. The Stability Analysis. To synthesize an adaptive NBIC with convergence capability, guaranteed stability, and disturbance attenuation, it is first necessary to ensure the chosen architecture of the RBFNN is capable. It can be seen from the theoretical and numerical studies that the performance of RBFNN highly depends on the locations of centers c^i. The radius σ_i determines the number of the clusters and affects the learning speed and accuracy. So, the choice of appropriate c^i and σ_i is particularly significant. In order to make it easier to analyze, parameters c^i and σ_i of RBFNN are learned by the nearest neighbor clustering algorithm and then kept fixed, since the weight matrix parameters and the structure of RBFNN can be adjusted by the nearest neighbor clustering algorithm, synchronously. The next work is to determine the adaptive weights of the NBIC according to Lyapunov stability analysis. And the Lyapunov function can be given as the follows:

$$V = \frac{1}{2}\mathbf{e}^T P\mathbf{e} + \frac{1}{2\gamma_1} tr\left(\tilde{W}_1^T \tilde{W}_1\right) + \frac{1}{2\gamma_2} tr\left(\tilde{W}_2^T \tilde{W}_2\right), \tag{18}$$

where $\gamma_1, \gamma_2 > 0$ is the learning rate of RBFNN. Assume **A** is a stable matrix, so there are positive-definite matrices $P = P^T > 0$ and $Q = Q^T > 0$ satisfy the following

$$A^T P = PA = -Q. \tag{19}$$

Along the trajectories of (17), the differential of Lyapunov function $\dot{V}$ can be obtained as follows:

$$\begin{aligned}
\dot{V} &= \frac{1}{2}\left(\mathbf{e}^T P\dot{\mathbf{e}} + \dot{\mathbf{e}}^T P\mathbf{e}\right) + \frac{1}{\gamma_1} tr\left(\dot{\tilde{W}}_1^T \tilde{W}_1\right) + \frac{1}{\gamma_2} tr\left(\dot{\tilde{W}}_2^T \tilde{W}_2\right)\\
&= \frac{1}{2}\left[\left(\mathbf{e}^T P\left(A\mathbf{e} + B\left(\eta + \tilde{W}_1^T\Phi(\bar{x}) - \tilde{W}_2^T\Phi(\bar{x})\right)\right)\right) + \left(\mathbf{e}^T A^T + \left(\eta + \tilde{W}_1^T\Phi(\bar{x}) - \tilde{W}_2^T\Phi(\bar{x})\right)^T B^T P\mathbf{e}\right)\right]\\
&\quad + \frac{1}{\gamma_1} tr\left(\dot{\tilde{W}}_1^T \tilde{W}_1\right) + \frac{1}{\gamma_2} tr\left(\dot{\tilde{W}}_2^T \tilde{W}_2\right)\\
&= \frac{1}{2}\left[\mathbf{e}^T\left(PA + A^T P\right)\mathbf{e} + \mathbf{e}^T PB\eta + \mathbf{e}^T PB\tilde{W}_1^T\Phi(\bar{x}) - \mathbf{e}^T PB\tilde{W}_2^T\Phi(\bar{x}) + \eta^T B^T P\mathbf{e} + \Phi^T(\bar{x})\tilde{W}_1 B^T P\mathbf{e} - \Phi^T(\bar{x})\tilde{W}_2 B^T P\mathbf{e}\right]\\
&\quad + \frac{1}{\gamma_1} tr\left(\dot{\tilde{W}}_1^T \tilde{W}_1\right) + \frac{1}{\gamma_2} tr\left(\dot{\tilde{W}}_2^T \tilde{W}_2\right)\\
&= -\frac{1}{2}\mathbf{e}^T Q\mathbf{e} + \eta^T B^T P\mathbf{e} + \Phi^T(\bar{x})\tilde{W}_1 B^T P\mathbf{e} - \Phi^T(\bar{x})\tilde{W}_2 B^T P\mathbf{e} + \frac{1}{\gamma_1} tr\left(\dot{\tilde{W}}_1^T \tilde{W}_1\right) + \frac{1}{\gamma_2} tr\left(\dot{\tilde{W}}_2^T \tilde{W}_2\right).
\end{aligned} \tag{20}$$

Noting that $P^T = P, \Phi^T(\bar{x})\tilde{W}_1 B^T P\mathbf{e} = tr(B^T P\mathbf{e}\Phi^T(\bar{x})\tilde{W}_1)$, and $\Phi^T(\bar{x})\tilde{W}_2 B^T P\mathbf{e} = tr(B^T P\mathbf{e}\Phi^T(\bar{x})\tilde{W}_2)$, so the differential Lyapunov function $\dot{V}$ can be re-described by

$$\begin{aligned}
\dot{V} = &-\frac{1}{2}\mathbf{e}^T Q\mathbf{e} + \eta^T B^T P\mathbf{e} + \frac{1}{\gamma_1} tr\left(\gamma_1 B^T P\mathbf{e}\Phi^T(\bar{x})\tilde{W}_1 + \dot{\tilde{W}}_1^T \tilde{W}_1\right)\\
&+ \frac{1}{\gamma_2} tr\left(\gamma_2 B^T P\mathbf{e}\Phi^T(\bar{x})\tilde{W}_2 + \dot{\tilde{W}}_2^T \tilde{W}_2\right).
\end{aligned} \tag{21}$$

There is $\dot{\tilde{W}} = -\dot{\hat{W}}$, if the weights adaptive update law of the RBFNN is chosen as

$$\begin{aligned}
\dot{\hat{W}}_1 &= \gamma_1\left(\Phi(\bar{x})\mathbf{e}^T PB - \|\mathbf{e}\|\hat{W}_1\right),\\
\dot{\hat{W}}_2 &= \gamma_2\Phi(\bar{x})\mathbf{e}^T PB.
\end{aligned} \tag{22}$$

Substituting (22) into (21) yields

$$\dot{V} = -\frac{1}{2}\mathbf{e}^T Q\mathbf{e} + \eta^T B^T P\mathbf{e} + \|\mathbf{e}\| tr\left(\tilde{W}_1^T \hat{W}_1\right). \tag{23}$$

According to the characteristic of the F norm in the preliminaries, (23) can be changed as follows:

$$\begin{aligned}
\dot{V} &= -\frac{1}{2}\mathbf{e}^T Q\mathbf{e} + \eta^T B^T P\mathbf{e} + \|\mathbf{e}\| tr\left(\tilde{W}_1^T\left(W_1^* - \tilde{W}_1\right)\right)\\
&\le -\frac{1}{2}\mathbf{e}^T Q\mathbf{e} + \eta^T B^T P\mathbf{e} + \|\mathbf{e}\|\|\tilde{W}_1\|_F\|W_1^*\|_F - \|\mathbf{e}\|\|\tilde{W}_1\|_F^2\\
&\le -\frac{1}{2}\lambda_{\min}(Q)\|\mathbf{e}\|^2 + \eta_0\lambda_{\max}(P)\|\mathbf{e}\| + \|\mathbf{e}\|\|\tilde{W}_1\|_F\|W_1^*\|_F\\
&\quad - \|\mathbf{e}\|\|\tilde{W}_1\|_F^2\\
&\le -\|\mathbf{e}\|\left(\frac{1}{2}\lambda_{\min}(Q)\|\mathbf{e}\| - \|\tilde{W}_1\|_F W_{1\ \max} + \|\tilde{W}_1\|_F^2 - \eta_0\lambda_{\max}(P)\right)\\
&\le -\|\mathbf{e}\|\left(\frac{1}{2}\lambda_{\min}(Q)\|\mathbf{e}\| + \left(\|\tilde{W}_1\|_F - \frac{W_{1\ \max}}{2}\right)^2\right.\\
&\quad \left. - \frac{1}{4}W_{1\ \max}^2 - \eta_0\lambda_{\max}(P)\right).
\end{aligned} \tag{24}$$

We can see that, in order to make $\dot{V} < 0$, the following inequality is presented

$$\frac{1}{2}\lambda_{\min}(Q)\|\mathbf{e}\| \geq \frac{1}{4}W_{1\ \max}^{2} + \eta_0\lambda_{\max}(P), \tag{25}$$

or

$$\left(\|\tilde{W}_1\|_F - \frac{W_{1\ \max}}{2}\right)^2 \geq \frac{1}{4}W_{1\ \max}^{2} + \eta_0\lambda_{\max}(P). \tag{26}$$

We can get

$$\|\mathbf{e}\| \geq \frac{2}{\lambda_{\min}(Q)}\left(\eta_0\lambda_{\max}(P) + \frac{1}{4}W_{1\ \max}^{2}\right), \tag{27}$$

or

$$\|\tilde{W}_1\|_F \geq \sqrt{\frac{1}{4}W_{1\ \max}^{2} + \eta_0\lambda_{\max}(P)} + \frac{W_{1\ \max}}{2} \tag{28}$$

The closed-loop control system is supposed to be globally stable, since all the variables of the RBFNN are bounded. Better track performance can be received from (27) when the characteristic value of Q is larger, and the smaller upper-bound η_0 of the estimation error of the RBFN, the eigenvalues of P and the $W_{1\ \max}$. That is to say, the convergence problem of the weights of neural network is solvable, and bounded neural network weights ensure bounded control input.

3.2. Design Procedure of Neural Network-Based Direct Inverse Controller with Disturbance Rejection. The design steps about the proposed neural network-based direct inverse controller (NBIC) with disturbance rejection are summarized as follows:

Step 1. Specify the design parameters of the NBIC.

Specify $k_1, k_2, \ldots, k_n$ to get all roots in the left-half plane for the polynomial $s^n + k_1 s^{n-1} + \cdots + k_n = 0$, solve the equation $-Q = A^T P + PA$ to obtain a symmetric matrix P based on a positive-definite matrix Q, and choose the RBF neural network, specifying the inputs and the initial weights of RBFNN.

Step 2. Structure the proposed method.

Considering the characteristics of the PMSM, choose a small value as the central value of the hidden layer node c^i and an appropriate number of the cluster $\sigma_i = 5$. Furthermore, calculate the hidden node parameters $c_i, \sigma, i = 1, \ldots, m$ based on the nearest neighbor clustering algorithm and then obtain the structure of RBFNN. Calculate the hidden node output vector Φ for input x, based on (9); get the approximated control law $\hat{u}$ from (10).

Step 3. Adapt the weight parameters.

Apply the control law (10) to the dynamic system expressed by (1). Use (22) to train the weight parameters W_1 and W_2 of RBFN, with the initial weight matrix parameters $W_{i,j}$ for subtle differences.

4. Position Tracking Control of PMSM Servo System with the Proposed NBIC Method

4.1. Control Design for PMSM Based on NBIC Method. In this section, a numerical simulation example will be performed to evaluate the effectiveness and applicability of the proposed RBFNN-based direct inverse control method. The dynamic equation of PMSM servo system can be described as

$$\begin{gathered}\begin{bmatrix}\dot{\theta}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -\dfrac{B}{J}\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix} + \begin{bmatrix}0\\ \dfrac{K_t}{J}\end{bmatrix} i_q^* + \begin{bmatrix}0\\ \dfrac{1}{J}\end{bmatrix} T_L,\\ y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \omega\end{bmatrix},\end{gathered} \tag{29}$$

where the coefficient K_t means torque constant, ω represents the angular velocity, and θ refers to rotor position. The parameters B and J are the viscous friction coefficient and the moment of inertia which are always vary with the different working conditions, i_q^* is the desired current input, and T_L denotes load torque which can be seen as the external disturbances. The following state space can be obtained with $x_1 = \theta$, $x_2 = \omega$, $u = i_q^*$, and $d = (1/J)T_L$:

$$\begin{aligned}\dot{x}_1 &= x_2,\\ \dot{x}_2 &= -\frac{B}{J}x_2 + \frac{K_t}{J}u + d,\\ y &= x_1.\end{aligned} \tag{30}$$

PMSM servo systems always confront load disturbances, friction force, and parameter uncertainties in some real industrial applications. The performance of the whole system will be degraded by these variations, disturbances, and uncertainties. Moreover, the control performance usually cannot be guaranteed with the fix control parameters. So, the adaptive position tracking control based on the proposed NBIC can be realized with as follows:

$$\begin{aligned}i_q^* &= K_t^{-1}\left(Jr + B\dot{\theta}\right) - K_t^{-1}d\\ &= \widehat{W}_1^T\Phi(x) - \widehat{W}_2^T\Phi(x),\\ r &= \ddot{\theta}_r - k_1\dot{e} - k_2 e\end{aligned} \tag{31}$$

where the parameter θ_r represents the reference position, and the position error between reference position and feedback position of the closed-loop system is $e = \theta_r - \theta$. The details of the proposed method for PMSM position control can be seen from Figure 3.

4.2. Simulation Comparison Results. Simulations on PMSM system have been performed to show the effectiveness of the proposed control method. For the purpose of comparison, two control methods, that is, model-based inverse controller (MBIC) and neural network-based adaptive direct inverse controller (NBIC), are applied for the position tracking control of PMSM servo system in the case of

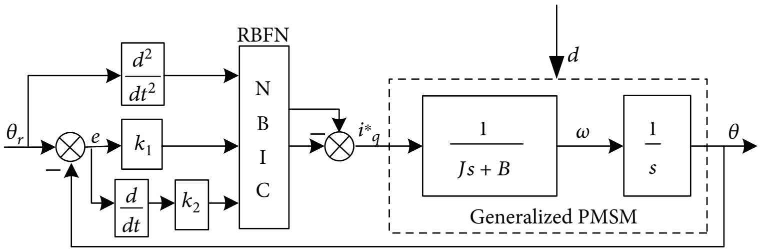


Figure 3: The PMSM position tracking control diagram based on the proposed NBIC.

Table 1: Nomenclature.

Symbol	Description (unit)	Symbol	Description (unit)
$R^{m\times n}$	$m\times n$ real matrix space	$tr(\bullet)$	Trajectory of the matrix$[\bullet]$
R	Real number set	$\|\bullet\|$	Euclidean norm
$\lambda_{\min}(P)$	The minimum eigenvalue of matrix	$\|W\|_F$	Frobenius norm (F norm)
$\lambda_{\max}(P)$	The maximum eigenvalue of matrix	u	Input
x	State vector of (1)	y	Output
R^n	Unknown continuous functions including internal uncertainties	y_d, x_d	Bounded reference trajectory
d	External disturbances	a	Arbitrarily nonnegative constant
$f(x)$	Approximated by estimated function $\hat{f}(\mathbf{x},\hat{W})$ with neural networks systems	$g(x)$	Approximated by estimated function $\hat{g}(\mathbf{x},\hat{W})$ with neural networks or fuzzy systems
$\hat{W}$	Estimated weights	$\bar{\mathbf{x}}$	Input vector
$\mathbf{u}$	RBFNN controller output	$\mathbf{\Phi}$	Gaussian activation function of the hidden layer
$W_{i,j}$	Weight	c^i	Central values of the ith hidden layer node
ε_0	Arbitrarily small positive number	$\hat{u}$	Output of the neural network
σ_i	Number of the clusters	η	Estimation error
ω	Angular velocity	K_t	Torque constant (Nm/A)
B	Viscous friction coefficient (Nms/rad)	θ	Rotor position (rad)
i_q^*	Current input	J	Inertia (kg·m^2)
θ_r	Reference position	T_L	Load torque (N·m)

nominal model and the case under load disturbance and parameter variations, respectively.

The parameters of PMSM are given as rated torque $T_L = 2.4\,\mathrm{N\cdot m}$, torque constant $K_t = 2.412\,\mathrm{Nm/A}$, moment of inertia $J_n = 1.068\times10^{-3}\,\mathrm{kg\cdot m^2}$, and viscous coefficient $B = 7.4\times10^{-5}\,\mathrm{Nms/rad}$. The reference position is given as $\theta_r = 0.5\sin(t)$ rad. For the sake of fair comparison, the control inputs of the two algorithms have the same control parameters. The sampling interval of the control processing in the simulations is set at 0.001 s. The controller parameters k_1 and k_2 are selected as $k_1 = 5.5$ and $k_2 = 0.5$, respectively. So the system matrix $A = \begin{bmatrix} 0 & 1 \\ -0.5 & -5.5 \end{bmatrix}$ is the Hurwitz matrix, further choosing $Q = \begin{bmatrix} 8 & 0 \\ 0 & 8 \end{bmatrix}$, $P = \begin{bmatrix} 13.3467 & 0.04 \\ 0.04 & 0.0135 \end{bmatrix}$, and $\gamma = 5$. The initial structure of the RBFNN is chosen as 3–4–2. The input layer vector is chosen as $[\ddot{\theta}_r, e, \dot{e}]^T$, the function of hidden layer is defined based on (9), the initial values of the weight matrices are selected as $W_1(0) = 0$ and $W_2(0) = 0$, and the hidden nodes and the radius of the RBFNN are decided by the nearest neighbor clustering algorithm, that is, $m = 4$, $\sigma = 4.5$, and $c = 0.1$. A nomenclature summarizing all the symbols throughout this brief is furnished in Table 1.

There are many literatures that show the superiority with respect to a standard approach, such as PID method and MPC. The RBFNN-based control methods with the nearest neighbor clustering algorithm, in the authors' previous studies [17, 21 and 22], verify that it permits to gain in terms of performance with respect to a PID-based method. Compared with the traditional control methods, the effectiveness of the MBIC method is also proposed in many literatures. Considering these facts, the effectiveness of the proposed control method is only supported by a comparison with the traditional model-based inverse controller (MBIC). The

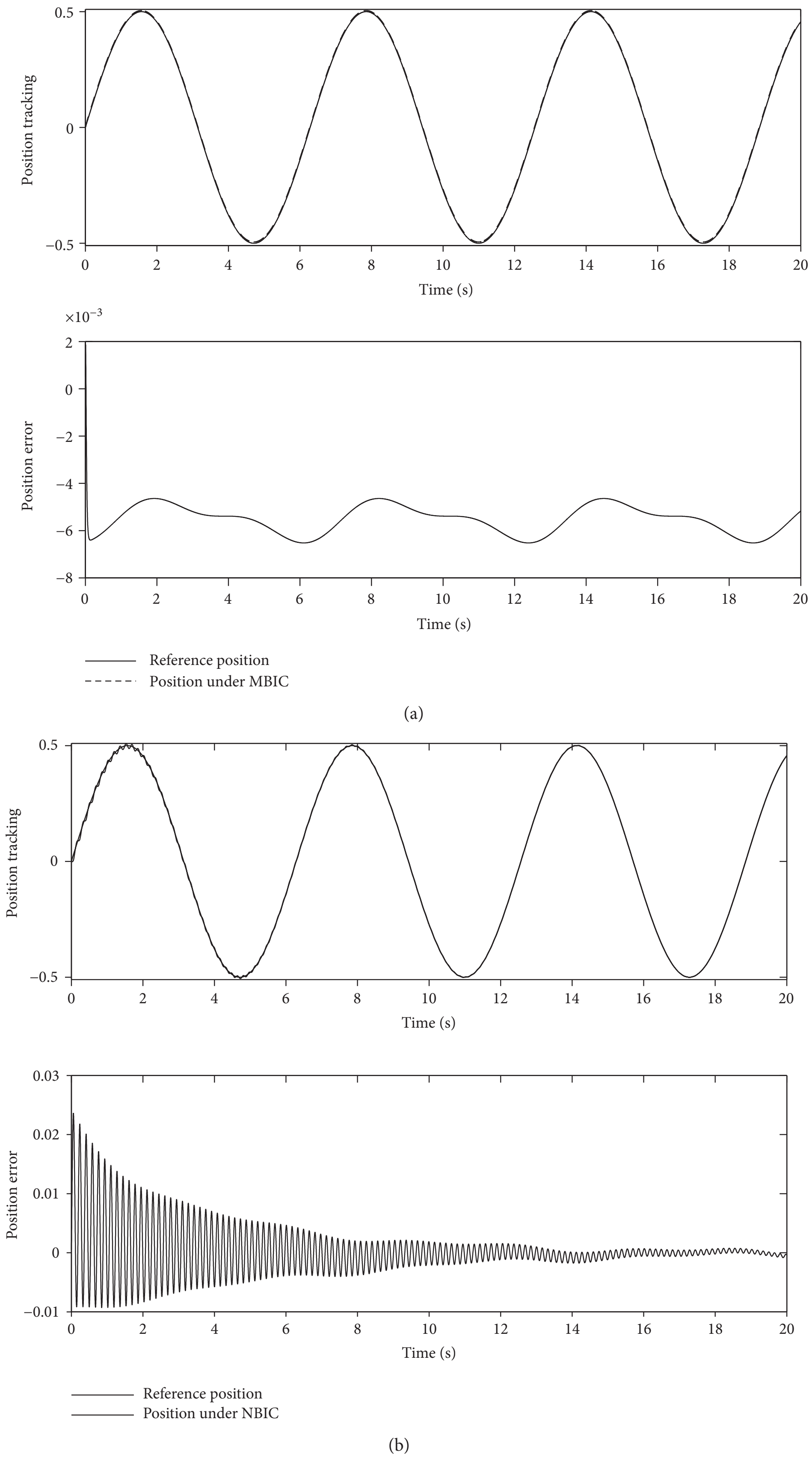
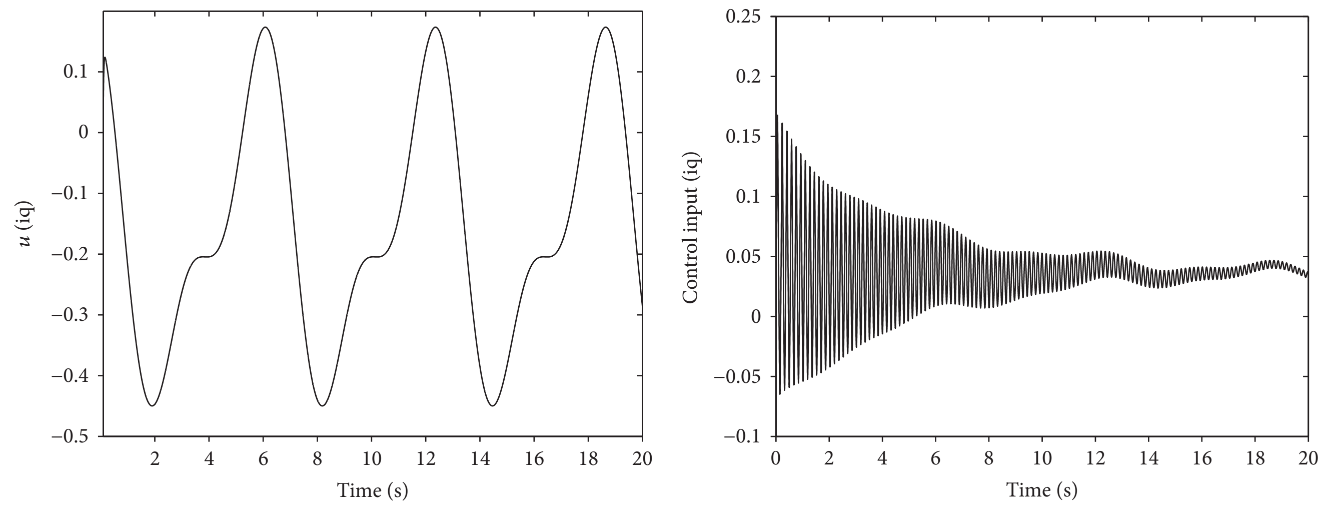


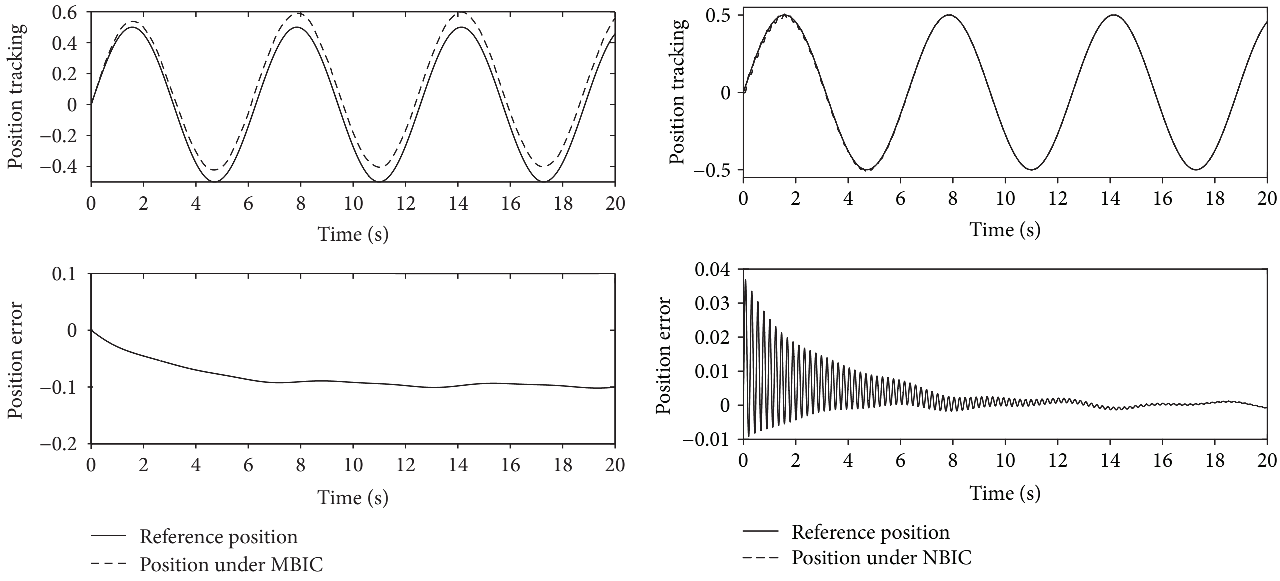
Figure 4: Position tracking and error performance in the nominal model case: (a) under the MBIC method and (b) under the NBIC method.



(a)

(b)

Figure 5: The output values of different control methods in the nominal model case: (a) the control output of MBIC and (b) the control output of NBIC.



(a)

(b)

Figure 6: Position tracking and error performance in case II: (a) the MBIC method and (b) the NBIC method.

NBIC based on RBFNN is training by the nearest neighbor clustering algorithm with clustering radius, which can ensure the real-time performance of the control method. So the position tracking and error performance in the cases with nominal model without any disturbance and with load disturbances and parameter variations are shown in this paper under MBIC and NBIC, respectively.

4.2.1. Case I: Comparison Results under Nominal Model. The position tracking performance is tested under no load disturbances and parameter variations in this part, that is, in case of a nominal model.

Figure 4 shows the position tracking and error performance in the nominal model case under the MBIC and NBIC methods, respectively. Figure 5 gives the output curves of the controllers MBIC and NBIC under the nominal model case, respectively. From Figure 4, we can see that both methods have good dynamic position tracking and error performances. However, obviously, the position tracking error of the NBIC in Figure 4(b) is a little bigger than the MBIC in Figure 4(a) at the beginning, since the initial network approximations may be quite poor during the early stages of learning. But due to the characteristics of the neural network, the position tracking error of the NBIC will be close to zero which is much better than the MBIC method. It is can be indicated from Figure 5(b) that the output of the MBIC asymptotically converged to a small region because of the learning ability of the neural

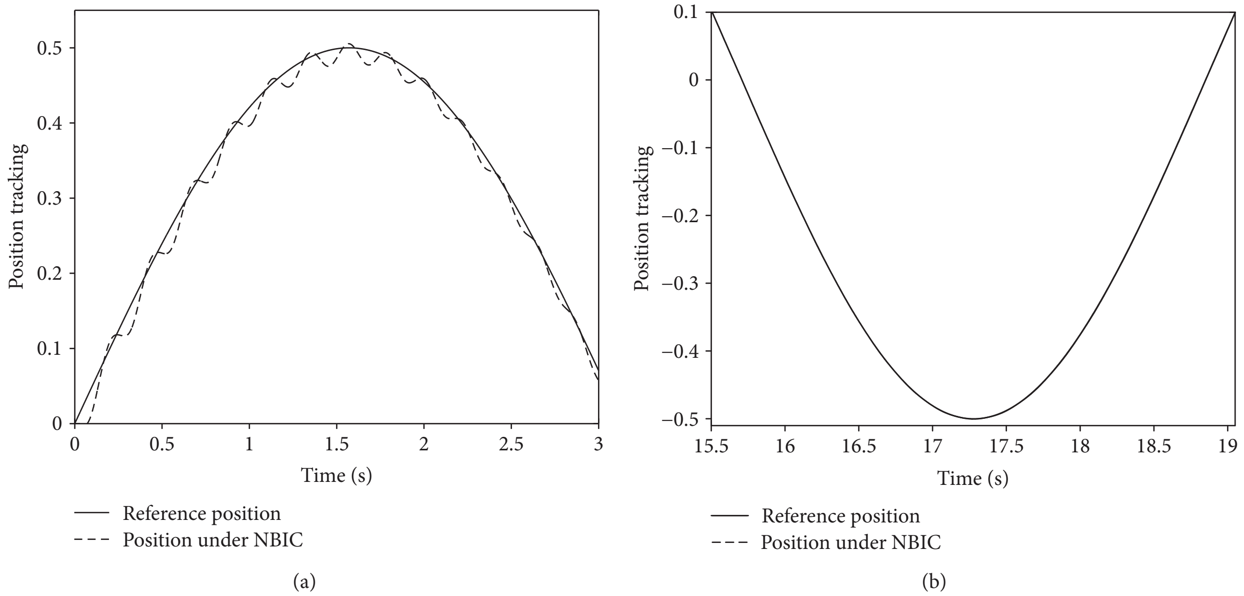


FIGURE 7: Partial enlargement of position tracking under the NBIC: (a) enlargement at the beginning and (b) enlargement at the end.

network. The NBIC controller has the satisfactory guaranteed stability and convergence capability.

4.2.2. Case II: Performance Comparisons with Load Disturbances and Parameter Variations. The position tracking performance is tested under unknown load disturbances T_L and parameter variations in inertia J and viscous coefficient B, which are supposed to happen on PMSM at the very beginning. The system response of position tracking and its errors are shown in Figure 6. In order to clarify the comparison performances, the partial enlargement of the position tracking for Figure 6 is represented in Figure 7. The controller outputs i_q^* are the same as those in Figure 5.

The position tracking performance of PMSM using the MBIC is affected considerably by the load disturbances and parameter variations based on Figure 6(a). It can be concluded that superior robustness performance in such case of uncertainties can be obtained with the proposed NBIC method from Figure 6(b). At the beginning, we can see from Figure 7(a) that the tracking performance is affected a little by the case of parameters and load variations. As times goes on, the NBIC makes the error between the position and the reference very close to zero due to the self-learning and adaptive ability of the RBFNN. The results can be indicated from Figure 7(b). The reason is that the parameter uncertainties and disturbances are modelled by the neural network at every time step and finally eliminated by feed forward channel.

5. Conclusion

Considering the adaptive self-learning ability of neural networks, an adaptive neural network-based direct inverse controller (NBIC) for a nonlinear system with uncertain parameters and unknown external disturbances is presented to achieve satisfactory tracking performance in this paper. The proposed NBIC is realized by one RBFNN which has two outputs, one output of the RBFNN acts as the main controller to handle the parameter uncertainties, and the other output of RBFNN is used to handle external disturbances. The problems of the uncertainties and the ability of the self-adaptive control can all be handled in one single neural network framework, since the accuracy of the system identification and the ability of antidisturbance can satisfy the requirements of the system. Moreover, the stable online weight matrices adjustment mechanism of the RBFNN is determined by the Lyapunov theory to achieve the stability and guarantee attenuation of the disturbances. Simulation results of position tracking for the PMSM servo system illustrate that the proposed method NBIC has the robust and effective control performance with good disturbance rejection.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant nos. 61773335 and 51405428), Natural Science Foundation of Jiangsu Province (Grant no. BK20171289), Natural Science Fund for Excellent Young Scholars of Yangzhou City (Grant no. YZ2017099), Opening Foundation of Jiangsu Engineering Research Center on Meteorological Energy Using and Control (Grant no. KCMEIC03), Opening Foundation of Ministry Education Key Laboratory of MCCSE (Grant nos. MCCSE2015A01 and MCCSE2016A01), and a Project Funded by the High Talent Support Program of Yangzhou University.

References

[1] J. Yang, W. H. Chen, S. Li, L. Guo, and Y. Yan, “Disturbance/uncertainty estimation and attenuation techniques in PMSM drives—a survey,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 4, pp. 3273–3285, 2017.

[2] G. J. Tan, H. Zhang, J. Li, Z. X. Ma, and R. W. Yu, “Design and implementation of robust controller for PMSM drive,” *International Review of Electrical Engineering-IREE*, vol. 7, no. 2, pp. 3769–3778, 2012.

[3] M. Van, S. S. Ge, and H. Ren, “Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control,” *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1681–1693, 2017.

[4] S. Y. Chen and S. S. Gong, “Speed tracking control of pneumatic motor servo systems using observation-based adaptive dynamic sliding-mode control,” *Mechanical Systems and Signal Processing*, vol. 94, pp. 111–128, 2017.

[5] S. E. Lyshevski, “Control of high-precision direct-drive mechatronic servos: tracking control with adaptive friction estimation and compensation,” *Mechatronics*, vol. 43, pp. 1–5, 2017.

[6] H. P. Ren and P. F. Gong, “Adaptive control of hydraulic position servo system using output feedback,” *Proceedings of the Institution of Mechanical Engineers Part I- Journal of Systems and Control Engineering*, vol. 231, no. 7, pp. 527–540, 2017.

[7] R. Mei, Q. X. Wu, and C. S. Jiang, “Robust adaptive backstepping control for a class of uncertain nonlinear systems based on disturbance observers,” *Science China Information Sciences*, vol. 53, no. 6, pp. 1201–1215, 2010.

[8] R. Li, M. Chen, and Q. X. Wu, “Adaptive neural tracking control for uncertain nonlinear systems with input and output constraints using disturbance observer,” *Neurocomputing*, vol. 235, pp. 27–37, 2017.

[9] H. K. Khalil, “High-gain observers in feedback control application to permanent magnet synchronous motors,” *IEEE Control Systems Magazine*, vol. 37, no. 3, pp. 25–41, 2017.

[10] H. X. Liu and S. H. Li, “Speed control for PMSM servo system using predictive functional control and extended state observer,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1171–1183, 2012.

[11] J. Yang and Z. Ding, “Global output regulation for a class of lower triangular nonlinear systems: a feedback domination approach,” *Automatica*, vol. 76, pp. 65–69, 2017.

[12] I. H. Kim and Y. I. Son, “A modular disturbance observer-based cascade controller for robust speed regulation of PMSM,” *Journal of Electrical Engineering & Technology*, vol. 12, no. 4, pp. 1663–1674, 2017.

[13] S. B. Wang, X. Ren, J. Na, and T. Zeng, “Extended-state-observer-based funnel control for nonlinear servomechanisms with prescribed tracking performance,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 98–108, 2017.

[14] J. Na, Q. Chen, X. Ren, and Y. Guo, “Adaptive prescribed performance motion control of servo mechanisms with friction compensation,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 486–494, 2014.

[15] H. Li, J. Wang, and P. Shi, “Output-feedback based sliding mode control for fuzzy systems with actuator saturation,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1282–1293, 2016.

[16] N. M. Ramli, M. A. Hussain, and B. M. Jan, “Multivariable control of a debutanizer column using equation based artificial neural network model inverse control strategies,” *Neurocomputing*, vol. 194, pp. 135–150, 2016.

[17] S. Li, J. Li, J. Qiu, H. Ji, and K. J. Zhu, “Control design for arbitrary complex nonlinear discrete-time systems based on direct NNMRAC strategy,” *Journal of Process Control*, vol. 21, no. 1, pp. 103–110, 2011.

[18] C. X. Mu, C. Sun, D. Wang, and A. Song, “Adaptive tracking control for a class of continuous-time uncertain nonlinear systems using the approximate solution of HJB equation,” *Neurocomputing*, vol. 260, pp. 432–442, 2017.

[19] A. U. Levin and K. S. Narendra, “Control of nonlinear dynamical systems using neural networks. II. Observability, identification, and control,” *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 30–42, 1996.

[20] S. Mukhopadhyay and K. S. Narendra, “Disturbance rejection in nonlinear systems using neural networks,” *IEEE Transactions on Neural Networks*, vol. 4, no. 1, pp. 63–72, 1993.

[21] J. Li, S. Li, and X. Chen, “Adaptive speed control of a PMSM servo system using an RBFN disturbance observer,” *Transactions of the Institute of Measurement and Control*, vol. 34, no. 5, pp. 615–626, 2012.

[22] J. Li, C. Wu, S. Li, J. Yang, and S. Li, “Optimal disturbance rejection control approach based on a compound neural network prediction method,” *Journal of Process Control*, vol. 24, no. 10, pp. 1516–1526, 2014.

[23] H. J. Rong, S. Suresh, and G. S. Zhao, “Stable indirect adaptive neural controller for a class of nonlinear system,” *Neurocomputing*, vol. 74, no. 16, pp. 2582–2590, 2011.

[24] J. Peng and R. Dubay, “Nonlinear inversion-based control with adaptive neural network compensation for uncertain MIMO systems,” *Expert Systems with Applications*, vol. 39, no. 9, pp. 8162–8171, 2012.

[25] Z. Wang, J. P. Yuan, Y. P. Pan, and D. Che, “Adaptive neural control for high order Markovian jump nonlinear systems with unmodeled dynamics and dead zone inputs,” *Neurocomputing*, vol. 247, pp. 62–72, 2017.

[26] C. G. Yang, Y. M. Jiang, Z. J. Li, W. He, and C. Y. Su, “Neural control of bimanual robots with guaranteed global stability and motion precision,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1162–1171, 2017.

[27] H. Li, Y. Pan, P. Shi, and Y. Shi, “Switched fuzzy output feedback control and its application to a mass–spring–damping system,” *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1259–1269, 2016.

[28] R. M. Sanner and J.-J. E. Slotine, “Gaussian networks for direct adaptive control,” *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 837–863, 1992.

[29] A. Isidori, *Nonlinear Control Systems*, Springer, New York, NY, USA, 1989.

[30] Q. Wang, K. F. He, W. Q. Qian, T. J. Zhang, Y. Q. Cheng, and K. Y. Wu, “Unsteady aerodynamics modeling for flight dynamics application,” *Acta Mechanica Sinica*, vol. 28, no. 1, pp. 14–23, 2012.

[31] J. Na, Y. Huang, X. Wu, G. Gao, G. Herrmann, and J. Z. Jiang, “Active adaptive estimation and control for vehicle suspensions with prescribed performance,” *IEEE Transactions on Control Systems Technology*, pp. 1–15, 2017.

[32] Y. B. Huang, J. Na, X. Wu, and G. Gao, “Approximation-free control for vehicle active suspensions with hydraulic actuator,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 7258–7267, 2018.