Output Feedback Neural Network Adaptive Robust Control With Application to Linear Motor Drive System

J. Q. Gong

Bin Yao
e-mail: byao@purdue.edu

School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907

In this paper, neural networks (NNs) and adaptive robust control design method are integrated to design a performance oriented control law with only output feedback for a class of single-input-single-output nth order nonlinear systems in a normal form. The nonlinearities in the system include repeatable unknown nonlinearities and nonrepeatable unknown nonlinearities such as external disturbances. In addition, unknown nonlinearities can exist in the control input channel as well. A high-gain observer is employed to estimate the states of system. All unknown but repeatable nonlinear functions are approximated by the outputs of multilayer neural networks with the estimated states as inputs to achieve a better model compensation. All NN weights are tuned on-line. In order to avoid possible divergence of on-line tuning, discontinuous projections with fictitious bounds are used in the weight adjusting law to make sure that all the weights are adapted within a prescribed range. Theoretically, the resulting controller achieves a guaranteed output tracking transient performance and a guaranteed final tracking accuracy in general. Certain robust control terms is then constructed to effectively attenuate various model uncertainties and estimate errors. Furthermore, if all the states are available and the unknown nonlinear functions are in the functional ranges of the neural networks, an asymptotic output tracking is also achieved to retain the perfect learning capability of NNs in the ideal situation provided that the ideal NN weights fall within the prescribed range. The output feedback neural network adaptive robust control is then applied to the control of a linear motor drive system. Experiments are carried out to show the effectiveness of the proposed algorithm and the excellent output tracking performance. [DOI: 10.1115/1.2199860]

Keywords: neural network, adaptive control, robust control, nonlinear systems, uncertainties

1 Introduction

In recent years, controller design for systems having complex nonlinear dynamics has become an increasingly important and challenging topic. Nonlinearities in physical systems may appear in various forms, hence, it is in general difficult to treat various nonlinearities under a unified framework. In some situations, to be worse, due to the limited knowledge about certain nonlinear physical phenomena (e.g., friction), it is also impossible to precisely describe the nonlinearities that can be used to capture those physical phenomena. All these factors make it difficult to design high performance controllers for nonlinear systems.

The appearance of neural networks (NNs) help us advancing the design of high performance controllers for general uncertain nonlinear systems considerably. Theoretically, as long as a sufficient number of neurons are employed, a neural network can approximate a continuous function to an arbitrary accuracy on any compact set [1–3]. Due to their universal approximation capability, neural networks can be used to model certain complex nonlinear physical phenomena effectively. It is thus of practical significance to use neural networks in nonlinear controller designs.

Neural networks have been applied to the control field recently [4] and various results have been achieved [5–9]. A survey of the application of NNs to control field was given in Ref. [4], where modeling, identification, and control of nonlinear systems via neural networks were discussed. There are two main issues to be dealt with in the use of neural networks for nonlinear control design. First, the ideal synaptic weights of a neural network for approximating an unknown nonlinear function are usually unknown. Certain algorithms have to be derived to tune these unknown NN weights on-line if NN is used to deal with various unknown nonlinear functions. In terms of control terminology, adaptation laws are needed. Second, the ideal NN weights for the neutral network to reconstruct an unknown nonlinear function exactly may not exist, i.e., the unknown nonlinear function to be approximated may not be in the functional range of the neural network. The approximation error between the ideal output of a neural network and the true nonlinear function cannot be assumed to be zero in general although it may be very small. Thus, the issue of robustness to the approximation errors needs to be considered when certain on-line tuning rules are derived for the NN weights. For applications in which only output is available for feedback, output feedback design [10–12] and certain observer design need to be employed. Unlike the cases in Refs. [13–15] where the structures of systems are known, nonlinear Luenberger-type observers cannot be applied here.

Recently, the adaptive robust control (ARC) approach has been proposed in Refs. [16,17] for nonlinear systems in the presence of both parametric uncertainties and nonrepeatable uncertain nonlinearities such as disturbances. The resulting ARC controllers achieve a guaranteed output tracking transient performance and final tracking accuracy in general. In addition, in the presence of

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Assoc. Editor: Reza Langari.

parametric uncertainties only, an asymptotic output tracking is achieved. These strong performance results achieved by ARC controllers motivate us to investigate whether the essential idea of ARC approach can be extended to the NN based controller designs to further improve the achievable performance of NN based controllers. At the same time, since only a special class of unknown nonlinear functions—a linear combination of known basis functions with unknown weights—has been considered in Refs. [16,17], such an extension is also of significant theoretical values since a more general class of unknown nonlinearities can be dealt with via NNs.

In this paper, the neural networks and ARC design philosophy will be integrated to design performance oriented control laws for a class of uncertain nonlinear systems. To present the idea in an easy-following way, the paper focuses on the development of neural network adaptive robust controllers (NNARCs) with output feedback for a class of single-input-single-output (SISO) nth order nonlinear system in a normal form. The form allows unknown nonlinearities existing in both system model and input channel, and the unknown nonlinearities could include nonrepeatable nonlinearities such as external disturbances as well. A high-gain observer is employed to estimate the system states. All unknown but repeatable nonlinear functions will be approximated by the outputs of multilayer neural networks to achieve a better model compensation for an improved performance. All NN weights are tuned on-line with no prior training needed. The discontinuous projection method with fictitious bounds [18] will be used to make sure that all NN weights are tuned within a prescribed range. By doing so, a controlled learning is achieved to avoid the possible destabilizing effect of on-line tuning of NN weights. Certain robust control terms are constructed to attenuate various model uncertainties effectively for a guaranteed output tracking transient performance and a guaranteed final tracking accuracy in general—a transient tracking performance that the existing NN based robust adaptive controllers [5–9] cannot achieve. In addition, if all the states of system are measurable, the unknown nonlinear function is in the functional range of the neural network and the ideal weights fall within the prescribed range, an asymptotic output tracking is also achieved to retain the perfect learning capability of neural networks in the ideal situation—a performance that existing NN based robust adaptive controllers [5–9] cannot have.

The rest of the paper is organized as follows: the problem formulation is detailed in Sec. 2. Some mathematical preliminaries are given in Sec. 3. The output feedback NNARC design is then developed in Sec. 4. The experimental results are presented in Sec. 5, followed by conclusions in Sec. 6.

2 Problem Formulation

The system to be considered in this paper has the following form [19]

$$x^{(n)} = f(\boldsymbol{x}) + b(\boldsymbol{x})u(t) + \Delta(\boldsymbol{x},t) \tag{1}$$

where x is the system output, $\boldsymbol{x}=[x,x^{(1)},\ldots,x^{(n-1)}]^T$ is the state variable vector with $x^{(i)}$ denoting the ith time derivative of the output x, $f(\boldsymbol{x})$ represents the unstructured state-dependent (or repeatable) unknown nonlinearity, $b(\boldsymbol{x})$ is the unknown nonlinear input gain, $u(t)$ is the system input, and $\Delta(\boldsymbol{x},t)$ represents the lumped *nonrepeatable* state-dependent nonlinearities such as disturbances.

Since $f(\boldsymbol{x})$ is not assumed to possess any special form, a three-layer neural network will be employed to approximate it for a better control performance. Thus, the following assumption is made

ASSUMPTION 1 [2]. *The NN approximation error associated with the nonlinear function f is assumed to be bounded by*

$$|f(\boldsymbol{x}) - \boldsymbol{w}_f^T\boldsymbol{g}_f(\boldsymbol{V}_f\boldsymbol{x}_a)| \leq \delta_f(\boldsymbol{x})d_f, \quad \forall\, \boldsymbol{x} \in \mathcal{R}^n \tag{2}$$

where $\delta_f(\boldsymbol{x})$ is a known non-negative bounded shape function of $\boldsymbol{x}$, d_f is an unknown positive constant, $\boldsymbol{x}_a=[x^T,-1]^T$ is the augmented input vector to the neural network (−1 term denotes the input bias), $\boldsymbol{w}_f=[w_{f1},\ldots,w_{fr_f}]^T$ is the hidden-output weight vector, $\boldsymbol{V}_f=[\boldsymbol{v}_{f1},\ldots,\boldsymbol{v}_{fr_f}]^T \in \mathcal{R}^{r_f\times(n+1)}$ is the input-hidden weight matrix with $\boldsymbol{v}_{fi} \in \mathcal{R}^{(n+1)\times 1}$, r_f is the number of neurons in the hidden layer, and $\boldsymbol{g}_f(\boldsymbol{V}_f\boldsymbol{x}^a)=[g_{f1}(\boldsymbol{v}_{f1}^T\boldsymbol{x}_a),\ldots,g_{fr_f}(\boldsymbol{v}_{fr_f}^T\boldsymbol{x}_a)]^T$ is the activation function vector, whose elements are usually smooth and bounded.

Remark 1. According to theorems in Refs. [1,2], nonlinearity f can be approximated by the output of a multilayer neural network to an arbitrarily accuracy on a compact set $\mathcal{A}_f$ provided that the number of the neurons is sufficiently large, i.e.,

$$|f(\boldsymbol{x}) - \boldsymbol{w}_f^T\boldsymbol{g}_f(\boldsymbol{V}_f x_a)| \leq \eta_f, \quad \forall\, \boldsymbol{x} \in \mathcal{A}_f \tag{3}$$

where η_f is an arbitrarily small positive number, and $\mathcal{A}_f$ is a compact subset of $\mathcal{R}^n$. Correspondingly, in Assumption 1, $\delta_f(\boldsymbol{x})$ can be chosen as 1 and d_f can be arbitrarily small when $\boldsymbol{x} \in \mathcal{A}_f$. Outside the compact set $\mathcal{A}_f$, the difference between the output of the neural networks and the true value of the nonlinear function may not be made arbitrarily small. It is however reasonable to expect that the approximation error outside the compact set $\mathcal{A}_f$ is bounded by a known nonlinear function multiplied by an unknown constant as assumed in Assumption 1.

In general, the form of the input gain $b(\boldsymbol{x})$ may not be known. However, it is practical to assume that $b(\boldsymbol{x})$ has a known sign. Thus, the following assumption is made:

ASSUMPTION 2. *The input gain $b(\boldsymbol{x})$ is nonzero with known sign. Thus, without loss of generality, assume*

$$b(\boldsymbol{x}) \geq b_l > 0 \quad \forall\, \boldsymbol{x} \in \mathcal{R}^n \tag{4}$$

where b_l is a known positive constant.

Furthermore, similar to Assumption 1, a three-layer neural network can be used to estimate the nonlinear input gain $b(\boldsymbol{x})$. Thus, the following assumption is made

ASSUMPTION 3. *The nonlinear input gain $b(\boldsymbol{x})$ can be approximated by the output of a multilayer neural network with*

$$|b(\boldsymbol{x}) - \boldsymbol{w}_b^T\boldsymbol{g}_b(\boldsymbol{V}_b x_a)| \leq \delta_b(\boldsymbol{x})d_b \quad \forall\, \boldsymbol{x} \in \mathcal{R}^n \tag{5}$$

where $\delta_b(\boldsymbol{x})$ is a known non-negative bounded shape function of $\boldsymbol{x}$, d_b is an unknown positive constant, $\boldsymbol{w}_b$, $\boldsymbol{g}_b$, and $\boldsymbol{V}_b$ are defined in similar ways as $\boldsymbol{w}_f$, $\boldsymbol{g}_f$, and *$\boldsymbol{V}_f$, respectively. It is assumed that the number of neurons in the hidden-layer is r_b. Then, $\boldsymbol{w}_b \in \mathcal{R}^{r_b\times 1}$, $\boldsymbol{g}_b \in \mathcal{R}^{r_b\times 1}$, and $\boldsymbol{V}_b \in \mathcal{R}^{r_b\times(n+1)}$. Similar to Remark 1, d_b can be made arbitrarily small, and $\delta_b=1$ when only a compact set $\mathcal{A}_b$ is considered.*

Although it is usually difficult to predict the type of disturbances that the system is going to encounter, it is always true that the disturbance is bounded in some ways. Hence, the following practical assumption is made

ASSUMPTION 4. *The nonrepeatable nonlinearity Δ is bounded by*

$$|\Delta(\boldsymbol{x},t)| \leq \delta_\Delta(\boldsymbol{x},t)d_\Delta(t) \tag{6}$$

where $\delta_\Delta(\boldsymbol{x},t)$ is a known non-negative bounded function, and $d_\Delta(t)$ is an unknown non-negative bounded time-varying function.

Remark 2. It seems that the system (1) is less general than the system $x^{(n)}=\phi^T(\boldsymbol{x},t)\theta+f(\boldsymbol{x})+b(\boldsymbol{x})u(t)+\Delta(\boldsymbol{x},t)$ proposed in Ref. [20] with $\phi^T(\boldsymbol{x},t)\theta$ representing the structured uncertainty. However, the structured nonlinearity $\phi^T(\boldsymbol{x},t)\theta$ can be viewed as the output of a two-layer (i.e., no hidden-layer) neural network with $\boldsymbol{x}$, t being the inputs, ϕ being the activation function vector, and θ being the weight vector. If ϕ does not depend on time t explicitly, then $\phi^T\theta$ can be included in $f(\boldsymbol{x})$, and the system considered in Ref. [20] is equivalent to system (1).

For any sufficiently smooth desired output trajectory $x_d(t)$, the desired state trajectory is $\boldsymbol{x}_d(t)=[x_d,x_d^{(1)},\ldots,x_d^{(n-1)}]^T$. *The control objective is to design a bounded control law for u such that the* *system state variable vector $\boldsymbol{x}$ tracks $\boldsymbol{x}_d$ as closely as possible.* If the tracking error vector is defined $\tilde{\boldsymbol{x}}(t)=\boldsymbol{x}(t)-\boldsymbol{x}_d(t)$, the control objective is equivalent to make the "size" of $\tilde{\boldsymbol{x}}(t)$ as small as possible.

In the following derivations $\hat{(\,)}$ represents the estimate of $(\,)$, $\tilde{(\,)}=\hat{(\,)}-(\,)$ is the estimate error of $(\,)$. For example, $\hat{\boldsymbol{w}}_f=[\hat{w}_{f1},\ldots,\hat{w}_{fr_f}]^T$ is the estimate of the hidden-output weight vector, and $\tilde{\boldsymbol{w}}_f=\hat{\boldsymbol{w}}_f-\boldsymbol{w}_f$ is the estimation error of the hidden-output weight vector. $\boldsymbol{g}_f$ is the shorthand notation for $\boldsymbol{g}_f(\boldsymbol{V}_f\boldsymbol{x}_a)$, $\hat{\boldsymbol{g}}_f$ represents $\boldsymbol{g}_f(\hat{\boldsymbol{V}}_f\boldsymbol{x}_a)$, and $\check{\boldsymbol{g}}_f$ stands for $\boldsymbol{g}_f(\hat{\boldsymbol{V}}_f\hat{\boldsymbol{x}}_a)$.

3 Mathematical Preliminaries

Before discussing the controller design, some mathematical preliminaries are presented.

3.1 Discontinuous Projection. Since $\boldsymbol{w}_f$ and $\boldsymbol{V}_f$ are constants, it is true that each element of $\boldsymbol{w}_f$ and $\boldsymbol{V}_f$ is bounded, i.e.,

$$\rho_{l,w_{fi}} \leqslant w_{fi} \leqslant \rho_{u,w_{fi}} \quad \rho_{l,v_{fij}} \leqslant v_{fij} \leqslant \rho_{u,v_{fij}}$$
$$i=1,\ldots,r_f \quad j=1,\ldots,n+1 \tag{7}$$

where the lower and upper bounds $\rho_{l,w_{fi}}$, $\rho_{u,w_{fi}}$, $\rho_{l,v_{fij}}$ and $\rho_{u,v_{fij}}$ may not be known. It is also reasonable to require that the estimates of the weights should be within the corresponding bounds. However, due to the fact that these bounds are not known a *prior*; certain fictitious bounds have to be used [18], and the discontinuous projection mapping will be constructed based on these fictitious bounds. The following general notation is introduced [21,22]

$$\mathbf{Proj}_{\hat{\star}}(\bullet)=\{Proj_{\hat{\star}}(\bullet_{ij})\} \tag{8}$$

with its ijth element defined as

$$Proj_{\hat{\star}}(\bullet_{ij})=\begin{cases} 0 \quad \text{if} \begin{cases} \hat{\star}_{ij}=\hat{\rho}_{u,\star_{ij}} \text{ and } \bullet_{ij}>0 \\ \text{or} \\ \hat{\star}_{ij}=\hat{\rho}_{l,\star_{ij}} \text{ and } \bullet_{ij}<0 \end{cases} \\ \bullet_{ij} \quad \text{otherwise} \end{cases} \tag{9}$$

where $\hat{\rho}_{l,\star_{ij}}$ and $\hat{\rho}_{u,\star_{ij}}$ are the fictitious lower and upper bound of $\star_{ij}$. For simplicity of notations, define $\hat{\rho}_{\star_{ij}}=\max\{|\hat{\rho}_{l,\star_{ij}}|,|\hat{\rho}_{u,\star_{ij}}|\}$, and denote $\hat{\rho}_\star=\{\hat{\rho}_{\star_{ij}}\}$ and $\rho_\star=\{\rho_{\star_{ij}}\}$.

3.2 Approximation Properties of Neural Networks. Since neural networks will be used in control design and their weights will be tuned on-line, it is beneficial to investigate the approximation properties of neural networks.

For a neural network with $\hat{\boldsymbol{x}}_{\{\text{in}\}}\in\mathcal{R}^{p+1}$, rather than $\boldsymbol{x}_{\{\text{in}\}}$, being its input vector, $\boldsymbol{V}=[\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{r_n}]^T\in\mathcal{R}^{r_n\times(p+1)}$ being its input-hidden weight matrix, $\boldsymbol{g}$ being the activation function vector, $\boldsymbol{w}\in\mathcal{R}^{m\times r_n}$ being its hidden-output weight matrix, we have the following theorem

THEOREM 3.1. *The output of neural network $\boldsymbol{w}^T\boldsymbol{g}(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}})$ can be approximated by the estimate $\hat{\boldsymbol{w}}^T\boldsymbol{g}(\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}})$ by the following form*

$$\boldsymbol{w}^T\boldsymbol{g}(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}})=\hat{\boldsymbol{w}}^T\check{\boldsymbol{g}}-\tilde{\boldsymbol{w}}^T(\check{\boldsymbol{g}}-\check{\boldsymbol{g}}'\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}})-\tilde{\boldsymbol{w}}^T\check{\boldsymbol{g}}'\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}+\check{d}_{\text{NN}} \tag{10}$$

where $\check{\boldsymbol{g}}=\boldsymbol{g}(\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}})$, $\check{\boldsymbol{g}}'=\text{diag}\{\check{g}_1',\ldots,\check{g}_r'\}$ *with* $\check{g}_i'=g_i'(\hat{\boldsymbol{v}}_i^T\hat{\boldsymbol{x}}_{\{\text{in}\}})=d_{g_i}(z)/d_z|_{z=\hat{\boldsymbol{v}}_i^T\hat{\boldsymbol{x}}_{\{\text{in}\}}}$, $i=1,\ldots,r_n$, *and residual term* $\check{d}_{\text{NN}}=-\tilde{\boldsymbol{w}}^T\check{\boldsymbol{g}}'\boldsymbol{V}\hat{\boldsymbol{x}}_{\{\text{in}\}}+\boldsymbol{w}^T[-\check{\boldsymbol{g}}'\boldsymbol{V}\tilde{\boldsymbol{x}}_{\{\text{in}\}}+\mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}}-\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|)]$ *with* $\mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}}-\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|)$ *being the sum of the higher order terms.* ◇

Proof. See Appendix A. □

LEMMA 3.1. *The residual term $\check{d}_{\text{NN}}$ can also be bounded by a linear-in-parameter function, i.e.,*

$$\|\check{d}_{\text{NN}}\|_2 \leqslant \check{\boldsymbol{\alpha}}^T\check{\boldsymbol{Y}} \tag{11}$$

where $\check{\boldsymbol{\alpha}}$ is an unknown vector constituting of positive elements, and the known function vector $\check{\boldsymbol{Y}}$ is defined as follows

$$\check{\boldsymbol{Y}}=[1,\ \|\hat{\boldsymbol{x}}_{\{\text{in}\}}\|_2,\ \|\hat{\boldsymbol{w}}\|_F\|\hat{\boldsymbol{x}}_{\{\text{in}\}}\|_2,\ \|\hat{\boldsymbol{V}}\|_F\|\hat{\boldsymbol{x}}_{\{\text{in}\}}\|_2]^T \quad \diamond \tag{12}$$

Proof. See Appendix B. □

According to Theorem 3.1 and Lemma 3.1, the following two equations hold

$$\boldsymbol{w}_f^T\boldsymbol{g}_f(\boldsymbol{V}_f\boldsymbol{x}_a)=\hat{\boldsymbol{w}}_f^T\check{\boldsymbol{g}}_f-\tilde{\boldsymbol{w}}_f^T(\check{\boldsymbol{g}}_f-\check{\boldsymbol{g}}_f'\hat{\boldsymbol{V}}_f\hat{\boldsymbol{x}}_a)-\hat{\boldsymbol{w}}_f^T\check{\boldsymbol{g}}_f'\tilde{\boldsymbol{V}}_f\hat{\boldsymbol{x}}_a+\check{d}_{f\text{NN}} \tag{13}$$

$$\boldsymbol{w}_b^T\boldsymbol{g}_b(\boldsymbol{V}_b\boldsymbol{x}_a)=\hat{\boldsymbol{w}}_b^T\check{\boldsymbol{g}}_b-\tilde{\boldsymbol{w}}_b^T(\check{\boldsymbol{g}}_b-\check{\boldsymbol{g}}_b'\hat{\boldsymbol{V}}_b\hat{\boldsymbol{x}}_a)-\hat{\boldsymbol{w}}_b^T\check{\boldsymbol{g}}_b'\tilde{\boldsymbol{V}}_b\hat{\boldsymbol{x}}_a+\check{d}_{f\text{NN}} \tag{14}$$

where the notations are defined in the same way as in Theorem 3.1 and Lemma 3.1, and

$$|\check{d}_{f\text{NN}}| \leqslant \check{\boldsymbol{\alpha}}_f^T\check{\boldsymbol{Y}}_f \quad \check{\boldsymbol{Y}}_f=[1,\ \|\hat{\boldsymbol{x}}_a\|_2,\ \|\hat{\boldsymbol{w}}_f\|_2\|\hat{\boldsymbol{x}}_a\|_2,\ \|\hat{\boldsymbol{V}}_f\|_F\|\hat{\boldsymbol{x}}_a\|_2]^T \tag{15}$$

$$|\check{d}_{b\text{NN}}| \leqslant \check{\boldsymbol{\alpha}}_b^T\check{\boldsymbol{Y}}_b \quad \check{\boldsymbol{Y}}_b=[1,\ \|\hat{\boldsymbol{x}}_a\|_2,\ \|\hat{\boldsymbol{w}}_b\|_2\|\hat{\boldsymbol{x}}_a\|_2,\ \|\hat{\boldsymbol{V}}_b\|_F\|\hat{\boldsymbol{x}}_a\|_2]^T \tag{16}$$

4 NNARC Design with Output Feedback

Although it is usually assumed that the input-hidden weights are known [5–7,23] and it can be achieved by the off-line training of neural networks, it might be more practical and beneficial if this assumption can be relaxed and input-hidden weights can be tuned on-line. Furthermore, since it may not be possible to measure all the system states in reality, the output feedback control will be developed in this section. In the following, for simplicity, the sigmoid function will be used as activation functions. Other type of activation functions (e.g., RBF [8], bipolar sigmoid function [5]) can be worked out in the same way as long as the activation functions and their derivatives are bounded functions.

4.1 High-Gain Observer Design. In the case that only the output x is available, an observer is needed to estimate $x^{(i)}$, $i=1,\ldots,n-1$. Since the forms of $f(\boldsymbol{x})$ and $b(\boldsymbol{x})$ are unknown, the design methods of a nonlinear Luenberger-type observer developed in Refs. [13–15] cannot be applied. Hence, as in Refs. [24,9], a high-gain observer is introduced.

LEMMA 4.1. *Suppose the function $y(t)$ and its first n derivatives are bounded. Consider the following linear system*

$$\begin{aligned} \varepsilon\dot{\xi}_1 &= \xi_2 \\ &\vdots \\ \varepsilon\dot{\xi}_{n-1} &= \xi_n \\ \varepsilon\dot{\xi}_n &= -b_1\xi_n-\cdots-b_{n-1}\xi_2-\xi_1+y(t) \end{aligned} \tag{17}$$

where ε is any small positive constant, and the parameters $b_1,\ldots,b_{n-1}$ are chosen such that the polynomial $s^n+b_1s^{n-1}+\cdots+b_{n-1}s+1$ is Hurwitz. Then, there exist positive constants D_k, $k=2,\ldots,n$, and t^* such that for all $t>t^*$,

$$\frac{\xi_{k+1}}{\varepsilon_k}-y^{(k)}=-\varepsilon\psi^{(k+1)} \quad k=1,\ldots,n-1 \tag{18}$$

$$\left|\frac{\xi_{k+1}}{\varepsilon_k}-y^{(k)}\right| \leqslant \varepsilon D_{k+1} \quad k=1,\ldots,n-1 \tag{19}$$

where $\psi=\xi_n+b_1\xi_{n-1}+\cdots+b_{n-1}\xi_1$, $\psi^{(k)}$ denotes the kth derivative of ψ, and $|\psi^{(k)}|\leqslant D_k$.

Proof. Refer to the proof of Lemma in Sec. 6.2 of Ref. [24].

4.2 Output Feedback NNARC Design. Since the control objective is to force $\boldsymbol{x}$ to track $\boldsymbol{x}_d$, a concise tracking error metric is defined by [8]

$$s(t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}(t) \quad \text{with } \lambda > 0 \tag{20}$$

where $\tilde{x}(t)=x(t)-x_d(t)$ is the output tracking error, λ is a positive constant. (20) can be rewritten as $s(t)=\boldsymbol{\lambda}^T\tilde{\boldsymbol{x}}(t)$ with the ith element of vector $\boldsymbol{\lambda}$ being given by

$$C_{n-1}^{i-1}\lambda^{n-i} = \frac{(n-1)!}{(n-i)!\,(i-1)!}\lambda^{n-i}.$$

The equation $s(t)=0$ defines a hyperplane in the $\mathcal{R}^n$ state space on which the tracking error vector exponentially converges to zero. As a result, the perfect tracking can be asymptotically achieved by maintaining this condition [25].

If the system states are available, same as in Refs. [19,20], the following state feedback NNARC can be designed

$$u = u_a + u_s \tag{21}$$

with

$$u_a = -\frac{1}{\hat{\boldsymbol{w}}_b^T\hat{\boldsymbol{g}}_b}[a_r(t) + \hat{\boldsymbol{w}}_f^T\hat{\boldsymbol{g}}_f]$$
$$u_s = u_{s1} + u_{s2} \quad u_{s1} = -\frac{1}{b_l}ks - \frac{1}{b_l}[\hat{\boldsymbol{\alpha}}_f^T\boldsymbol{Y}_f + \hat{\boldsymbol{\alpha}}_b^T\boldsymbol{Y}_b|u_a|]\mathrm{sgn}(s) \tag{22}$$

where $k>0, a_r(t)=\boldsymbol{\lambda}_v^T\tilde{\boldsymbol{x}}-x_d^{(n)}$ with $\boldsymbol{\lambda}_v^T=[0,\lambda^{n-1},\ldots,C_{n-1}^{i-2}\lambda^{n-i+1},\ldots,(n-1)\lambda]$, and u_{s2} is a robust term.

However, since only the system output is available, the state estimate $\hat{\boldsymbol{x}}$ is in order,

$$\hat{\boldsymbol{x}} = \left[x, \frac{\xi_2}{\varepsilon}, \ldots, \frac{\xi_n}{\varepsilon^{n-1}}\right]^T \tag{23}$$

Then, the error between $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}_d$ is $\hat{\boldsymbol{\zeta}}=\hat{\boldsymbol{x}}-\boldsymbol{x}_d=[x-x_d,\xi_2/\varepsilon-x_d^{(1)},\ldots,\xi_n/\varepsilon^{n-1}-x_d^{(n-1)}]^T$. Furthermore, the following variables are defined

$$\hat{\boldsymbol{x}}_a = [\hat{\boldsymbol{x}}^T, -1]^T$$
$$\boldsymbol{\psi} = [0, \psi^{(2)}, \ldots, \psi^{(n)}]^T$$
$$\hat{s} = \boldsymbol{\lambda}^T\boldsymbol{\zeta} = \boldsymbol{\lambda}^T(\hat{\boldsymbol{x}} - \boldsymbol{x}_d) = s - \varepsilon\boldsymbol{\lambda}^T\boldsymbol{\psi} \tag{24}$$
$$\hat{a}_r = \boldsymbol{\lambda}_u^T\boldsymbol{\zeta} - x_d^{(n)} = a_r(t) - \varepsilon\boldsymbol{\lambda}_v^T\boldsymbol{\psi}$$

By replacing the states with their estimates in control law (21), the following output feedback NNARC is obtained

$$u = u_a + u_s \tag{25}$$

with

$$u_a = -\frac{1}{\hat{\boldsymbol{w}}_b^T\check{\boldsymbol{g}}_b}[a_r(t) + \hat{\boldsymbol{w}}_f^T\check{\boldsymbol{g}}_f],$$
$$u_s = u_{s1} + u_{s2}, \quad u_{s1} = -\frac{1}{b_l}k\hat{s} - \frac{1}{b_l}[\check{\boldsymbol{\alpha}}_f^T\check{\boldsymbol{Y}}_f + \check{\boldsymbol{\alpha}}_b^T\check{\boldsymbol{Y}}_b|u_a|]\mathrm{sgn}(\hat{s}) \tag{26}$$

where $k>0$, and u_{s2} is a robust term to be synthesized later.

By using control law (25) and considering the third equation in (24), the following error equation can be obtained

$$\dot{\hat{s}} = \dot{s} - \varepsilon\boldsymbol{\lambda}^T\dot{\boldsymbol{\psi}} = x^{(n)} + a_r(t) - \varepsilon\boldsymbol{\lambda}^T\dot{\boldsymbol{\psi}}$$
$$= [f(\boldsymbol{x}) + a_r(t)] + b(\boldsymbol{x})u_a + b(\boldsymbol{x})u_s + \Delta - \varepsilon\boldsymbol{\lambda}^T\dot{\boldsymbol{\psi}}$$
$$= [f(\boldsymbol{x}) - \boldsymbol{w}_f^T\boldsymbol{g}_f] + [\boldsymbol{w}_f^T\boldsymbol{g}_f - \hat{\boldsymbol{w}}_f^T\check{\boldsymbol{g}}_f] + [a_r(t) - \hat{a}_r(t)] + [b(\boldsymbol{x})$$
$$- \boldsymbol{w}_b^T\boldsymbol{g}_b]u_a + [\boldsymbol{w}_b^T\boldsymbol{g}_b - \hat{\boldsymbol{w}}_b^T\check{\boldsymbol{g}}_b]u_a + \Delta + b(\boldsymbol{x})u_{s2} + b(\boldsymbol{x})u_{s1} - \varepsilon\boldsymbol{\lambda}^T\dot{\boldsymbol{\psi}}$$
$$= [f(\boldsymbol{x}) - \boldsymbol{w}_f^T\boldsymbol{g}_f + \check{d}_{f\mathrm{NN}}] + [-\tilde{\boldsymbol{w}}_f^T(\check{\boldsymbol{g}}_f - \check{\boldsymbol{g}}_f'\hat{\boldsymbol{V}}_f\hat{\boldsymbol{x}}_a) - \hat{\boldsymbol{w}}_f^T\check{\boldsymbol{g}}_f'\tilde{\boldsymbol{V}}_f\hat{\boldsymbol{x}}_a]$$
$$+ \varepsilon[\boldsymbol{\lambda}_u^T\boldsymbol{\psi} - \boldsymbol{\lambda}^T\dot{\boldsymbol{\psi}}] + \{[b(\boldsymbol{x}) - \boldsymbol{w}_b^T\boldsymbol{g}_b - \check{d}_{b\mathrm{NN}}] + [-\tilde{\boldsymbol{w}}_b^T(\check{\boldsymbol{g}}_b - \check{\boldsymbol{g}}_b'\hat{\boldsymbol{V}}_b\hat{\boldsymbol{x}}_a)$$
$$- \hat{\boldsymbol{w}}_b^T\check{\boldsymbol{g}}_b'\tilde{\boldsymbol{V}}_b\hat{\boldsymbol{x}}_a]\}u_a - k\frac{b(\boldsymbol{x})}{b_l}\hat{s} - \frac{b(\boldsymbol{x})}{b_l}\check{\boldsymbol{\alpha}}_f^T\check{\boldsymbol{Y}}_f\mathrm{sgn}(\hat{s})$$
$$- \frac{b(\boldsymbol{x})}{b_l}\check{\boldsymbol{\alpha}}_b^T\check{\boldsymbol{Y}}_b|u_a|\mathrm{sgn}(\hat{s}) + b(\boldsymbol{x})u_{s2} + \Delta \tag{27}$$

where Eqs. (13) and (14) are used in the derivation of the fourth equality.

Similar to the adaptation laws in Refs. [19,20], the following adaptation laws are proposed

$$\dot{\hat{\boldsymbol{w}}}_f = \mathbf{Proj}_{\hat{\boldsymbol{w}}_f}(\Gamma_{wf}\hat{s}(\check{\boldsymbol{g}}_f - \check{\boldsymbol{g}}_f'\hat{\boldsymbol{V}}_f\hat{\boldsymbol{x}}_a)) \tag{28}$$

$$\dot{\hat{\boldsymbol{V}}}_f = \mathbf{Proj}_{\hat{V}_f}[(\Gamma_{vf}\hat{\boldsymbol{x}}_a\hat{s}\hat{\boldsymbol{w}}_f^T\check{\boldsymbol{g}}_f')^T] \tag{29}$$

$$\dot{\check{\boldsymbol{\alpha}}}_f = \mathbf{Proj}_{\check{\boldsymbol{\alpha}}_f}(\Gamma_{\alpha f}|\hat{s}|\check{\boldsymbol{Y}}_f) \tag{30}$$

$$\dot{\hat{\boldsymbol{w}}}_b = \mathbf{Proj}_{\hat{\boldsymbol{w}}_b}\{\Gamma_{wb}[u_a\hat{s}(\check{\boldsymbol{g}}_b - \check{\boldsymbol{g}}_b'\hat{\boldsymbol{V}}_b\hat{\boldsymbol{x}}_a)]\} \tag{31}$$

$$\dot{\hat{\boldsymbol{V}}}_b = \mathbf{Proj}_{\hat{V}_b}\{[\Gamma_{Vb}u_a\hat{\boldsymbol{x}}_a\hat{s}\hat{\boldsymbol{w}}_b^T\check{\boldsymbol{g}}_b']^T\} \tag{32}$$

$$\dot{\check{\boldsymbol{\alpha}}}_b = \mathbf{Proj}_{\check{\boldsymbol{\alpha}}_b}\{\Gamma_{\alpha b}[|\hat{s}|\check{\boldsymbol{Y}}_b|u_a|]\} \tag{33}$$

The robust control term u_{s2} is synthesized to satisfy the following conditions

$$\hat{s}\{[f(\boldsymbol{x}) + a_r(t)] + b(\boldsymbol{x})u_a + \Delta + b(\boldsymbol{x})u_{s2}\} \leqslant \epsilon_{s0} \tag{34}$$

$$\hat{s}u_{s2} \leqslant 0 \tag{35}$$

where

$$\epsilon_{s0} = \epsilon_{ss} + \frac{\epsilon^2|\boldsymbol{\lambda}_v^T\boldsymbol{\psi} - \boldsymbol{\lambda}^T\dot{\boldsymbol{\psi}}|^2}{D^2}\epsilon_6 \tag{36}$$

with

$$\epsilon_{ss} = \left|1 + \frac{|(\|\rho_{wf}\|_2 - \|\hat{\rho}_{wf}\|_2)|}{\|\hat{\rho}_{wf}\|_2}\right|^2\epsilon_1 + \left|1 + \frac{|(\|\rho_{wb}\|_2 + \|\hat{\rho}_{wb}\|_2)|}{\|\hat{\rho}_{wb}\|_2}\right|^2\epsilon_2$$
$$+ \epsilon_3 d_f^2 + \epsilon_4 d_b^2 + \epsilon_5\|d\|_\infty^2 \tag{37}$$

in which ϵ_i, $i=1,\ldots,6$, and D are the positive design constants.

Remark 3. The robust term u_{s2} satisfying Eqs. (34) and (35) may be chosen by

$$u_{s2} = -k_{s2}s \tag{38}$$

where k_{s2}, is a nonlinear gain large enough such that $k_{s2} \geqslant \sum_{i=1}^{i=6}h_i^2/4\epsilon_i$ with $h_1 \geqslant (1/\sqrt{b_l})\|\hat{\boldsymbol{g}}_f\|(\|\hat{\boldsymbol{w}}_f\| + \|\hat{\rho}_{wf}\|)$, $h_2 \geqslant (1/\sqrt{b_l}) \times \|\hat{\boldsymbol{g}}_b\|(\|\hat{\boldsymbol{w}}_b\| + \|\hat{\rho}_{wb}\|)|u_a|$, $h_3 \geqslant (1/\sqrt{b_l})\delta_f$, $h_4 \geqslant (1/\sqrt{b_l})|u_a|\delta_b$, $h_5 \geqslant (1/\sqrt{b_l})\delta(x,t)$, $h_6 \geqslant (1/\sqrt{b_l})D$, $\|\hat{\boldsymbol{g}}_f\| = \sqrt{\sum_{i=1}^{i=r_f}\|\hat{g}_{fi}\|_\infty^2}$, and $\|\hat{\boldsymbol{g}}_b\| = \sqrt{\sum_{i=1}^{i=r_b}\|\hat{g}_{bi}\|_\infty^2}$.

THEOREM 4.1. *With NNARC (25) and the adaptation laws (28)–(33), the following results hold:*

A. *In general, all signals are bounded. Furthermore, the sliding error $s(t)$ exponentially converges to a small value and is bounded by*

$$|s| \leqslant \sqrt{\exp(-2kt)[s(0) - \varepsilon\boldsymbol{\lambda}^T\boldsymbol{\psi}(0)]^2 + \frac{\epsilon_{so}}{k}[1 - \exp(-2kt)]}$$
$$+ \varepsilon|\boldsymbol{\lambda}^T\boldsymbol{\psi}| \tag{39}$$

B. *The actual tracking error is asymptotically bounded by*

$$\lim_{t\to\infty}|\tilde{x}^{(i)}| \leq 2^i\lambda^{i-n+1}\left[\sqrt{\frac{\varepsilon_{s0}}{k}} + \varepsilon|\boldsymbol{\lambda}^T\boldsymbol{\psi}|\right] \qquad (40)$$

C. *When the discontinuous sign function $sgn(\hat{s})$ in* Eq. (26) *is dropped or replaced by any continuous function $v(\hat{s})$ satisfying $\hat{s}v(\hat{s}) \geq 0$, Results A and B still remain valid.*

D. *If all the system states are available, $f(\boldsymbol{x}) = \boldsymbol{w}_f^T\boldsymbol{g}_f(\boldsymbol{V}_f\boldsymbol{x}_a)$ and $b(\boldsymbol{x}) = \boldsymbol{w}_b^T\boldsymbol{g}_b(\boldsymbol{V}_b\boldsymbol{x}_a)$* [3], *i.e., the nonlinear functions f and b are in the functional range of the corresponding neural network respectively, then, in addition to the results in A and B, an asymptotic output tracking is achieved provided that there is no external disturbance (i.e., $\Delta=0$), and all the ideal weights lie within the fictitious bounds (i.e., $\hat{\rho}_{l,w_{fi}} \leq w_{fi} \leq \hat{\rho}_{u,w_{fi}}$, $\hat{\rho}_{l,v_{fij}} \leq v_{fij} \leq \hat{\rho}_{u,v_{fij}}$, $\forall i=1,\ldots,r_f$, $\forall j=1,\ldots,n$, $0 \leq \alpha_{fk} \leq \hat{\rho}_{u,\alpha_{fk}}$, $k=1,\ldots,4$, $\hat{\rho}_{l,w_{bi}} \leq w_{bi} \leq \hat{\rho}_{u,w_{bi}}$, $\hat{\rho}_{l,v_{bij}} \leq v_{bij} \leq \hat{\rho}_{u,v_{bij}}$, $\forall i=1,\ldots,r_b$, $\forall j=1,\ldots,n$, $0 \leq \alpha_{bk} \leq \hat{\rho}_{u,\alpha_{bk}}$, $k=1,\ldots,4$).*

Proof. See Appendix C.

Remark 4. Results *A* and *B* of Theorem 4.1 show that the proposed NNARC achieves a guaranteed transient performance and final tracking accuracy in general; it is seen that the exponentially converging rate $2k$ and the bounds of the final output tracking error ($\|\tilde{x}\|_\infty \leq \lambda^{-n+1}[\sqrt{\varepsilon_{so}/k} + \varepsilon|\boldsymbol{\lambda}^T\boldsymbol{\psi}|]$) are related to the design parameters k, ε_1, ε_2, ε_3, and ε_4 in known form, and can be adjusted by suitably choosing those design parameter. These results are thus much stronger than those in Refs. [5–8]. In all those schemes in Refs. [5–8], the transient performance is not guaranteed. Although the final tracking error can be made arbitrarily small by properly choosing the NN and the high-gain observer in Ref. [9], the transient performance is not clear, either. In this sense, the result obtained here is stronger than that in Ref. [9].

Remark 5. Based on the idea in Ref. [10], it can also be shown that the asymptotic tracking is possible in the case of output feedback when the persistent excitation condition is satisfied, $f = \boldsymbol{w}_f^T\boldsymbol{g}_f$, $g = w_b^T\boldsymbol{g}_b$, and $\Delta=0$.

Remark 6. The high-gain observer may exhibit a peaking phenomenon during the transient period [12]. In order to preventing the peaking from being transmitted to the plant, the input saturation method can be used [12].

Remark 7. Remark 5 shows one possibility to achieve an asymptotic tracking when only output feedback is used. However, it is usually hard to check the condition of persistent excitation. Result D of Theorem 4.1 shows another possibility when the full state feedback is possible. The proposed NNARC is able to accomplish its learning goal (the assumptions in D of Theorem 4.1 represents the ideal situation that a neural network is intended to be used for). As a result, an improved tracking performance—asymptotic output tracking—is achieved. It is noted that all previous research [5–7,9] cannot attain this level of perfect learning capability.

Remark 8. It is clear that the larger the fictitious parameter ranges chosen in the construction of the projection mappings (9) are, the wider the approximation range of the resulting NN would be. Thus, theoretically, if the control input is unlimited as implicitly assumed in the proof of Theorem 4.1, the fictitious parameter ranges should be chosen larger enough so that the resulting NN is able to approximate the unknown nonlinearities well to obtain asymptotic output tracking performance stated in Result D of Theorem 4.1. However, as in Ref. [18], this should not be overdone when the control input has a limited authority and the system may be occasionally subjected to large disturbances as in most applications; it has been observed in Refs. [26,18] that using too large fictitious bounds for on-line parameter estimates may result in phenomenon similar to the usual integration windup problem when the control input is saturated in the presence of occasional large disturbances. Although the theoretical results have not been well established in Refs. [26,18], it is shown through experiments and simulations that the integration windup problem may be alleviated by choosing the fictitious bounds appropriately. Thus, a compromise has to be made between the NN approximation range and the degree of the robustness to large disturbances and control saturations in practice.

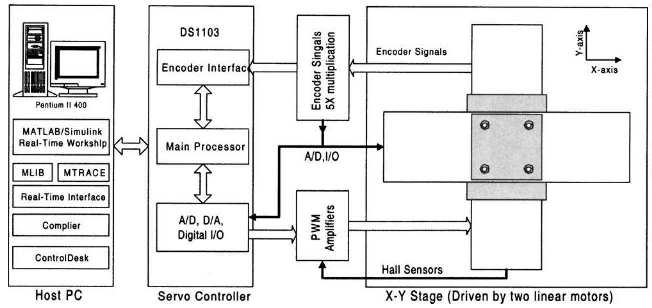

Fig. 1 Experimental setup

5 Experimental Studies

To test the proposed NNARC strategy and study fundamental problems associated with high-accuracy motion control of linear motor drive systems, the proposed NNARC will be used to control a two-axis *X-Y* positioning stage.

5.1 System Setup and Design Model. As shown in Fig. 1, the two-axis *X-Y* positioning stage consists of four major components: a precision *X-Y* stage with two integrated linear drive motors, two linear encoders with a measurement resolution of 1 μm after quadrature, a servo controller, and a host PC. The two axes of the *X-Y* stage are mounted orthogonally on a horizontal plane with *X* axis on top of *Y* axis. A particular feature of the setup is that the two linear motors are of different type: *X* axis is driven by an Anorad LEM-S-3-S linear motor (epoxy core) and *Y* axis is driven by an Anorad LCK-S-1 linear motor (iron core). They represent the two most commonly used linear motors and have different characteristics. In the experiments, only *X* axis is used. The control system is implemented using a dSPACE DS1103 controller board. The controller executes programs at a sampling frequency $f_s=2.5$ kHz.

The system considered here is a linear positioning stage driven by a current-controlled three-phase epoxy core linear motor supported by recirculating bearings. Since the system has a much faster electrical response in comparison to the mechanical response, the electrical dynamics are neglected. The mathematical model of the system can thus be described by [20]

$$M\ddot{x} = -Bx^{(1)} - F_a(x,x^{(1)}) + u + F_d(x,x^{(1)},t) \qquad (41)$$

where x represents the position of the inertia load, M is the normalized (with respect to the unit input voltage) mass of the inertia load plus the coil assembly, u is the input voltage to the motor, B is the equivalent viscous friction coefficient of the system, F_a is the lumped nonlinear force which includes the effects of nonlinear friction and depends on both position and velocity, and F_d is the normalized external disturbance force (e.g., cutting force in machining), which may be state dependent and time-varying. It is assumed that M, B are unknown constants, and F_d is bounded. Dividing both sides of Eq. (41) yields the same form as Eq. (1) with $\boldsymbol{x}=[x,x^{(1)}]^T$, $f(\boldsymbol{x})=-(B/M)(x)^{(1)}-F_a/M$, $b(\boldsymbol{x})=1/M$, $\Delta = F_d/M$. In order to focus on the effects of the totally unknown force F_a, according to Remark 2, $f(\boldsymbol{x})$ is split into two terms, $-(B/M)x^{(1)}$ and $-F_a/M$ with $-(B/M)x^{(1)}$ having the form of

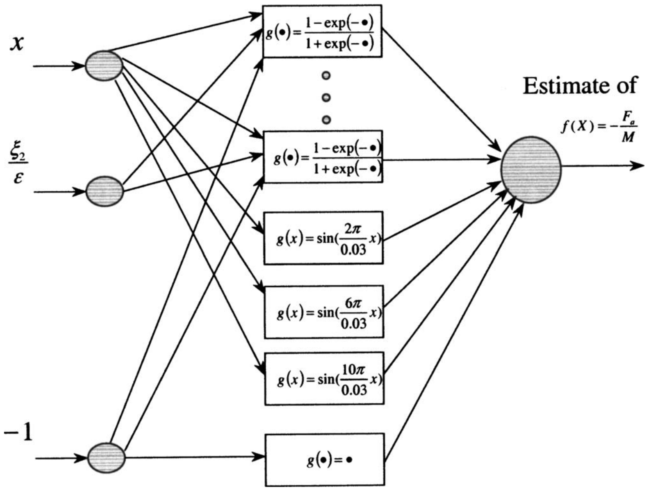

Fig. 2 NN structure for estimating $-F_a/M$

$\phi^T\theta$, $\phi = x^{(1)}$ and $\theta = -B/M$. In order to save the notation, from now on, $f(\boldsymbol{x}) = -F_a/M$. Since B/M and $b(\boldsymbol{x})$ are simply unknown constants, the main nonlinearity in the system is $f(\boldsymbol{x})$, and the focus of NN design will be on $f(\boldsymbol{x})$.

5.2 Design of Neural Network and Adaptation Laws. Based on the observations in Ref. [20], the input vector to the NN for $f(\boldsymbol{x})$ is $[x, x^{(1)}, -1]^{(T)}$. Since $x^{(1)}$ is not accessible, the high-gain observer (17) is employed with $n=2$, $b_1 = \sqrt{2}$, $\varepsilon = 1/(2\pi \times 260) = 6.12 \times 10^{-4}$, and $y = x$ to get the estimate of velocity as ξ_2/ε. Hence, the actual input to the NN is $[x, \xi_2/\varepsilon, -1]^T$. Similar to that in Ref. [20], the structure of the NN is shown in Fig. 2, which has three types of neurons.

1. *Type-I*: Five hidden neurons with a conventional sigmoid function $g(\bullet) = (1-\exp(-\bullet))/(1+\exp(-\bullet))$ as activation function are used to capture the phenomenon that F_a depends on the velocity (especially, the direction of the velocity) and slowly changes with position. Since the slope of $g(\bullet)$ around the origin is unknown, the corresponding input-hidden weights are assumed to be unknown.
2. *Type-II*: Three hidden neurons with sinusoid type of functions $(\sin(2\pi/0.03)x), \sin((6\pi/0.03)x)$, and $\sin((10\pi/0.03)x)$ being activation functions are used to compensate for the force ripple, which is a periodic function of the position. The input-hidden weights is set to be $[1,0,0]^T$ while the hidden-output weights are unknown.
3. *Type-III*: One hidden neuron with an identity function $g(\bullet) = \bullet$ as activation function is used to capture the lumped average effect of all uncertainties including disturbance. Corresponding to the input $[x, \xi_2/\epsilon, -1]^T$, its input-hidden weights are set to be $[0,0,1]^T$. The hidden-output weight is unknown. In this case, since $g(\bullet)$ is an identity function, it may also be viewed that there is no hidden layer neuron.

Since the input-hidden weights are fixed in the last two types of hidden neurons and the output of position is available, the corresponding adaptation laws can be simplified into

$$\dot{\hat{\boldsymbol{w}}}_f = \mathbf{Proj}_{\hat{\boldsymbol{w}}_f}(\Gamma_{w_f}\hat{s}\boldsymbol{g}_f) \tag{42}$$

where $g_{fi} = \sin(2\pi/0.03x)$, $\sin(6\pi/0.03x)$, $\sin(10\pi/0.03x)$, respectively, for *Type-II* neurons, and $g_f = 1$ for *Type-III* neurons.

For the structured uncertainty term $\phi^T\theta = -(B/M)x^{(1)}$, according to Remark 2 and similar to Eq. (42), the adaptation law for unknown constant $\theta = -B/M$ is given as follows

$$\dot{\hat{\theta}} = \mathbf{Proj}_{\hat{\theta}}\left(\Gamma_\theta \hat{s}\frac{\xi_2}{\varepsilon}\right) \tag{43}$$

where ξ_2/ε is the estimate of velocity $x^{(1)}$.

From Eq. (41), it is clear that the input gain $1/M$ is a constant. This property will be taken into account to simplify the NN for $b(\boldsymbol{x})$ by letting g_b to be identity function. Consequently, there is no hidden layer in NN for $b(\boldsymbol{x})$, and only $\hat{w}_b$ (i.e., $\hat{b}$) needs to be adjusted. Correspondingly, adaptation laws (32) and (33) are not needed, and the adaptation law (31) becomes

$$\dot{\hat{w}}_b = \dot{\hat{b}} = \mathbf{Proj}_{\hat{b}}\{\Gamma_b u_a \hat{s}\} \tag{44}$$

5.3 Output Feedback NNARC Control Law. Since no hidden layer exists in NN for $b(\boldsymbol{x})$, the robust control term $-(1/b_l)\check{\boldsymbol{\alpha}}_b^T\check{\mathbf{Y}}_b|u_a|\mathrm{sgn}(\hat{s})$ in Eq. (26) is not needed. Furthermore, as stated in Result C of Theorem 4.1, $(2/\pi)\arctan(\psi_a\hat{s})$ can be used to replace the sign function in Eq. (26) to avoid control input chattering. The resulting u_{s1} is

$$u_{s1} = -\frac{1}{b_l}\left[k\hat{s} + \frac{2}{\pi}\check{\boldsymbol{\alpha}}_f^T\check{\mathbf{Y}}_f\arctan(\psi_a\hat{s})\right] \tag{45}$$

in which $\psi_a = 90$ is used in the experiments. For easy reference, the NNARC for the linear motor system is given as follows

$$u_{\mathrm{NNARC}} = \frac{\hat{a}_r(t) + \hat{\theta}\frac{\xi_2}{\varepsilon} + \hat{\boldsymbol{w}}_f^T\check{\boldsymbol{g}}_f}{\hat{b}} - \frac{1}{b_l}\left[k\hat{s} + \frac{2}{\pi}\check{\boldsymbol{\alpha}}_f^T\check{\mathbf{Y}}_f\arctan(\psi_a\hat{s})\right] + u_{s2} \tag{46}$$

where u_{s2} assumes the form (38) with $k_{s2} = \max(3, \Sigma_{i=1}^{i=5}(h_i^2/4\epsilon_i))$.

5.4 Parameters in Control Law and Adaptation Laws. The parameters used in control law are $\lambda = 200$ for the sliding plane and $k = 20$ as the constant gain. The final tracking accuracy indices are $\epsilon_i = 5000$, $i = 1, \ldots, 6$. As discussed in Remark 6, a saturation level $u = 10$ V is set for the control input voltage.

As stated in Remark 8, the fictitious bounds of the NN weights should be chosen based on the particular properties of the system and the types of weights and neurons used. For *Type-I* neurons, large fictitious bounds can be assumed for the input-hidden weights since sigmoidal functions are always bounded by ± 1. For other types of neurons, the fictitious bounds of their weights need to be chosen conservatively to avoid the possible integration windup problem. Under this general guideline, the following different adaptation rates and fictitious bounds are chosen empirically

1. *Type-I*: For input-hidden weights, 2×10^3 is used as adaptation rate, and the fictitious lower and upper bounds are -3×10^3 and 3×10^3, respectively. For hidden-output weights, adaptation rate is 3×10^2, and the fictitious lower and upper bounds are -20 and 20, respectively.
2. *Type-II*: Only hidden-output weights are to be adjusted. The adaptation rate is 10^3. The fictitious lower and upper bounds are -20 and 20, respectively.
3. *Type-III*: Same as *Type-II* except that the adaptation rate is 1.35×10^5.

The initial values of all the weights are zeros.

Since $0.025 \leqslant M \leqslant 0.1$, $b_l = 10$ results. Standard least-square identification is performed to obtain the parameters of the system. The nominal values of M is 0.027 ($\mathrm{V/m/s^2}$). So the nominal value for $\hat{b}$ is $1/M = 37.037$. Since the true value of b may be unknown, $\hat{b}(0) = 20$ is used in the experiment. The adaptation rate

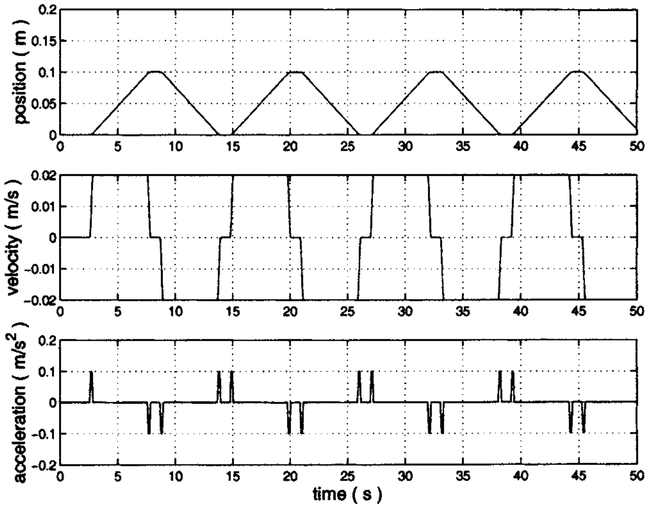

Fig. 3 Position, velocity, and acceleration of the desired trajectory

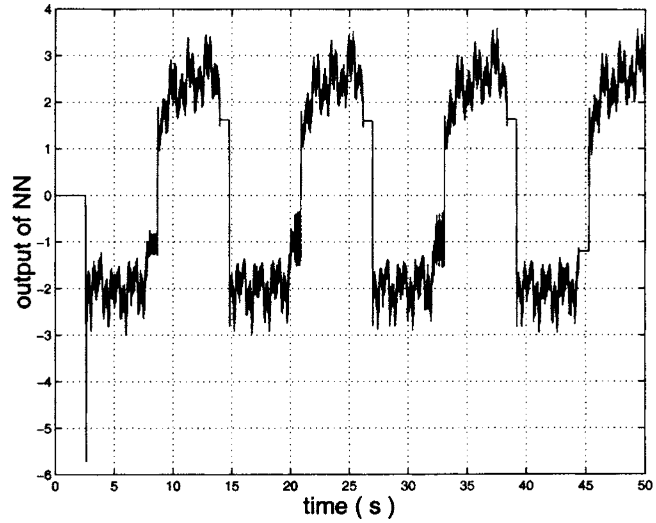

Fig. 5 NN output under output feedback NNARC (no filter)

is 6×10^2. The fictitious lower and upper bounds of θ are assumed the values of -11 and 0, the initial value is $\hat{\theta}(0)=-8.111$, and the adaptation rate is 2×10^2.

5.5 Experimental Results. It is known that the main nonlinearity in the model (41) is F_a and the force ripple may produce noticeable effects when the motor moves at a low speed [20]. In order to investigate how NNARC deals with this unknown nonlinearity, a low speed point-to-point desired trajectory with a maximum speed of 0.02 m/s is used, whose position, velocity, and acceleration are given in Fig. 3.

The experimental results under the output feedback NNARC are given in Figs. 4–9. From the result in Fig. 4, it is seen that the output tracks the desired trajectory very well. In fact, the tracking error mainly stays within ±2 μm except that the appearance of spikes when the direction of velocity changes. It is not surprising to see these spikes since NN can only approximate continuous functions to arbitrary accuracy and may not be able to handle discontinuous nonlinearity like Coulomb friction well. It is also observed that the tracking error during the entire run is within ±8.5 μm. Based on all these observations, it can be concluded that a high precision control of linear motor drive systems is achieved at low speed with output feedback only.

The output of the neural network is given in Figs. 5 and 6, respectively. Figure 5 shows that the output of NN has a discontinuous-jump-like shape when the velocity changes directions, which indicates the good approximation of the friction force at the low speed. To gain more knowledge about the shape of the NN output, a stable filter with transfer function $50/(s+50)$ is used to get rid of the noise effects. The filtered NN output is plotted versus the position as shown in Fig. 6. It can be seen that the filtered NN output tends to have a shape like the actual one in Fig. 7, which is cited from Ref. [20]. In terms of position x, the approximate period of the shape of the output of the NN is 0.03 m, which is exactly the same as the length between the centers of two adjacent permanent magnets of the motor.

In order to investigate how well the high-gain observer works, the "true" velocity is obtained by the backward difference of position measurement. The resolution of velocity feedback is (position resolution/sampling interval)$=10^{-6}/(4\times10^{-4})=0.0025$ m/s. The comparative plots are given in Fig. 8. It can be seen that the maximum error is about 0.005 m/s, which is acceptable since the resolution of "true" velocity is 0.0025 m/s. For completeness, the control input is shown in Fig. 9, which is

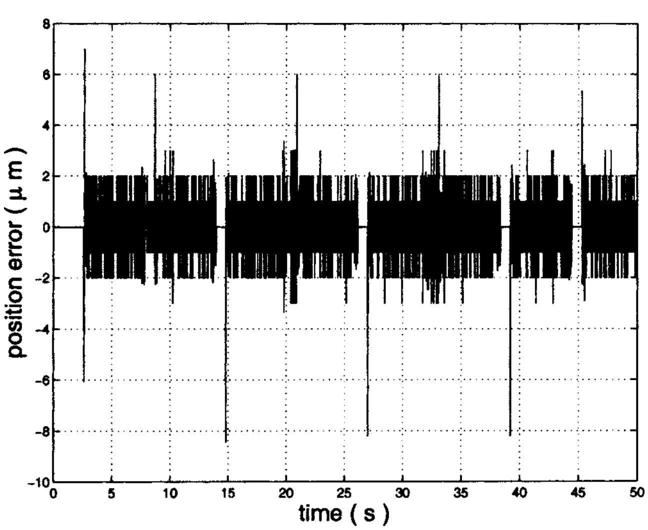

Fig. 4 Position error

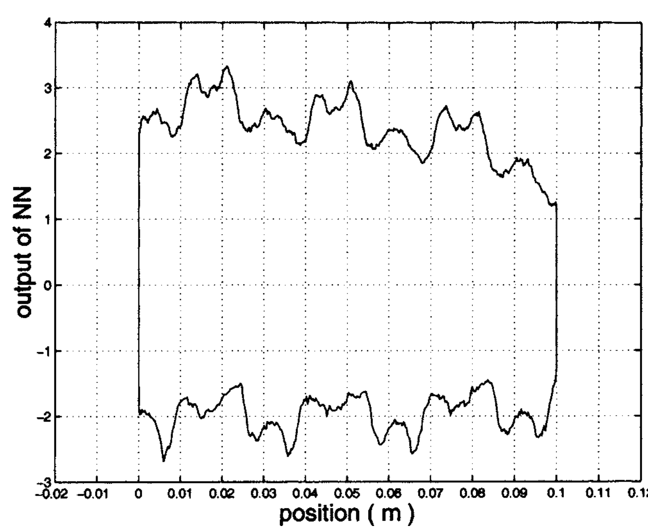

Fig. 6 NN output (with filter $50/(s+50)$) for the segment of $t=25.8248$ s to $t=38.9052$ s

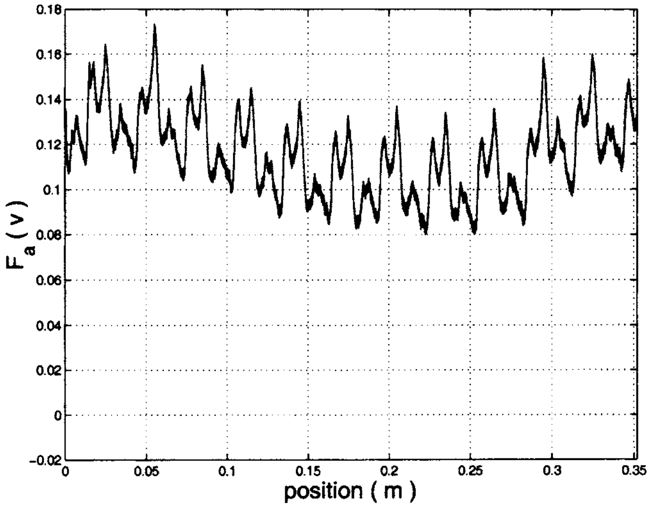

Fig. 7 ***F**$_a$* **versus position (constant velocity 0.017 m/s)**

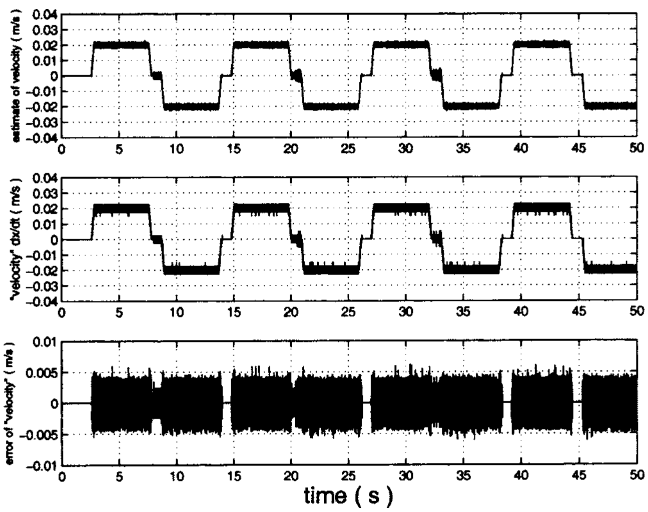

Fig. 8 Estimate of velocity and "true" velocity

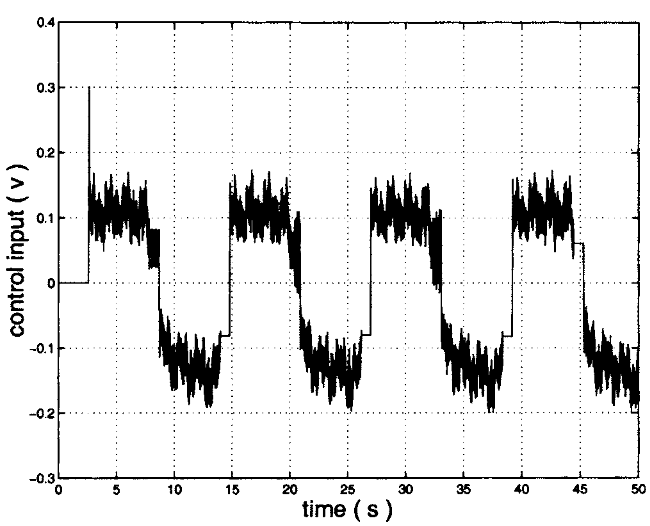

Fig. 9 Control input

clearly bounded, and is less than the saturation level. Overall, the proposed output feedback NNARC achieves an excellent output tracking performance even with little knowledge of the system and only the measurement of the position.

6 Conclusion

In this paper, the performance oriented NNARC laws have been constructed for a class of nth order SISO uncertain nonlinear systems. Both full state feedback and output feedback cases are investigated. The proposed NNARC laws take full advantage of both neural networks and ARC designs. The universal approximation capability of neural networks is utilized to approximate all unknown but repeatable nonlinear functions to achieve a better model compensation for an improved performance. All NN weights are tuned on-line with no prior training needed. Discontinuous projection mappings with fictitious bounds are used to achieve a controlled learning even in the presence of neural network approximation error and nonrepeatable nonlinearities such as external disturbances. Certain robust feedback is constructed to attenuate various model uncertainties effectively for a guaranteed output tracking transient performance and a guaranteed final tracking accuracy in general—a transient tracking performance that existing NN based robust adaptive controllers cannot achieve. The resulting full state feedback NNARC has the nice feature that if the unknown nonlinear functions are in the functional ranges of the neural networks and the ideal weights fall within the prescribed range, an asymptotic output tracking is also achieved—a performance that existing NN based robust adaptive controllers cannot achieve. By applying the proposed NNARC to a linear motor, the extensive experiments have been carried out to illustrate the effectiveness of the proposed strategy and a high resolution performance has been obtained with only output feedback available.

Acknowledgment

The work is supported by the National Science Foundation under CARREER Grant No. CMS-9734345.

Appendix A

According to Taylor-series expansion, $\boldsymbol{g}(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}})$ can be approximated by $\boldsymbol{g}(\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}})$ as follows

$$
\begin{aligned}
\boldsymbol{g} &= \check{\boldsymbol{g}} + \check{\boldsymbol{g}}'(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}) + \mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|) \\
&= \check{\boldsymbol{g}} + \check{\boldsymbol{g}}'(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \boldsymbol{V}\hat{\boldsymbol{x}}_{\{\text{in}\}} + \boldsymbol{V}\hat{\boldsymbol{x}}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}) + \mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|) \\
&= \check{\boldsymbol{g}} - \check{\boldsymbol{g}}'\tilde{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}} - \check{\boldsymbol{g}}'\boldsymbol{V}\tilde{\boldsymbol{x}}_{\{\text{in}\}} + \mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|) \qquad \text{(A1)}
\end{aligned}
$$

where $\mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|)$ denotes the sum of higher order terms.

Subsequently,

$$
\begin{aligned}
\boldsymbol{w}^T\boldsymbol{g}(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}}) &= \hat{\boldsymbol{w}}^T\boldsymbol{g}(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}}) - \tilde{\boldsymbol{w}}^T\boldsymbol{g}(\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}}) \\
&= \hat{\boldsymbol{w}}^T[\check{\boldsymbol{g}} - \check{\boldsymbol{g}}'\tilde{\boldsymbol{V}}\hat{\boldsymbol{x}}\{\text{in}\} - \check{\boldsymbol{g}}'\boldsymbol{V}\tilde{\boldsymbol{x}}_{\{\text{in}\}} + \mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|)] \\
&\quad - \tilde{\boldsymbol{w}}^T[\check{\boldsymbol{g}} - \check{\boldsymbol{g}}'\tilde{\boldsymbol{V}}\hat{\boldsymbol{x}}\{\text{in}\} - \check{\boldsymbol{g}}'\boldsymbol{V}\tilde{\boldsymbol{x}}_{\{\text{in}\}} + \mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|)] \\
&= \hat{\boldsymbol{w}}^T\check{\boldsymbol{g}} - \hat{\boldsymbol{w}}^T\check{\boldsymbol{g}}'\tilde{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}} - \tilde{\boldsymbol{w}}^T\check{\boldsymbol{g}} + \tilde{\boldsymbol{w}}^T\check{\boldsymbol{g}}'\tilde{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}} + \boldsymbol{w}[-\check{\boldsymbol{g}}'\boldsymbol{V}\tilde{\boldsymbol{x}}_{\{\text{in}\}} \\
&\quad + \mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|)] \\
&= \hat{\boldsymbol{w}}^T\check{\boldsymbol{g}} - \hat{\boldsymbol{w}}^T\check{\boldsymbol{g}}'\tilde{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}} - \tilde{\boldsymbol{w}}^T(\check{\boldsymbol{g}} - \check{\boldsymbol{g}}'\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}) - \tilde{\boldsymbol{w}}^T\check{\boldsymbol{g}}'\boldsymbol{V}\hat{\boldsymbol{x}}_{\{\text{in}\}} \\
&\quad + \boldsymbol{w}[-\check{\boldsymbol{g}}'\boldsymbol{V}\tilde{\boldsymbol{x}}_{\{\text{in}\}} + \mathcal{O}(\|\boldsymbol{V}\boldsymbol{x}_{\{\text{in}\}} - \hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}\|)] \\
&= \hat{\boldsymbol{w}}^T\check{\boldsymbol{g}} - \hat{\boldsymbol{w}}^T\check{\boldsymbol{g}}'\tilde{\boldsymbol{V}}\hat{\boldsymbol{x}}\{\text{in}\} - \tilde{\boldsymbol{w}}^T(\check{\boldsymbol{g}} - \check{\boldsymbol{g}}'\hat{\boldsymbol{V}}\hat{\boldsymbol{x}}_{\{\text{in}\}}) + \check{d}_{\text{NN}} \qquad \text{(A2)}
\end{aligned}
$$

which leads to Theorem 3.1.

Appendix B

From Eq. (A1), $-\breve{g}'V\tilde{x}_{\{in\}}+\mathcal{O}(\|Vx_{\{in\}}-\hat{V}\hat{x}_{\{in\}}\|)=g-\breve{g}+\breve{g}'\cdot(\tilde{V}\hat{x}_{\{in\}})$, therefore,

$$\begin{aligned}\|-\breve{g}'V\tilde{x}\{in\}+\mathcal{O}(\|Vx_{\{in\}}-\hat{V}\hat{x}_{\{in\}}\|)\| &= \|g-\breve{g}+\breve{g}'\cdot(\tilde{V}\hat{x}_{\{in\}})\| \\ &\leqslant \|g\|_2+\|\breve{g}\|_2+\|\breve{g}'\|_F\|\tilde{V}\|_F\|\hat{x}_{\{in\}}\|_2 \\ &\leqslant c_1+c_2\|\tilde{V}\|_F\|\hat{x}_{\{in\}}\|_2 \end{aligned} \quad \text{(B1)}$$

where c_1 and c_2 are bounds of $\|g\|_2+\|\breve{g}\|_2$, and $\|\breve{g}_f'\|_F$, respectively. The above inequality can be satisfied since it is true that activation function vector g_f and its derivatives are usually bounded.

Noting $\breve{d}_{NN}=-\tilde{w}^T\breve{g}'V\hat{x}_{\{in\}}+w^T[-\breve{g}'V\tilde{x}\{in\}+\mathcal{O}(\|Vx_{\{in\}}-\hat{V}\hat{x}_{\{in\}}\|)]$ from Eq. (B1), it is clear that

$$\begin{aligned}\|\breve{d}_{NN}\|_2 &= \|-\tilde{w}^T\breve{g}'V\hat{x}_{\{in\}}+w^T[-\breve{g}'V\tilde{x}\{in\}+\mathcal{O}(\|Vx_{\{in\}}-\hat{V}\hat{x}_{\{in\}}\|)]\| \\ &\leqslant \|\tilde{w}^T\breve{g}'V\hat{x}_{\{in\}}\|+\|w\|_F(c_1+c_2\|\tilde{V}\|_F\|\hat{x}_{\{in\}}\|) \\ &\leqslant \|w^T\breve{g}'V\tilde{x}\{in\}\|+\|\hat{w}^T\breve{g}'V\hat{x}_{\{in\}}\|+\|w\|_F(c_1+c_2\|\tilde{V}\|_F\|\hat{x}_{\{in\}}\|) \\ &\leqslant c_2\|w\|_F\|V\|_F\|\hat{x}_{\{in\}}\|_2+c_2\|V\|_F\|\hat{w}\|_F\|\hat{x}_{\{in\}}\|_2+c_1\|w\|_F \\ &\quad +c_2\|w\|_F\|V\|_F\|\hat{x}_{\{in\}}\|_2+c_2\|w\|_F\|\tilde{V}\|_F\|\hat{x}_{\{in\}}\|_2 \\ &= \breve{\alpha}^T\breve{Y} \end{aligned} \quad \text{(B2)}$$

where $\breve{\alpha}=[c_1\|w\|_F,\ 2c_2\|w\|_F\|V_f\|_F,\ c_2\|V\|_F,\ c_2\|w\|_F]^T$.

Since neither w nor V is known, $\breve{\alpha}$ is an unknown parameter vector. □

Appendix C

A. Consider a Lyapunov function candidate $V=(1/2)\hat{s}^2$. In viewing Eqs. (4) and (34) as well as the fact that $\breve{\alpha}_{fi}\geqslant 0$, and $\breve{\alpha}_{bi}\geqslant 0$, the time derivative of V is

$$\begin{aligned}\dot{V}=\hat{s}\dot{\hat{s}} &= -k\frac{b(x)}{b_l}\hat{s}^2-\frac{b(x)}{b_l}\breve{\alpha}_f\breve{Y}_f|\hat{s}|-\frac{b(x)}{b_l}\breve{\alpha}_b\breve{Y}_b|u_a||\hat{s}| \\ &\quad +\hat{s}\{[f(x)+a_r(t)]+b(x)u_a+\Delta+b(x)u_{s2}\} \\ &\leqslant -k\hat{s}^2+\epsilon_{so}\end{aligned} \quad \text{(C1)}$$

which leads to

$$\hat{s}^2(t)\leqslant \exp(-2kt)\hat{s}^2(0)+\frac{\epsilon_{so}}{k}[1-\exp(-2kt)]$$

or

$$|\hat{s}(t)|\leqslant\sqrt{\exp(-2kt)\hat{s}^2(0)+\frac{\epsilon_{so}}{k}[1-\exp(-2kt)]} \quad \text{(C2)}$$

By using the the third equation in Eq. (24), inequality (39) is obtained.

B. From inequality (39), it is seen that

$$\lim_{t\to\infty}|s|\leqslant\sqrt{\frac{\epsilon_{so}}{k}}+\varepsilon|\boldsymbol{\lambda}^T\boldsymbol{\psi}| \quad \text{(C3)}$$

Hence, the inequality (40) is obtained [27].

C. Using the same positive definite function as in A, it can be easily verified that $\dot{V}\leqslant -k\hat{s}^2+\epsilon_{so}$ since $-\hat{s}\nu(\hat{s})\leqslant 0$ holds. Hence, Result A remains valid, so does Result B.

D. When all the system states are available, the output feedback NNARC (25) changes to the state feedback NNARC (21), and the state estimates in all the adaptation laws changes to the actual states. Therefore, the result is the same as that in [19,20].

□

References

[1] Funahashi, K.-I., 1989, "On the Approximate Realization of Continuous Mappings by Neural Networks," Neural Networks, **2**, pp. 183–192.
[2] Hornik, K., 1991, "Approximation Capabilities of Multilayer Feedforward Networks," Neural Networks, **4**, pp. 251–257.
[3] Cybenko, G., 1989, "Approximation to Superpositions of Sigmoidal Function," Math. Control, Signals, Syst., **2**, pp. 303–314.
[4] Hunt, K. J., Sbarbaro, D., Zbikowski, R., and Gawthrop, P. J., 1992, "Neural Networks for Control Systems-A Survey," Automatica, **28**(6), pp. 1083–1112.
[5] Lewis, F. L., Yesidirek, A., and Liu, K., 1995, "Neural Net Robot Controller with Guaranteed Tracking Performance," IEEE Trans. Neural Netw., **6**, pp. 703–715.
[6] Polycarpou, M. M., 1996, "Stable Adaptive Neural Control Scheme for Nonlinear Systems," IEEE Trans. Autom. Control, **41**, pp. 447–451.
[7] Zhang, Y., and Peng, P. Y., 1999, "Stable Neural Controller Design for Unknown Nonlinear Systems using Backstepping," in *Proceedings of the American Control Conference*, San Diego, CA, pp. 1067–1071.
[8] Sanner, R. M., and Slotine, J.-J. E., 1992, "Gaussian Networks for Direct Adaptive Control," IEEE Trans. Neural Netw., **3**(6), pp. 837–863.
[9] Ge, S. S., Hang, C. C., and Zhang, T., 1999, "Neural-Based Direct Adaptive Neural Network Control of Nonlinear Systems by State and Output Feedback," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., **29**(6), pp. 818–828.
[10] Khalil, H. K., 1996, "Adaptive Output Feedback Control of Nonlinear Systems Represented by Input-Output Models," IEEE Trans. Autom. Control, **41**(2), pp. 177–188.
[11] Kanellakopoulos, I., Kokotovic, P. V., and Morse, A. S., 1991, "Adaptive Output-Feedback Control of Systems with Output Nonlinearrities," in *Foundations of Adaptive Control*, P. V. Kokotovic, ed., Springer-Verlag, Berlin, pp. 495–525.
[12] Esfandiari, F., and Khalil, H. K., 1992, "Output Fedback Stabilization of Fully Linearizable Systems," Int. J. Control, **56**(5), pp. 1007–1037.
[13] Zeitz, M., 1987, "The Extended Luenberger Observer for Nonlinear Systems," Syst. Control Lett., **9**, pp. 149–156.
[14] Ding, X., and Frank, P. M., 1990, "Nonlinear Observer Design via an Extended Observer Canonical Form," Syst. Control Lett., **15**, pp. 313–322.
[15] Lin, W., and Byrnes, C. I., 1995, "Remarks on Linearization of Discrete-Time Autonomous Systems and Nonlinear Observer Design," Syst. Control Lett., **25**, pp. 31–40.
[16] Yao, B., and Tomizuka, M., 1997, "Smooth Robust Adaptive Sliding Mode Control of Robot Manipulators with Guaranteed Transient Performance," in *Proceedings of the American Control Conference*, pp. 1176–1180; The full paper appeared in, 1996, ASME J. Dyn. Syst., Meas., Control, **118**(4) pp. 764–775.
[17] Yao, B., 1997, "High Performance Adaptive Robust Control of Nonlinear Systems: A General Framework and New Schemes," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2489–2494.
[18] Gong, J. Q., and Yao, B., 1999, "Adaptive Robust Control Without Knowing Bounds of Parameter Variations," in *Proceedings 38th of the IEEE Conference on Decision and Control, December 7–10, 1999*, Phoenix, AD, pp. 3334–3339.
[19] Gong, J. Q., and Yao, B., 2000, "Neural Network-Based Adaptive Robust Control of a Class of Nonlinear Systems in Normal Form," in *Proceedings of the Amercian Control Conference, June 28–30*, Chicago, IL, PP. 149–1423. (The revised full paper will appear in the Asian Journal of Control.)
[20] Gong, J. Q., and Yao, B., 2000, "Neural Network Adaptive Robust Control with Application to Precision Motion Control of Linear Motors," *International Journal of Adaptive Control and Signal Processing* (accepted for the special issue on Develpoments in Intelligent Control for Industrial Applications).
[21] Sastry, S., and Bodsin, M., 1989, *Adaptive Control: Stability, Convergence and Robustness, Prentice-Hall*, Englewood Cliffs, NJ.
[22] Goodwin, G. C., and Mayne, D. Q., 1989, "A Parameter Estimation Perspective of Continuous Time Model Reference Adaptive Control," Automatica, **23**(1), pp. 57–70.
[23] Polycarpou, M. M., and Loannou, P. A., 1992, "Neural Network as On-line Approximators of Nonlinear Systems," in *Proceedings of the 31st IEEE Conference on Decision and Control*, pp. 7–12.
[24] Behtash, S., 1990, "Robust Output Tracking for Nonlinear Systems," Int. J. Control, **51**(6), pp. 1381–1407.
[25] Utkin, V. I., 1997, "Variable Structure Systems with Sliding Modes," IEEE Trans. Autom. Control, **22**(2), pp. 212–222.
[26] Yao, B., Al-Majed, M., and Tomizuka, M., 1997, "High Performance Robust Motion Control of Machine Tools: An Adaptive Robust Control Approach and Comparative Experiments," IEEE/ASME Trans. Mechatron., **2**(2), pp. 63–76 (part of the paper also appeared in Proceedings of the 1997 American Control Conference).
[27] Slotine, J. J. E., and Li, W., 1991, *Applied Nonlinear Control, Prentice-Hall*, Englewood Cliffs, NJ.