INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2001; 15:837–864 (DOI: 10.1002/acs.658)

Neural network adaptive robust control with application to precision motion control of linear motors

J. Q. Gong and Bin Yao*,†

School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, U.S.A.

SUMMARY

In this paper, neural networks (NNs) and adaptive robust control (ARC) design philosophy are integrated to design performance-oriented control laws for a class of single-input–single-output (SISO) *n*th-order non-linear systems. Both repeatable (or state dependent) unknown non-linearities and non-repeatable unknown non-linearities such as external disturbances are considered. In addition, unknown non-linearities can exist in the control input channel as well. All unknown but repeatable non-linear functions are approximated by outputs of multi-layer neural networks to achieve a better model compensation for an improved performance. All NN weights are tuned on-line with no prior training needed. In order to avoid the possible divergence of the on-line tuning of neural network, discontinuous projection method with fictitious bounds is used in the NN weight adjusting laws to make sure that all NN weights are tuned within a prescribed range. By doing so, even in the presence of approximation error and non-repeatable non-linearities such as disturbances, a controlled learning is achieved and the possible destabilizing effect of on-line tuning of NN weights is avoided. 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. In addition, if the unknown repeatable model uncertainties are in the functional range of the neural networks and the ideal weights fall within the prescribed range, asymptotic output tracking is also achieved to retain the perfect learning capability of neural networks in the ideal situation. The proposed neural network adaptive control (NNARC) strategy is then applied to the precision motion control of a linear motor drive system to help to realize the high-performance potential of such a drive technology. NN is employed to compensate for the effects of the lumped unknown non-linearities due to the position dependent friction and electro-magnetic ripple forces. Comparative experiments verify the high-performance nature of the proposed NNARC. With an encoder resolution of 1 μm, for a low-speed back-and-forth movement, the position tracking error is kept within $\pm$ 2 μm during the most execution time while the maximum tracking error during the entire run is kept within $\pm$ 5.6 μm. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: neural network; adaptive control; robust control; linear motors

1. INTRODUCTION

Non-linearities in physical systems may appear in various forms. In general, it is difficult to treat various non-linearities under a unified framework. In some situations, due to the limited

* Correspondence to: Bin Yao, School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, U.S.A.
† E-mail: byao@ecn.purdue.edu

Contract/grant sponsor: National Science Foundation; contract/grant number: CMS-9734345

Published online 24 May 2001

knowledge about certain non-linear physical phenomena (e.g. friction, force ripple in a linear motor system), it is also impossible to precisely describe the non-linearities that can be used to capture those physical phenomena. These factors make it difficult to design high-performance controllers for non-linear systems.

The appearance of neural networks (NN) helps us advancing the design of high-performance controllers for general uncertain non-linear 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 [1–5]. It was shown in Reference [2] that the standard multi-layer feedforward networks with only a single hidden layer and arbitrary bounded and non-constant activation function are universal approximators with respect to $L^p(\mu)$ performance criteria, for arbitrary finite input environment measure μ, provided that a sufficient number of hidden neurons are available. In practice, the activation functions should be chosen based on different applications although sigmoidal type of functions and radial basis function (RBF) are usually used. The approximation capabilities of networks with sigmoidal function being activation function are discussed in Reference [3] while RBF networks are considered in References [4,5]. Due to their universal approximation capability, neural networks can be used to model certain complex non-linear physical phenomena effectively. It is thus of practical significance to use neural networks in the design of controller for uncertain non-linear systems.

In the research field of neural networks itself, focus is on the investigation of various NN characteristics, such as network structure, stability, convergence, and uniqueness of weights, etc. In fact, many works have been done on the stability analyses of a variety of neural networks [6–10] and the evolution of the weights of neural networks [11,12]. In all these papers, in order to guarantee the stability of neural networks and/or the uniqueness of the weights, the NN weights have to satisfy some restrictive conditions, which may limit the approximation capability of neural networks since weights can only be tuned in a relatively small region. Hence, researchers are still keeping on looking for NN structures with less restrictive conditions for the convergence of NN weights. Fortunately, when neural networks are used for control design purposes, the main focus is on the performance of the closed-loop system in terms of output tracking as long as all signals are bounded. Whether or not the NN weights converge to their ideal values may not be the key issue. As such, the NN weights can be tuned in a relatively larger region. Consequently, the approximation range of a neural network becomes larger, which is helpful in the control of non-linear systems when little is known about the non-linearities in the system. Thus, in this paper, not much attention will be paid to the convergence of weights of neural networks, and only the boundednesses of all the signals in neural networks are guaranteed.

Neural networks have been applied to the control field recently [13] and various results have been achieved [14–21]. A survey of the application of neural networks to control field was given in Reference [13], where modeling, identification, and control of non-linear systems via neural networks were discussed. The applications of NN to the control of robotic manipulators were detailed in Reference [14]. Two main issues have to be dealt with in the use of neural networks for non-linear control design. Firstly, the ideal weights of a neural network for approximating an unknown non-linear 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 non-linear functions. In terms of control terminology, adaptation laws are needed. Secondly, the ideal NN weights for the neutral network to reconstruct an unknown non-linear function exactly may not exist, i.e. the unknown non-linear 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 non-linear function cannot be assumed to be zero in general although it may be very small within a compact set. 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. In Reference [15], based on the assumption that both the input-hidden weights and the bounds of the hidden-output weights are known, backpropagation neural networks were used to design a robust adaptive controller (RAC) for multi-link rigid robots. In Reference [16], with the σ-modification type weight-tuning law, adaptive neural network control schemes were proposed for non-linear systems with uncertainties not satisfying matching conditions, where the input-hidden weights are also assumed to be known. Backstepping method was used in Reference [17] to design a neural network controller to guarantee the semi-global stability of the closed system. RBF networks were used in Reference [18] to adaptively compensate for the plant non-linearities, and the resulting adaptive controller achieves global stability and the final tracking accuracy. All these works are based on the assumption that the input-hidden weights of neural networks are known. It may be beneficial if this assumption can be relaxed so that one can fully explore the generality and flexibility of neural networks. In References [19,20], by using an NN controller to approximate the control law derived in the ideal case and introducing certain robust term, both transient performance and final tracking accuracy can be guaranteed in certain compact set. By using the same technique, the case of output feedback control was further discussed in Reference [21] with the aid of high-gain observer. Since the σ-modification type weight tuning method is used in References [15,16,19–21], asymptotic output tracking cannot be achieved even when the unknown non-linear function is in the functional range of the neural network. In other words, the ideal perfect learning capability of neural networks is lost. In addition, in References [15–17], transient tracking performance is in general not known. Transient period may be long and large transient tracking errors may exhibit.

Recently, the adaptive robust control (ARC) approach has been proposed in References [22–25] for non-linear systems in the presence of both parametric uncertainties and non-repeatable uncertain non-linearities 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 parametric uncertainties only, asymptotic output tracking is achieved. These strong performance results achieved by ARC controllers motivate us to investigate whether the essence 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 non-linear functions—a linear combination of known basis functions with unknown weights—have been considered in References [22–25], such an extension is also of significant theoretical values since a more general class of unknown functions can be dealt with via neural networks.

In this paper, neural networks and ARC design philosophy will be integrated to design a performance oriented control law, i.e. NNARC, for nth-order single-input–single-output (SISO) non-linear systems. Unknown non-linearities can exist in both system model and input channel, and could include non-repeatable non-linearities such as external disturbances as well. All unknown but repeatable non-linear functions (i.e. non-linear functions that depend on state only) will be approximated by the outputs of multi-layer neural networks to achieve a better model compensation for an improved performance. All NN weights are tuned on-line with no prior training needed. Discontinuous projection method with fictitious bounds [26] will be used to make sure that all NN weights are tuned within a prescribed range. By doing so, even in the presence of approximation error and non-repeatable non-linearities such as disturbances, 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 existing NN-based robust adaptive controllers [15–18] cannot achieve. In addition, if the unknown non-linear function is in the functional range of the neural network and the ideal weights fall within the prescribed range, 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 [15–21] cannot have.

The proposed NNARC is then applied to the precision motion control of linear motor drive systems. Linear motors offer several advantages over their rotary counterparts in many applications requiring linear motion. Usually, linear motions are realized by rotary motors with mechanical transmission mechanisms such as reduction gears and lead screw. Such mechanical transmissions not only significantly reduce linear motion speed and dynamic response, but also introduce backlash, large frictional and inertial loads, and structural flexibility. Backlash and structural flexibility physically limit the accuracy that any control system can achieve. As an alternative, direct drive linear motors, which eliminate the use of mechanical transmissions, show promise for widespread use in high-speed/high-accuracy positioning systems [27–29].

Although linear motors gain their high-performance potential by eliminating mechanical transmissions, it also loses the advantage of using mechanical transmissions—gear reduction reduces the effect of model uncertainties such as parameter variations (e.g. uncertain payloads) and external disturbances (e.g. cutting forces in machining). Furthermore, certain types of linear motors such as the iron core have significant uncertain non-linearities due to electro-magnetic force ripple and magnetic cogging force [29]. These uncertain non-linearities are directly transmitted to the load and thus have significant effect on the motion of the load. Thus, to realize the high-performance potential of a linear motor system, a controller which can achieve the required high accuracy in spite of various parametric uncertainties and uncertain non-linear effects, has to be employed.

A great deal of effort has been devoted to solving the difficulties in controlling linear motors [27–32]. In Reference [27], Alter and Tsao presents a comprehensive design approach for the control of linear motor driven machine tool axes. H_∞ optimal feedback control is used to provide high dynamic stiffness to external disturbances (e.g. cutting forces in machining). Feedforward is also introduced in Reference [28] to improve tracking performance. Practically, H_∞ design may be conservative for high-speed/high-accuracy tracking control and there is no systematic way to translate practical information about plant uncertainty and modelling inaccuracy into quantitative terms that allow the application of H_∞ techniques. In Reference [30], a disturbance compensation method based on disturbance observer (DOB) [33,31] was proposed to make the linear motor system robust to model uncertainties. It was shown both theoretically and experimentally in Reference [34] that DOB design may not handle discontinuous disturbances such as Coulomb friction well and cannot deal with large extent of parametric uncertainties. To reduce the non-linear effect of force ripple and cogging force, in Reference [29], feedforward compensation terms, which are based on an off-line experimentally identified model of first-order approximation of ripple force, were added to the position controller. Since neither all magnets in a linear motor nor all linear motors of the same type are identical, feedforward compensation based on off-line identification model may be too sensitive and costly to be useful.

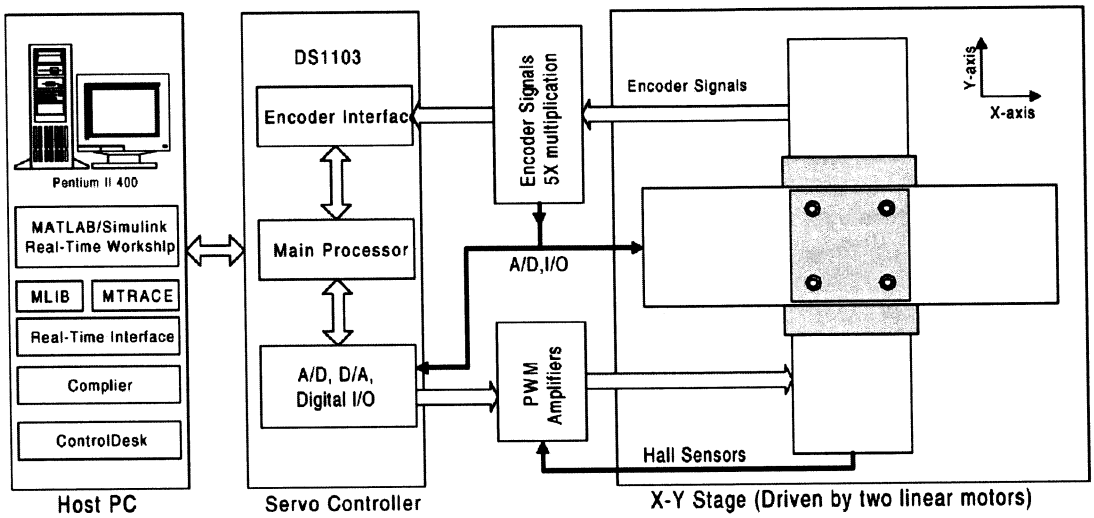

Figure 1. Experimental setup.

To overcome the shortcomings in the above controller designs, in this paper, under the proposed NNARC strategy, NN is used to adaptively compensate for the lumped effect of various unknown non-linearities on-line. Furthermore, particular structure of the major non-linearities associated with the linear motor drive system is fully explored to construct a low-dimensional NN with three types of hidden neurons for a tractable and yet effective NN implementation. Experimental results on the control of an epoxy core linear motor show that the proposed NN is able to compensate for the effect of various non-linearities in the system very quickly and accurately; both transient and final tracking errors are in the same order as the encoder resolution level of 1 μm.

The rest of the paper is organized as follows. The linear motor model is presented in Section 2 to provide a motivation for the class of SISO non-linear systems considered in the paper. The proposed NNARC is detailed in Section 3. Comparative experimental studies are carried out on the control of an epoxy core linear motor in Section 4 to illustrate the effectiveness of the proposed NNARC strategy. Conclusions are drawn in Section 5.

2. LINEAR MOTOR MODEL AND PROBLEM FORMULATION

Linear servo motors essentially work the same as rotary motors, only opened up and laid out flat as shown in Figure 1. Each motor is made of two parts—a permanent magnet assembly and a coil assembly. The coil assembly encapsulates copper windings within a core material (e.g. epoxy, steel). The copper windings conduct current. The magnet assembly consists of rare earth magnets, mounted in alternating polarity on a steel plate, which generates magnetic flux density. By applying a three-phase current to the coil assembly, a sequence of attracting and repelling forces between the poles and the permanent magnets will be generated. This results in a thrust force being experienced by the coil assembly. Ideally, this force is linear in phase currents and

independent of position. However, in practice, the fluctuations of the field distribution and the armature MMF, which depend on the current waveform and motor structure, give rise to position-dependent force ripple [29,32,35].

In motion control, force ripple is a major source of uncertainty which varies with the coil assembly position, thus causing tracking error. At very high speeds, they are of high frequency and are usually filtered out by the system inertia. But at low speeds, they produce noticeable effects that may not be tolerable [29,32]. Another major source of uncertainty which may degrade the performance of any motion control system is *friction force*, which includes stiction, Coulomb friction, Stribeck effect and viscous friction.

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 current dynamics are neglected. The mathematical model of the system can thus be described by

$$M\ddot{x} = u - F_f - F_r(x, \dot{x}) + F_d \tag{1}$$

where x represents the position of the inertia load, M is the normalized[‡] mass of the inertia load plus the coil assembly, u is the input voltage to the motor, F_f is the normalized friction, F_r represents the normalized electro-magnetic force ripple, and F_d represents the normalized external disturbance force (e.g. cutting force in machining). While there have been many friction models proposed [36], a simple and often adequate approach is to regard friction force as a static non-linear function of the velocity

$$F_f(\dot{x}) = B\dot{x} + F_{fn}(\dot{x}) \tag{2}$$

where B is the equivalent viscous friction coefficient of the system, and F_{fn} is the non-linear friction term modelled as [36,37]

$$F_{fn}(\dot{x}) = [f_c + (f_s - f_c)e^{-|\dot{x}/\dot{x}_s|^{\xi}}]\,\mathrm{sgn}(\dot{x}) \tag{3}$$

in which f_s represents the level of static friction, f_c is the Coulomb friction, and $\dot{x}_s$ and ξ are empirical parameters used to describe the Stribeck effect. In practice, due to the inaccuracy of the positioning stage and ball bearings, the friction force may depend on the position x also, which is verified by the experimental results shown later. With this position-dependent friction force in mind, noting (2), the linear motor system (1) can be rewritten as

$$\ddot{x} = -\frac{B}{M}\dot{x} - \frac{1}{M}F_a(x, \dot{x}) + \frac{1}{M}u + \Delta(x, \dot{x}, t) \tag{4}$$

where $F_a \triangleq F_{fn}(x, \dot{x}) + F_r(x, \dot{x})$ is the lumped non-linear force which depends on the state only, and $\Delta \triangleq (1/M)F_d$ is the equivalent external disturbance, which may be state dependent and time-varying.

3. NNARC DESIGN

As discussed in Section 2, it is difficult to obtain a precise model of the lumped non-linear force F_a. Therefore, NN will be employed to estimate this unknown non-linearity. However, the

[‡] Normalized with respect to the unit input voltage.

conventional NN designs may have a slow on-line tuning speed and may be sensitive to disturbances and modelling errors, which usually results in a poor transient performance and final tracking accuracy that could not meet the high precision requirements of modern drive technologies. In viewing the excellent tracking performance achieved by the ARC design [22,23,34], it is natural for us to investigate whether the ARC design methodology can be integrated with the universal approximation capability of NN to maximize the achievable performance of a controlled system. Such an integrated design is presented in this section for a class of SISO nth-order non-linear systems in normal form [38].

3.1. *Problem formulation*

In order to make the results applicable for a broader class of systems motivated by the linear motor model (4), the following nth-order SISO non-linear system is considered [18,38]

$$x^{(n)} = \phi^{\mathrm{T}}(\mathbf{x}, t)\theta + f(\mathbf{x}) + b(\mathbf{x})u(t) + \Delta(\mathbf{x}, t) \tag{5}$$

where x is the system output, $\mathbf{x} = [x, x^{(1)}, \ldots, x^{(n-1)}]^{\mathrm{T}}$ is the state variable vector with $x^{(i)}$ denoting the ith time derivative of the output x, $\phi(\mathbf{x}, t) = [\phi_1(\mathbf{x}, t), \ldots, \phi_r(\mathbf{x}, t)]^{\mathrm{T}}$ is the vector of known basis functions, $\theta = [\theta_1, \ldots, \theta_r]^{\mathrm{T}}$ is the constant unknown parameter vector, $\phi^{\mathrm{T}}(x, t)\theta$ denotes the structured non-linearity [26], $f(\mathbf{x})$ represents the unstructured state-dependent (or repeatable) unknown non-linearity, $b(\mathbf{x})$ is the unknown non-linear input gain, $u(t)$ is the system input, and $\Delta(\mathbf{x}, t)$ represents the lumped *non-repeatable* (or *time-dependent*) non-linearities such as disturbances.

Remark 1

It is seen that the linear motor model (4) is in the form of (5) with $n = 2$, $\phi(\mathbf{x}, t) = \dot{x}$, $\theta = -B/M$, $f(\mathbf{x}) = -(1/M)F_a$, $b(\mathbf{x}) = 1/M$, and $\Delta(\mathbf{x}, t) = \Delta(x, \dot{x}, t)$.

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

Assumption 1

The NN approximation error associated with the non-linear function $f(\mathbf{x})$ is assumed to be bounded [2], i.e.

$$|f(\mathbf{x}) - \mathbf{w}_f^{\mathrm{T}}\mathbf{g}_f(\mathbf{V}_f\mathbf{x}_a)| \leqslant d_f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathscr{R}^n \tag{6}$$

where $d_f(\mathbf{x}) \geqslant 0$ is the approximation error, $\mathbf{x}_a = [\mathbf{x}^{\mathrm{T}}, -1]^{\mathrm{T}}$ is the augmented input vector to the neural network (-1 term denotes the input bias), $\mathbf{w}_f = [w_{f1}, \ldots, w_{fr_f}]^{\mathrm{T}}$ is the hidden-output weight vector, $\mathbf{V}_f = [\mathbf{V}_{f1}, \ldots, \mathbf{V}_{fr_f}]^{\mathrm{T}} \in \mathscr{R}^{r_f \times (n+1)}$ is the input-hidden weight matrix with $\mathbf{v}_{fi} \in \mathscr{R}^{(n+1)\times 1}$, r_f is the number of hidden neurons and $\mathbf{g}_f(\mathbf{V}_f\mathbf{x}_a) = [g_{f1}(\mathbf{V}_{f1}^{\mathrm{T}}\mathbf{x}_a), \ldots, g_{fr_f}(\mathbf{V}_{fr_f}^{\mathrm{T}}\mathbf{x}_a)]^{\mathrm{T}}$ is the activation function vector.

Remark 2

From the theorems in References [1,2], non-linearity $f(\mathbf{x})$ can be approximated by the output of a multi-layer neural network to an arbitrarily accuracy on a compact set $\mathscr{A}_f$ provided that the

number of neurons is sufficiently large, i.e.

$$|f(\mathbf{x}) - \mathbf{w}_f^{\mathrm{T}}\mathbf{g}_f(\mathbf{V}_f\mathbf{x}_a)| \leqslant \eta_f, \quad \forall \mathbf{x} \in \mathscr{A}_f \tag{7}$$

where η_f is an arbitrarily small positive number, and $\mathscr{A}_f$ is a compact subset of $\mathscr{R}^n$. Correspondingly, in Assumption 1, d_f can be arbitrarily small when $\mathbf{x} \in \mathscr{A}_f$. Outside the compact set $\mathscr{A}_f$, the difference between the output of the neural network and the true value of the non-linear function may not be made arbitrarily small. It is however reasonable to expect that the approximation error outside the compact set $\mathscr{A}_f$ is bounded by a known non-linear function as assumed in Assumption 1.

Since θ is constant, the following trivial assumption is made.

Assumption 2

The elements of the system parameter vector θ are bounded although their bounds may not be known, i.e.

$$\rho_{l,\theta_i} \leqslant \theta_i \leqslant \rho_{u,\theta_i}, \quad i = 1, 2, \ldots, r \tag{8}$$

where $\rho_{l,i}$ and $\rho_{u,i}$ represents the lower and upper bound of θ_i, respectively, which may be unknown.

Remark 3

As a matter of fact, the structured non-linearity $\phi^{\mathrm{T}}(\mathbf{x}, t)\theta$ can be viewed as the output of a two-layer (i.e. no hidden-layer) neural network with $\mathbf{x}$, t being the inputs, ϕ being the activation function vector, and θ being the weight vector. However, since $\phi^{\mathrm{T}}(\mathbf{x}, t)$ may explicitly depend on time t, the structured non-linearity $\phi^{\mathrm{T}}(\mathbf{x}, t)\theta$ is separated from the unstructured non-linearity $f(\mathbf{x})$, which depends on the states only.

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

Assumption 3

The input gain $b(\mathbf{x})$ is non-zero with known sign. Thus, without loss of generality, assume

$$b(\mathbf{x}) \geqslant b_l > 0, \quad \forall \mathbf{x} \in \mathscr{R}^n \tag{9}$$

where b_l is a known positive constant.

Similar to Assumption 1, a neural network will be used to approximate the non-linear input gain $b(\mathbf{x})$. Thus, the following assumption is made:

Assumption 4

The non-linear input gain $b(\mathbf{x})$ can be approximated by the output of a multi-layer neural network with

$$|b(\mathbf{x}) - \mathbf{w}_b^{\mathrm{T}}\mathbf{g}_b(\mathbf{V}_b\mathbf{x}_a)| \leqslant d_b(\mathbf{x}), \quad \forall \mathbf{x} \in \mathscr{R}^n \tag{10}$$

where $d_b(\mathbf{x})$ is a non-negative function of $\mathbf{x}$, $\mathbf{w}_b$, $\mathbf{g}_b$ and $\mathbf{V}_b$ are defined in similar ways as $\mathbf{w}_f$, $\mathbf{g}_f$ and $\mathbf{V}_f$ in Assumption 1, respectively. It is assumed that the number of neurons in the hidden-layer is r_b. The dimensions of $\mathbf{w}_b$, $\mathbf{g}_b$ and $\mathbf{V}_b$ are decided accordingly. Similar to Remark 2, $d_b(\mathbf{x})$ can be made arbitrarily small on some compact set $\mathcal{A}_b$.

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 5

The non-repeatable non-linearity Δ is bounded by

$$|\Delta| \leqslant h_\Delta(\mathbf{x}, t)d(t) \tag{11}$$

where $h_\Delta(\mathbf{x}, t)$ is a known function and $d(t)$ is an unknown, but bounded positive time-varying function.

For any sufficiently smooth desired output trajectory $x_d(t)$, the desired state trajectory is $\mathbf{x}_d(t) = [x_d, x_d^{(1)}, \ldots, x_d^{(n-1)}]^T$. *The control objective is to design a control law for u such that the system state variable vector* $\mathbf{x}$ *tracks* $\mathbf{x}_d$ *as closely as possible.* If the tracking error vector is defined as $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_d(t)$, the control objective is equivalent to make the 'size' of $\tilde{\mathbf{x}}(t)$ as small as possible.

In the following derivations, $\hat{\theta} = [\hat{\theta}_1, \ldots, \hat{\theta}_r]^T$ represents the estimate of the system parameters, $\theta = \hat{\theta} - \theta$ is the estimation error, $\mathbf{w}_f = [\hat{w}_{f1}, \ldots, \hat{w}_{fr_f}]^T$ represents the estimate of the hidden-output weight vector, $\tilde{\mathbf{w}}_f = \hat{\mathbf{w}}_f - \mathbf{w}_f$ is the estimation error of the hidden-output weight vector, $\hat{\mathbf{V}}_f = [\mathbf{V}_{f1}, \ldots, \hat{\mathbf{V}}_{fr_f}]^T$ is the estimate of the input-hidden weight matrix, $\tilde{\mathbf{V}}_f = \hat{\mathbf{V}}_f - \mathbf{V}_f$ is the corresponding estimation error matrix, and $\mathbf{g}_f$ is the shorthand notation for $\mathbf{g}_f(\mathbf{V}_f \mathbf{x}_a)$. Notations $\hat{\mathbf{w}}_b$, $\tilde{\mathbf{w}}_b$, $\hat{\mathbf{V}}_b$, $\tilde{\mathbf{V}}_b$ and $\mathbf{g}_b$ are defined in a same way.

3.2. *Discontinuous projection mapping*

Since all the weights in neural networks are constants, it is true that each element of $\mathbf{w}_f$, $\mathbf{w}_b$, $\mathbf{V}_f$, and $\mathbf{V}_b$ is bounded, i.e.

$$\rho_{l,w_{fi}} \leqslant w_{fi} \leqslant \rho_{u,w_{fi}}, \quad \rho_{l,v_{fij}} \leqslant v_{fij} \leqslant \rho_{u,v_{fij}}, \quad i = 1, \ldots, r_f, \; j = 1, \ldots, n+1 \tag{12}$$

$$\rho_{l,w_{bi}} \leqslant w_{bi} \leqslant \rho_{u,w_{bi}}, \quad \rho_{l,v_{bij}} \leqslant v_{bij} \leqslant \rho_{u,v_{bij}}, \quad i = 1, \ldots, r_b, \; j = 1, \ldots, n+1 \tag{13}$$

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 may not be known *a priori*, certain fictitious bounds have to be used [26]. Let $\hat{\rho}_{l,\star_{ij}}$ and $\hat{\rho}_{u,\star_{ij}}$ be the fictitious lower and upper bound for $\star_{ij}$, where $\star$ could represent any of the unknown parameter vectors θ, $\mathbf{w}_f$, and $\mathbf{w}_b$ or matrices $\mathbf{V}_f$ and $\mathbf{V}_b$, respectively. Based on these fictitious lower and upper bounds, a discontinuous projection mapping $\mathbf{Proj}(\cdot)$ can be defined as follows [39,40]:

$$\mathbf{Proj}_{\hat{\star}}(\cdot) = \{Proj_{\hat{\star}}(\cdot_{ij})\} \tag{14}$$

with its ijth element being

$$Proj_{\hat{\bigstar}}(\bullet_{ij}) = \begin{cases} 0 & \text{if } \begin{cases} \hat{\bigstar}_{ij} = \hat{\rho}_{u,\bigstar_{ij}} & \text{and } \bullet_{ij} > 0 \\ \text{or} & \\ \hat{\bigstar}_{ij} = \hat{\rho}_{l,\bigstar_{ij}} & \text{and } \bullet_{ij} < 0 \end{cases} \\ \bullet_{ij} & \text{otherwise} \end{cases} \tag{15}$$

The discontinuous projection mapping defined above will be used in the construction of adaptation law for the estimate of $\bigstar$. For simplicity of notations, define $\hat{\rho}_{\bigstar_{ij}} = \max\{|\hat{\rho}_{l,\bigstar_{ij}}|, |\hat{\rho}_{u,\bigstar_{ij}}|\}$, and denote $\hat{\rho}_{\bigstar} = \{\hat{\rho}_{\bigstar_{ij}}\}$ and $\rho_{\bigstar} = \{\rho_{\bigstar_{ij}}\}$.

3.3. *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 $\mathbf{x}_{\{in\}} \in \mathcal{R}^{p+1}$ being its input vector, $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{r_n}]^T \in \mathcal{R}^{r_n \times (p+1)}$ being its input-hidden weight matrix, $\mathbf{g}$ being the activation function vector, $\mathbf{w} \in \mathcal{R}^{m \times r_n}$ being its hidden-output weight matrix, we have the following theorem [41]:

Theorem 3.1

The output of neural network $\mathbf{w}^T\mathbf{g}(\mathbf{V}\mathbf{x}_{\{in\}})$ can be approximated by its estimate $\hat{\mathbf{w}}^T\mathbf{g}(\hat{\mathbf{V}}\mathbf{x}_{\{in\}})$ by the following form:

$$\mathbf{w}^T\mathbf{g}(\mathbf{V}\mathbf{x}_{\{in\}}) = \hat{\mathbf{w}}^T\hat{\mathbf{g}} - \tilde{\mathbf{w}}^T(\hat{\mathbf{g}} - \hat{\mathbf{g}}'\hat{\mathbf{V}}\mathbf{x}_{\{in\}}) - \hat{\mathbf{w}}^T\hat{\mathbf{g}}'\tilde{\mathbf{V}}\mathbf{x}_{\{in\}} + d_{NN} \tag{16}$$

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

Proof. Refer to Reference [41]. □

Lemma 3.1

The residual term d_{NN} can be bounded by a linear-in-parameter function [41], i.e.

$$|d_{NN}| \leqslant \alpha^T\mathbf{Y} \tag{17}$$

where α is an unknown vector constituting positive elements and the known function vector $\mathbf{Y}$ is defined as follows:

$$\mathbf{Y} = [1, \ \|\mathbf{x}_{\{in\}}\|_2, \ \|\hat{\mathbf{w}}\|_F\|\mathbf{x}_{\{in\}}\|_2, \ \|\hat{\mathbf{V}}\|_F\|\mathbf{x}_{\{in\}}\|_2]^T \tag{18}$$

where $\|\bullet\|_F$ denotes the Frobenius norm of a matrix $\bullet$, which is defined as $\|\bullet\|_F^2 = \text{Trace}\{\bullet^T\bullet\}$.

Proof. Refer to Reference [41]. □

From Theorem 3.1 and Lemma 3.1, the following two equations are obtained:

$$\mathbf{w}_f^T\mathbf{g}_f(\mathbf{V}_f\mathbf{x}_a) = \hat{\mathbf{w}}_f^T\hat{\mathbf{g}}_f - \tilde{\mathbf{w}}_f^T(\hat{\mathbf{g}}_f - \hat{\mathbf{g}}'_f\hat{\mathbf{V}}_f\mathbf{x}_a) - \hat{\mathbf{w}}_f^T\hat{\mathbf{g}}'_f\tilde{\mathbf{V}}_f\mathbf{x}_a + d_{fNN} \tag{19}$$

$$\mathbf{w}_b^T\mathbf{g}(\mathbf{V}_b\mathbf{x}_a) = \hat{\mathbf{w}}_b^T\hat{\mathbf{g}}_b - \tilde{\mathbf{w}}_b^T(\hat{\mathbf{g}}_b - \hat{\mathbf{g}}'_b\hat{\mathbf{V}}_b\mathbf{x}_a) - \hat{\mathbf{w}}_b^T\hat{\mathbf{g}}'_b\tilde{\mathbf{V}}_b\mathbf{x}_a + d_{bNN} \tag{20}$$

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

$$|d_{f\mathrm{NN}}| \leqslant \alpha_f^{\mathrm{T}} Y_f, \quad Y_f = [1, \|x_a\|_2, \|\hat{w}_f\|_2\|x_a\|_2, \|\hat{V}_f\|_F\|x_a\|_2]^{\mathrm{T}} \tag{21}$$

$$|d_{b\mathrm{NN}}| \leqslant \alpha_b^{\mathrm{T}} Y_b, \quad Y_b = [1, \|x_a\|_2, \|\hat{w}_b\|_2\|x_a\|_2, \|\hat{V}_b\|_F\|x_a\|_2]^{\mathrm{T}} \tag{22}$$

with $\|\bullet\|_2$ being the 2-norm of a vector $\bullet$. It is noted that $\|\bullet\|_F = \|\bullet\|_2$ for a vector $\bullet$.

3.4. *NNARC design*

Although it is usually assumed that the input-hidden weights are known [15] and it can be achieved by 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. In the following, for simplicity, the sigmoid function will be used as activation functions. Other type of activation functions (e.g. RBF [18], bipolar sigmoid function [15]) can be worked out in the same way as long as the activation functions and their derivatives are bounded functions.

Since the control objective is to force x to track x_d, a concise tracking error metric can be defined as [18]:

$$s(t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} x(\tilde{t}) \quad \text{with } \lambda > 0 \tag{23}$$

where λ is a positive constant. (23) can be rewritten as $s(t) = \lambda^{\mathrm{T}}\tilde{x}(t)$ with the ith element of vector λ being given by $C_{n-1}^{i-1}\lambda^{n-i} = ((n-1)!/(n-i)!(i-1)!)\lambda^{n-i}$. The equation $s(t) = 0$ defines a time-varying hyperplane in $\mathscr{R}^n$ on which the tracking error vector decays exponentially to zero. Thus, the perfect tracking can be asymptotically achieved by maintaining this condition [42].

Consider the control law [38]

$$u = u_a + u_s \tag{24}$$

with

$$u_a = -\frac{1}{\hat{w}_b^{\mathrm{T}}\hat{g}_b}[a_r(t) + \phi^{\mathrm{T}}(x, t)\hat{\theta} + w_f^{\mathrm{T}}\hat{g}_f]$$

$$u_s = u_{s1} + u_{s2}, u_{s1} = -\frac{1}{b_l}ks - \frac{1}{b_l}[\hat{\alpha}_f^{\mathrm{T}}Y_f + \hat{\alpha}_b^{\mathrm{T}}Y_b|u_a|]\,\mathrm{sgn}(s) \tag{25}$$

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

Using control law (24), the following error equation can be obtained:

$$\begin{aligned}
\dot{s} &= x^{(n)} + a_r(t) \\
&= \phi^{\mathrm{T}}(x, t)\theta + [f(x) + a_r(t)] + b(x)u_a + b(x)u_s + \Delta \\
&= [\phi^{\mathrm{T}}(x, t)\theta - \phi^{\mathrm{T}}(x, t)\hat{\theta}] + [f(x) - w_f^{\mathrm{T}}g_f] + [w_f^{\mathrm{T}}g_f - \hat{w}_f^{\mathrm{T}}\hat{g}_f] \\
&\quad + [b(x) - w_b^{\mathrm{T}}g_b]u_a + [w_b^{\mathrm{T}}g_b - \hat{w}_b^{\mathrm{T}}\hat{g}_b]u_a + \Delta + b(x)u_{s2} + b(x)u_{s1} \\
&= -\phi^{\mathrm{T}}(x, t)\tilde{\theta} + [f(x) - w_f^{\mathrm{T}}g_f + d_{f\mathrm{NN}}] + [-\tilde{w}_f^{\mathrm{T}}(\hat{g}_f - \hat{g}_f'\hat{V}_f x_a) - \hat{w}_f^{\mathrm{T}}\hat{g}_f'\tilde{V}_f x_a] \\
&\quad + \{[b(x) - w_b^{\mathrm{T}}g_b + d_{b\mathrm{NN}}] + [-\tilde{w}_b^{\mathrm{T}}(\hat{g}_b - \hat{g}_b'\hat{V}_b x_a) - \hat{w}_b^{\mathrm{T}}\hat{g}_b'\tilde{V}_b x_a]\}u_a \\
&\quad - k\frac{b(x)}{b_l}s - \frac{b(x)}{b_l}\hat{\alpha}_f^{\mathrm{T}}Y_f\,\mathrm{sgn}(s) - \frac{b(x)}{b_l}\hat{\alpha}_b^{\mathrm{T}}Y_b|u_a|\mathrm{sgn}(s) + b(x)u_{s2} + \Delta
\end{aligned} \tag{26}$$

where Equations (19) and (20) are used in the derivation of the fourth equality.

In viewing the fourth equality of (26), by using the discontinuous projection mapping in Section 3.2, the following adaptation laws are constructed [38]:

$$\dot{\hat{\theta}} = \mathbf{Proj}_{\hat{\theta}}(\Gamma_\theta s \phi(\mathbf{x}, t)) \tag{27}$$

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

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

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

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

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

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

where Γ_*'s are diagonal positive-definite adaptation rate matrices. In (30) and (33), the lower fictitious bounds for $\hat{\alpha}_{fi}$ and $\hat{\alpha}_{bi}$ are chosen to be zero since α_{fi} and α_{bi} are positive from Lemma 3.1. It is also assumed that the fictitious lower and upper bounds $\hat{\rho}_{l,w_{bi}}$ and $\hat{\rho}_{u,w_{bi}}$ used in projection mapping $\mathbf{Proj}_{\hat{\mathbf{w}}_b}\{\bullet\}$ for $\hat{w}_{bi}$, and $\hat{\rho}_{b,v_{bij}}$ and $\hat{\rho}_{u,v_{bij}}$ in $\mathbf{Proj}_{\hat{\mathbf{v}}_b}\{\bullet\}$ for $\hat{v}_{bij}$ are chosen in such a way that $\hat{\mathbf{w}}_b^{\mathrm{T}} \hat{\mathbf{g}}_b > 0$, $\forall \hat{w}_{bi} \in [\hat{\rho}_{l,w_{bi}}, \hat{\rho}_{u,w_{bi}}]$, $\forall \hat{v}_{bij} \in [\hat{\rho}_{l,v_{bij}}, \hat{\rho}_{u,v_{bij}}]$; this requirement is reasonable since the ideal value of $\mathbf{w}_b^{\mathrm{T}} \mathbf{g}_b$ is $b(\mathbf{x})$, which is greater than zero as assumed in Assumption 3.

Using similar arguments as in Reference [22], it can be shown that the above adaptation laws with projection mapping have the following nice properties [38]:

P1. Each component of parameter estimate is always within its known fictitious bound, i.e.

$$\hat{\rho}_{l,\theta_i} \leqslant \hat{\theta}_i \leqslant \hat{\rho}_{u,\theta_i}, \quad i = 1, \ldots, r \tag{34}$$

$$\hat{\rho}_{l,w_{fi}} \leqslant \hat{w}_{fi} \leqslant \hat{\rho}_{u,w_{fi}}, \quad i = 1, \ldots, r_f \tag{35}$$

$$\hat{\rho}_{l,v_{fij}} \leqslant \hat{v}_{fij} \leqslant \hat{\rho}_{u,v_{fij}}, \quad i = 1, \ldots, r_f, \; j = 1, \ldots, n+1 \tag{36}$$

$$0 \leqslant \hat{\alpha}_{fk} \leqslant \hat{\rho}_{u,\alpha_{fk}}, \quad k = 1, \ldots, 4 \tag{37}$$

$$\hat{\rho}_{l,w_{bi}} \leqslant \hat{w}_{bi} \leqslant \hat{\rho}_{u,w_{bi}}, \quad i = 1, \ldots, r_b \tag{38}$$

$$\hat{\rho}_{l,v_{bij}} \leqslant \hat{v}_{bij} \leqslant \hat{\rho}_{u,v_{bij}}, \quad i = 1, \ldots, r_b, \; j = 1, \ldots, n+1 \tag{39}$$

$$0 \leqslant \hat{\alpha}_{bk} \leqslant \hat{\rho}_{u,\alpha_{bk}}, \quad k = 1, \ldots, 4 \tag{40}$$

P2. In addition, if ideal weights actually lie within their fictitious bounds, then

$$\tilde{\theta}^{\mathrm{T}}(\Gamma_\theta^{-1} \mathbf{Proj}_{\hat{\theta}}(\Gamma_\theta \bullet) - \bullet) \leqslant 0, \quad \forall \bullet \tag{41}$$

$$\tilde{\mathbf{w}}_f^{\mathrm{T}}(\Gamma_{wf}^{-1} \mathbf{Proj}_{\hat{\mathbf{w}}_f}(\Gamma_{wf} \bullet) - \bullet) \leqslant 0, \quad \forall \bullet \tag{42}$$

$$\mathrm{Trace}\{\tilde{\mathbf{V}}_f(\Gamma_{vf}^{-1} \mathbf{Proj}_{\hat{\mathbf{V}}_f}(\Gamma_{vf} \bullet) - \bullet)\} \leqslant 0, \quad \forall \bullet \tag{43}$$

$$\tilde{\boldsymbol{\alpha}}_f^{\mathrm{T}}(\Gamma_{\alpha f}^{-1} \mathbf{Proj}_{\hat{\alpha}_f}(\Gamma_{\alpha f} \bullet) - \bullet) \leqslant 0, \quad \forall \bullet \tag{44}$$

$$\tilde{\mathbf{w}}_b^{\mathrm{T}}(\Gamma_{wb}^{-1} \mathbf{Proj}_{\hat{\mathbf{w}}_b}(\Gamma_{wb} \bullet) - \bullet) \leqslant 0, \quad \forall \bullet \tag{45}$$

$$\mathrm{Trace}\{\tilde{\mathbf{V}}_b(\Gamma_{vb}^{-1} \mathbf{Proj}_{\hat{\mathbf{V}}_b}(\Gamma_{vb} \bullet) - \bullet)\} \leqslant 0, \quad \forall \bullet \tag{46}$$

$$\tilde{\boldsymbol{\alpha}}_b^{\mathrm{T}}(\Gamma_{\alpha b}^{-1} \mathbf{Proj}_{\hat{\alpha}_b}(\Gamma_{\alpha b} \bullet) - \bullet) \leqslant 0, \quad \forall \bullet \tag{47}$$

Remark 4

Through the use of discontinuous projection mappings in the adaptation laws, the weight estimates are always bounded and stay within the fictitious bounds as seen from Property P1. Thus, a controlled on-line weight tuning process is achieved and the possible divergence problem of on-line weight-tuning in the presence of disturbances and approximation errors is avoided. Furthermore, the bound of the output of a neural network can be predicted to certain degree, which enables certain robust control terms to be constructed in the design of control law for a guaranteed transient performance and final tracking accuracy in general as shown later.

The robust control term u_{s2} is synthesized to satisfy the following conditions [38]:

$$s\{\phi^{\mathrm{T}}\theta + [f(\mathbf{x}) + a_r(t)] + b(\mathbf{x})u_a + \Delta + b(\mathbf{x})u_{s2}\} \leqslant \varepsilon_s \tag{48}$$

$$su_{s2} \leqslant 0 \tag{49}$$

where

$$\varepsilon_s = \left|1 + \frac{|(\|\rho_\theta\|_2 - \|\hat{\rho}_\theta\|_2)|}{\|\hat{\rho}_\theta\|_2}\right|^2 \varepsilon_1 + \left|1 + \frac{|(\|\rho_{wf}\|_2 - \|\hat{\rho}_{wf}\|_2)|}{\|\hat{\rho}_{wf}\|_2}\right|^2 \varepsilon_2 + \left|1 + \frac{|(\|\rho_{wb}\|_2 - \|\hat{\rho}_{wb}\|_2)|}{\|\hat{\rho}_{wb}\|_2}\right|^2 \varepsilon_3 + \varepsilon_4 + \|d\|_\infty^2 \varepsilon_5 \tag{50}$$

in which ε_1, ε_2, ε_3, ε_4, and ε_5 are the positive design constants.

Remark 5

The robust term u_{s2} in (24) may be chosen in the following way. Let

$$u_{s2} = -k_{s2}s \tag{51}$$

where k_{s2} is a non-linear gain large enough such that $k_{s2} \geqslant \sum_{i=1}^{i=5} h_i^2/4\varepsilon_i$ in which h_1, h_2, h_3, h_4 and h_5 are any non-linear bounding functions satisfying $h_1 \geqslant (1/\sqrt{b_l})\|\phi(\mathbf{x}, t)\|(\|\hat{\theta}\|_2 + \|\hat{\rho}_\theta\|_2)$, $h_2 \geqslant (1/\sqrt{b_l})\|\hat{\mathbf{g}}_f\|(\|\hat{\mathbf{w}}_f\| + \|\hat{\rho}_{wf}\|)$, $h_3 \geqslant (1/\sqrt{b_l})\|\hat{\mathbf{g}}_b\|(\|\hat{\mathbf{w}}_b\| + \|\hat{\rho}_{wb}\|)|u_a|$, $h_4 \geqslant (1/\sqrt{b_l})[d_f + |u_a|d_b]$, and $h_5 \geqslant (1/\sqrt{b_l})h_\Delta(x, t)$, respectively, $\|\phi\| = \sqrt{\sum_{i=1}^{i=r}|\phi_i|^2}$, $\|\hat{\mathbf{g}}_f\| = \sqrt{\sum_{i=1}^{i=r_f}\|\hat{g}_{fi}\|_\infty^2}$ and $\|\hat{\mathbf{g}}_b\| = \sqrt{\sum_{i=1}^{i=r_b}\|\hat{g}_{bi}\|_\infty^2}$. The detailed proof of u_{s2} given by (51) satisfying (48) and (49) is given in Reference [38] and is omitted.

Usually, the activation functions g_{fi}'s and g_{bi}'s are sigmoid functions; hence, $|\hat{g}_{fi}| \leqslant 1$. It is also clear that $\|\hat{\mathbf{w}}_f\|_2 \leqslant \|\hat{\rho}_{wf}\|_2$. Therefore, we can simply choose $h_2 = 2\sqrt{r_f/b_l}\|\hat{\rho}_{wf}\|_2$ in practice. Similarly, h_3 can be chosen as $h_3 = 2\sqrt{r_b/b_l}\|\hat{\rho}_{wb}\|_2|u_a|$.

Theorem 3.2

With NNARC (24) and the adaptation laws (27)–(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 above by

$$s^2(t) \leqslant \exp(-2kt)s^2(0) + \frac{\varepsilon_s}{k}[1 - \exp(-2kt)] \leqslant \exp(-2kt)s^2(0) + \frac{\varepsilon_s}{k} \tag{52}$$

B. By setting the initial state vector of the system equal to the initial state vector of the desired trajectory, i.e. $\mathbf{x}(0) = \mathbf{x}_d(0)$, $s(0) = 0$ can be obtained. Consequently, the actual tracking error is

asymptotically bounded by

$$\|\tilde{x}^{(i)}\|_\infty \leqslant 2^i \lambda^{i-n+1} \sqrt{\frac{\varepsilon_s}{k}} \tag{53}$$

C. If $f(\mathbf{x}) = \mathbf{w}_f^T \mathbf{g}_f(\mathbf{V}_f \mathbf{x}_a)$, and $b(\mathbf{x}) = \mathbf{w}_b^T \mathbf{g}_b(\mathbf{V}_b \mathbf{x}_a)$, [3], i.e. the non-linear 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, asymptotic output tracking is achieved provided that there is no external disturbance (i.e. $\Delta = 0$), and all true parameters and weights lie within the fictitious bounds (i.e. $\hat{\rho}_{l,\theta_i} \leqslant \theta_i \leqslant \hat{\rho}_{u,\theta_i}$, $\forall i = 1, \ldots, r$, $\hat{\rho}_{l,w_{fi}} \leqslant w_{fi} \leqslant \hat{\rho}_{u,w_{fi}}$, $\hat{\rho}_{l,v_{fij}} \leqslant v_{fij} \leqslant \hat{\rho}_{u,v_{fij}}$, $\forall i = 1, \ldots, r_f$, $\forall j = 1, \ldots, n+1$, $0 \leqslant \alpha_{fk} \leqslant \hat{\rho}_{u,\alpha_{fk}}$, $k = 1, \ldots, 4$, $\hat{\rho}_{l,w_{bi}} \leqslant w_{bi} \leqslant \hat{\rho}_{u,w_{bi}}$, $\hat{\rho}_{l,v_{bij}} \leqslant v_{bij} \leqslant \hat{\rho}_{u,v_{bij}}$, $\forall i = 1, \ldots, r_b$, $\forall j = 1, \ldots, n+1$, $0 \leqslant \alpha_{bk} \leqslant \hat{\rho}_{u,\alpha_{bk}}$, $k = 1, \ldots, 4$).

D. When the discontinuous sign function sgn(s) in (25) is dropped or replaced by any continuous function $\psi(s)$ satisfying $s\psi(s) \geqslant 0$, Results A and B still remain valid.

Proof. See the appendix. □

Remark 6

Results A and B of Theorem 3.2. 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 ($\|\tilde{x}\|_\infty \leqslant \lambda^{-n+1}\sqrt{\varepsilon_s/k}$) are related to the design parameters $k, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ and ε_5 in a known form, and can be adjusted freely by suitably choosing those design parameters. These results are thus much stronger than those in References [15–17]. In all those schemes in References [15–17], transient performance is not guaranteed.

Remark 7

Result C of Theorem 3.2 shows that the proposed NNARC is able to accomplish its learning goal (the assumptions in C of Theorem 3.2 represent 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 [15–21] 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 (15) 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 3.2, the fictitious parameter ranges should be chosen larger enough so that the resulting NN is able to approximate the unknown non-linearities well to obtain asymptotic output tracking performance stated in the Result C of Theorem 3.2. However, the same as in Reference [26], this should not be overdone when the control input has limited authority and the system may be occasionally subjected to large disturbances as in most applications; it has been observed in References [34,26] 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 [34,26], 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.

Remark 9

As mentioned in Remark 3, $\phi^T\theta$ may be viewed as the output of a two-layer neural network with ϕ being the activation function vector and θ being the weight vector. From this point of view, the adaptation law (27) is compatible with (28).

3.5. *Trajectory initialization and generation*

It is seen from (52) that transient tracking error is affected by the initial value $s(0)$. To further reduce transient tracking error, the idea of trajectory initialization [22,43] can be utilized. Namely, instead of simply letting the desired trajectory for the controller be the actual reference trajectory (i.e. $x_d(t) = x_r(t)$), $x_d(t)$ is generated using a stable filter. For example, $x_d(t)$ can be generated by the following nth-order stable filter

$$x_d^{(n)} + \sum_{i=0}^{n-1} \beta_{n-i} x_d^{(i)} = x_r^{(n)} + \sum_{i=0}^{n-1} \beta_{n-i} x_r^{(i)} \tag{54}$$

with the initial conditions given by $\mathbf{x}_d(0) = \mathbf{x}(0)$ and $\beta_i > 0$, $i = 0, \ldots, n-1$. By choosing β_i properly, x_d can track x_r with any prescribed transient. At the same time, $s(0) = 0$ is achieved, and the transient tracking error is reduced.

4. COMPARATIVE EXPERIMENTAL STUDY

4.1. *Experimental set-up*

To test the proposed NNARC strategy and study fundamental problems associated with high-speed/high-accuracy motion control of linear motor drive systems, a two-axis X–Y positioning stage is set up as a test-bed. As shown in Figure 1, the test-bed 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 of $f_s = 2.5$ kHz. Since backward difference of position measurement is used to obtain velocity, the resolution of velocity feedback is position resolution/sampling interval $= 10^{-6}/4 \times 10^{-4} = 0.0025$ m/s.

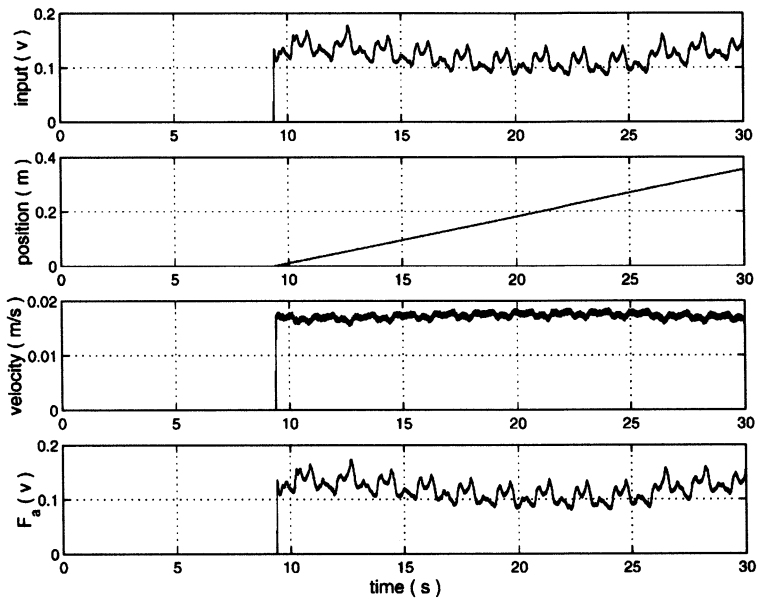

Figure 2. Histories of the input, position, velocity and the lumped non-linear force F_a (constant velocity 0.017 m/s).

4.2. *Design of neural network and adaptation law*

Off-line system identification reveals that the epoxy core linear motor used in the system can be adequately described by (4). However, it is observed that it is very hard to obtain the exact form of the lumped non-linear force F_a. As such, F_a will be approximated by an NN as follows.

Although it is well known that better performance may be expected with more neurons, longer computation time results, which is not suitable for real-time implementation. Therefore, in the following, the particular properties of the linear motor system will be used to construct a low dimensional neural network for a tractable and yet effective NN implementation. For this purpose, simple tests are first run to gain some insights on the nature of F_a as follows. Specifically, to find out how F_a depends on the position x, a simple velocity-feedback control law is used to try to maintain a constant velocity movement as shown in Figure 2. Since $\Delta = 0$, F_a can be calculated from (4) as $F_a = u - Mx - B\dot{x}$, in which $M = 0.027$ and $B = 0.273$ are the nominal values of M and B obtained by off-line identification. The resulting F_a is shown in Figure 2 and is plotted versus the position x in Figure 3. As shown, F_a depends on the position significantly. The position dependency of F_a may be explained as follows. Firstly, for linear motors, the contacting condition between the moving coil assembly and the fixed base changes with the position x. Thus, the level of friction may change slowly with the position, which contributes to the non-periodic low frequency portion of F_a in terms of the position x shown in Figure 3. The high-frequency portion of F_a in terms of the position x is quite periodic with a period of 0.03 m. The period is exactly the same as the length between the centers of two adjacent permanent magnets of the motor, which indicates that this high-frequency portion is due to the electro-magnetic force ripple F_r.

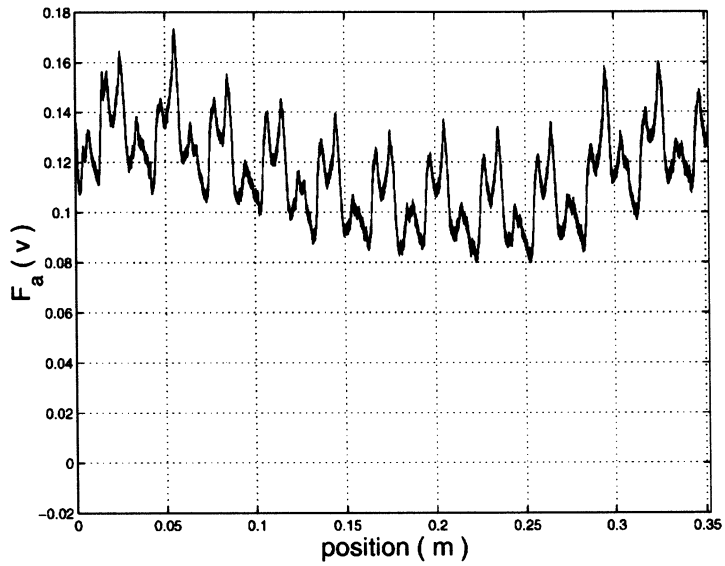

Figure 3. F_a vs position (constant velocity 0.017 m/s).

Based on the above observations, the NN designed for $f(\mathbf{x}) = -(1/M)F_a$ should depend on both position x and velocity $\dot{x}$. Hence, the input vector to the NN is $[x, \dot{x}, -1]^T$. Three types of neurons are used in the hidden layer to take into account the particular phenomenon observed above.

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 ($\sin(2\pi/0.03x)$, $\sin(6\pi/0.03x)$, and $\sin(10\pi/0.03x)$) 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 $[x, \dot{x}, -1]^T$, 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.

For clarity, the structure of the above NN is also given in Figure 4. Since the input-hidden weights are fixed in the last two types of hidden neurons, the corresponding adaptation laws can be simplified to

$$\dot{\hat{\mathbf{w}}}_f = \mathbf{Proj}_{\hat{\mathbf{w}}_f}(\Gamma_{wf} s \mathbf{g}_f) \tag{55}$$

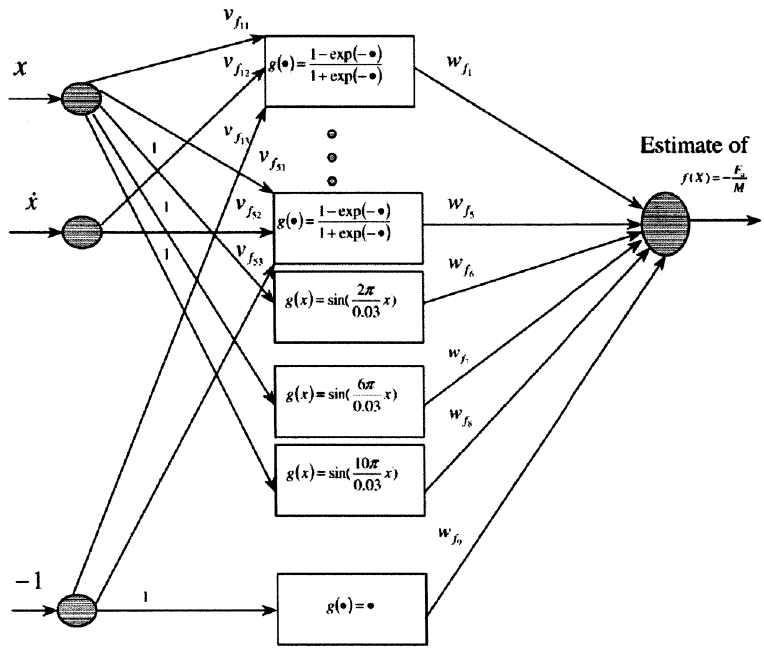

Figure 4. NN structure for estimating $-F_a/M$.

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

From (4), it is clear that the input gain $1/M$ is a constant. This property will be taken into account to simplify the NN for $b(\mathbf{x})$ by letting g_b to be identity function. Consequently, there is no hidden layer in NN for $b(\mathbf{x})$, and only $\hat{w}_b$ (i.e. $\hat{b}$) needs to be adjusted. Correspondingly, adaptation laws (32) and (33) are not needed, and adaptation law (31) becomes

$$\dot{\hat{w}}_b = \dot{\hat{b}} = \mathbf{Proj}_{\hat{b}}\{\Gamma_b u_a s\} \tag{56}$$

4.3. DRC and NNARC control laws

Since no hidden layer exists in NN for $b(\mathbf{x})$, the robust control term $\hat{\alpha}_b^{\mathrm{T}}\mathbf{Y}_b|u_a|\mathrm{sgn}(s)$ in (25) is not needed. Furthermore, as stated in Result D of Theorem 3.2, $(2/\pi)\arctan(\psi_a s)$ can be used to replace the sign function in (25) to avoid control input chattering. The resulting u_{s1} is

$$u_{s1} = -\frac{1}{b_l}\left[ks + \frac{2}{\pi}\hat{\alpha}_f^{\mathrm{T}}\mathbf{Y}_f \arctan(\psi_a s)\right] \tag{57}$$

in which $\psi_a = 90$ is used in the experiments.

In the experiments, comparative study between a traditional deterministic robust controller (DRC) and the proposed NNARC is carried out.

For reference, the NNARC for the linear motor system is given as follows:

$$u_{\text{nnarc}} = -\frac{a_r(t) + \hat{\theta}\dot{x} + \hat{\mathbf{w}}_f^{\mathrm{T}}\hat{\mathbf{g}}_f}{\hat{b}} - \frac{1}{b_l}\left[ks + \frac{2}{\pi}\hat{\alpha}_f^{\mathrm{T}}\mathbf{Y}_f \arctan(\psi_a s)\right] + u_{s2} \tag{58}$$

where u_{s2} assumes the form (51) with $k_{s2} = \max(3, \sum_{i=1}^{i=5} h_i^2/4\varepsilon_i)$.

As in Reference [26], when neither the NN weights nor unknown parameters are adjusted, a DRC law is obtained. Corresponding to (58), the following DRC is proposed:

$$u_{drc} = -\frac{a_r(t) + \hat{\theta}(0)\dot{x} + \hat{\mathbf{w}}_f^{\mathrm{T}}(0)\hat{\mathbf{g}}_f(0)}{\hat{b}(0)} - \frac{1}{b_l}\left[ks + \frac{2}{\pi}\hat{\alpha}_f^{\mathrm{T}}(0)\mathbf{Y}_f \arctan(\psi_a s)\right] + u_{s2} \tag{59}$$

Considering the system model (4), under the control law (58), the following dynamic equation is obtained:

$$\begin{aligned}\dot{s} = &-\frac{1}{M}\left(B + \frac{\hat{\theta}}{\hat{b}}\right)\dot{x} - \frac{1}{M}\left(F_a + \frac{\hat{\mathbf{w}}_f^{\mathrm{T}}\hat{\mathbf{g}}_f}{\hat{b}}\right) + \left(1 - \frac{1}{M\hat{b}}\right)a_r(t) \\ &- \left(\frac{k}{Mb_l} + \frac{k_{s2}}{M}\right)s - \frac{\hat{\alpha}_f^{\mathrm{T}}\mathbf{Y}_f}{Mb_l}\mathrm{sat}\left(\frac{s}{\psi}\right) + \Delta\end{aligned} \tag{60}$$

where the first three terms on the right-hand side of the equation represent the model compensation parts.

4.4. Parameters in control laws and adaptation laws

For comparison, the parameters for both controllers are assumed the same if they have the same meanings. $\lambda = 200$ is used for the sliding plane and $k = 20$ is used as the constant gain. The final tracking accuracy indices are $\varepsilon_1 = 5000$, $\varepsilon_2 = 5000$, $\varepsilon_3 = 5000$, $\varepsilon_4 = 5000$ and $\varepsilon_5 = 5000$.

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, 10^4 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 10^3, 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 2×10^5.

The initial values of all the weights are zeros.

Since $0.025 \leqslant M \leqslant 0.1$, $b_l = 10$ results. Standard least-squares identification is performed to obtain the parameters of the system. The nominal values of M is 0.027 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

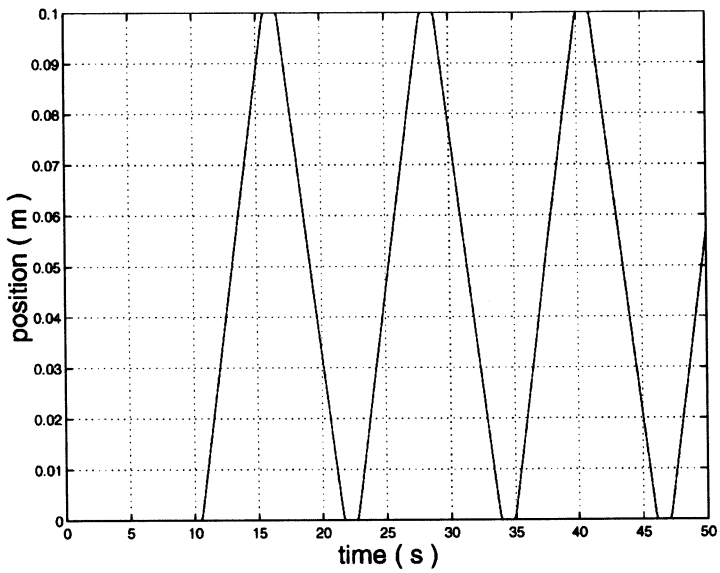

Figure 5. Position of the desired trajectory.

experiment. The adaptation rate is 5000. 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 4×10^4.

4.5. Experimental results

As stated in Section 2, the main non-linearity in the model (4) is F_a and the force ripple may produce noticeable effects when the motor moves at a low speed. In order to investigate how NNARC deals with this unknown non-linearity, a low speed point-to-point desired trajectory with a maximum speed of 0.02 m/s is used, whose position and velocity of the initial several periods are given in Figures 5 and 6, respectively.

As in Section 3.5, a second-order stable filter with $\beta_1 = 100$ and $\beta_2 = 2500$ is used to further reduce the transient error.

The experimental results under two controllers are given in Figures 7–13. From the result in Figure 7, a large tracking error exhibits for DRC due to the 'wrong' initial values of the parameter estimates. Comparatively, in Figure 8, the output under NNARC tracks the desired trajectory very well. In fact, the tracking error mainly stays within $\pm$ 2 μm, except that the appearance of spikes when the direction of velocity changes. It is not surprise to see these spikes since NN can only approximate continuous functions to arbitrary accuracy and may not be able to handle discontinuous non-linearity like Coulomb friction well. However, it can be seen that the magnitudes of these spikes keep decreasing until reaching to a small value of $\pm$ 2.5 μm. It is also observed that the tracking error during the entire run is within $\pm$ 5.6 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.

The time histories of the estimate $\hat{\theta}$ and $\hat{b}$ are given in Figure 9. Since the desired trajectory is not rich enough and the persistent excitation condition cannot be satisfied, the estimates 'drift

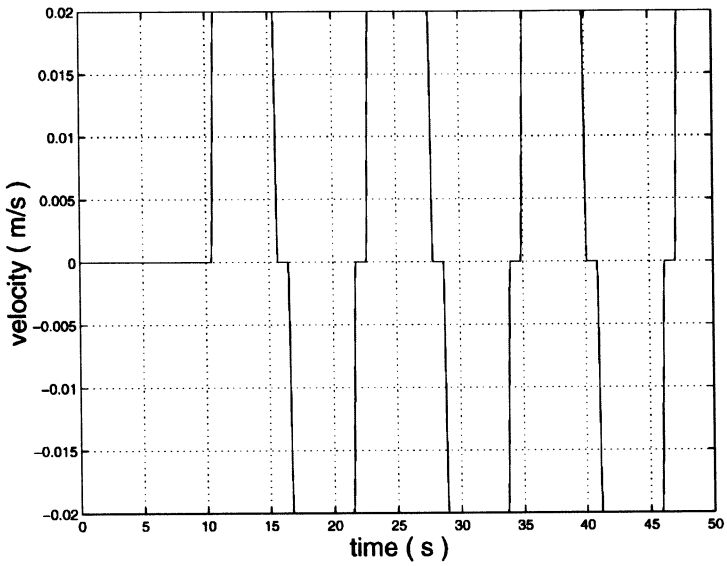

Figure 6. Velocity of the desired trajectory.

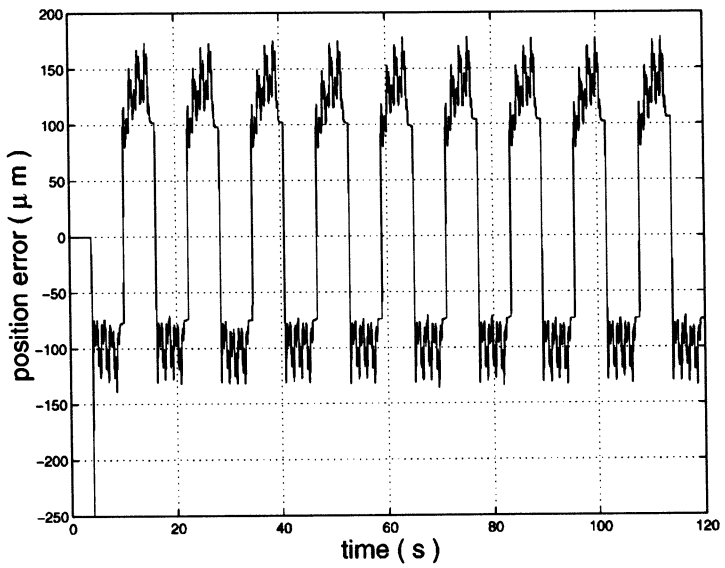

Figure 7. Position error under DRC.

away'. Fortunately, through the use of projection mappings, the estimates cannot go unbounded. Both of them are kept within the fictitious bounds. In this sense, a controlled learning is achieved.

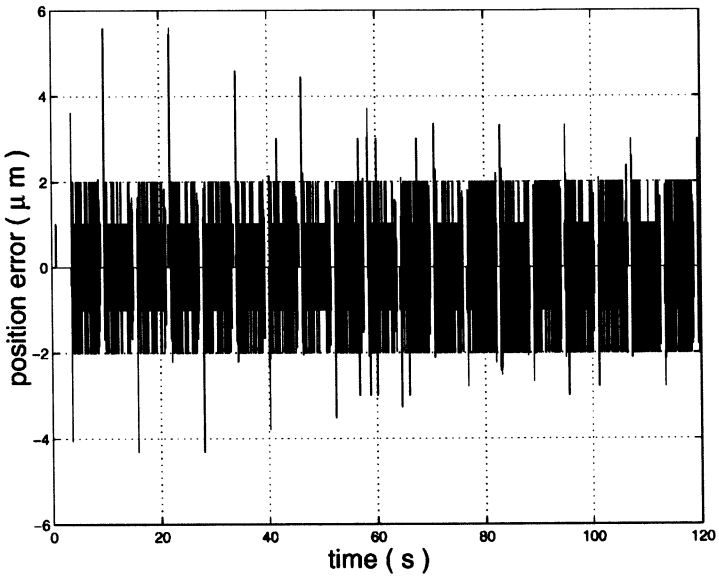

Figure 8. Position error under NNARC.

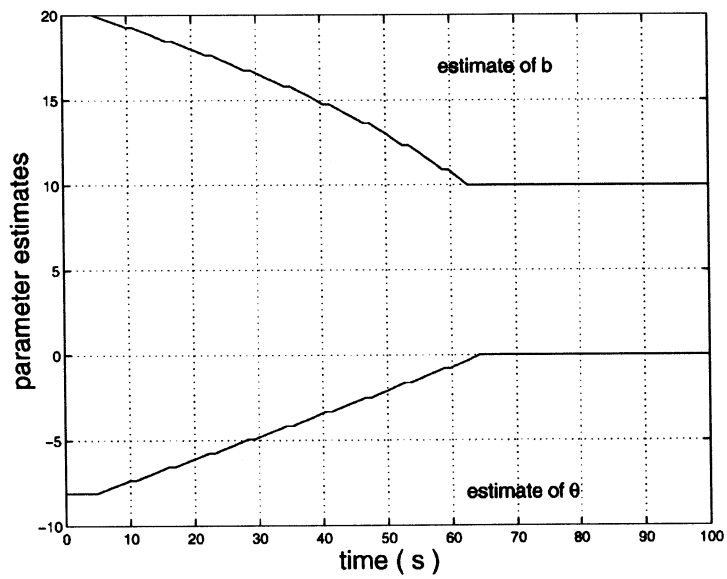

Figure 9. Estimates of b and θ under NNARC.

The output of the neural network is given in Figures 10 and 11, respectively. As seen from Figure 10, 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

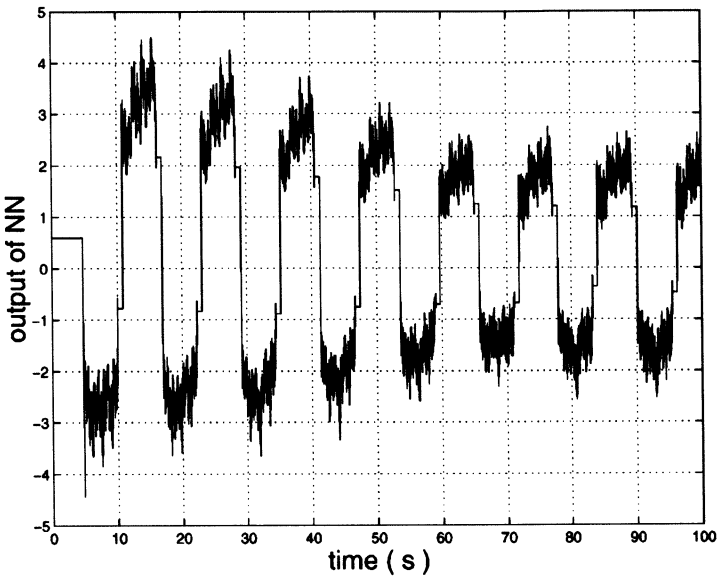

Figure 10. NN output under NNARC (no filter).

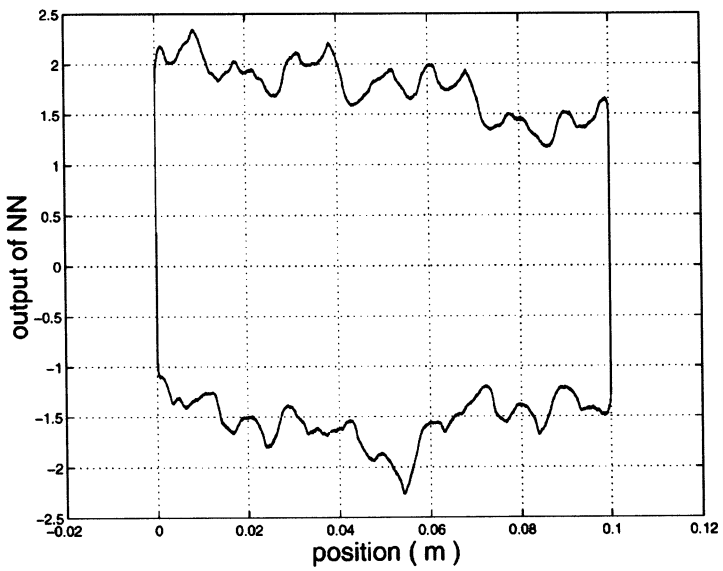

Figure 11. NN output under NNARC (with filter $50/(s + 50)$) for the segment of $t = 79.9996$ 92.1780 s.

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 Figure 11. It can be seen that the filtered NN output tends to have a shape like the actual one in Figure 3.

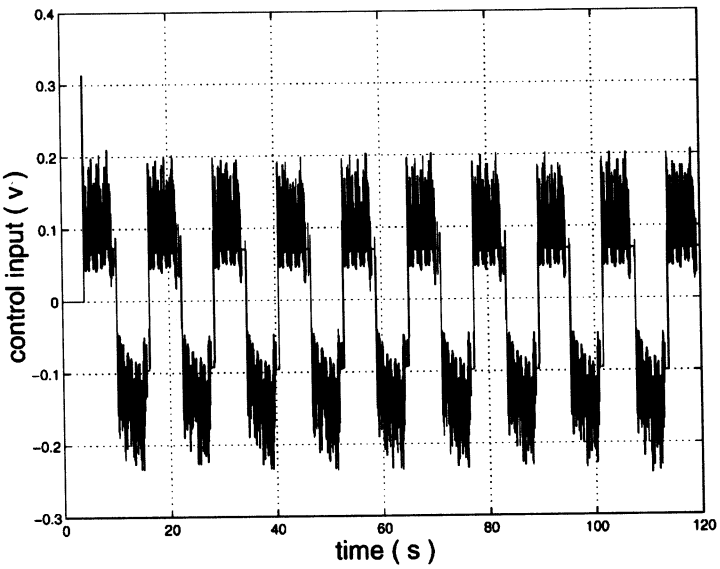

Figure 12. Control input under DRC.

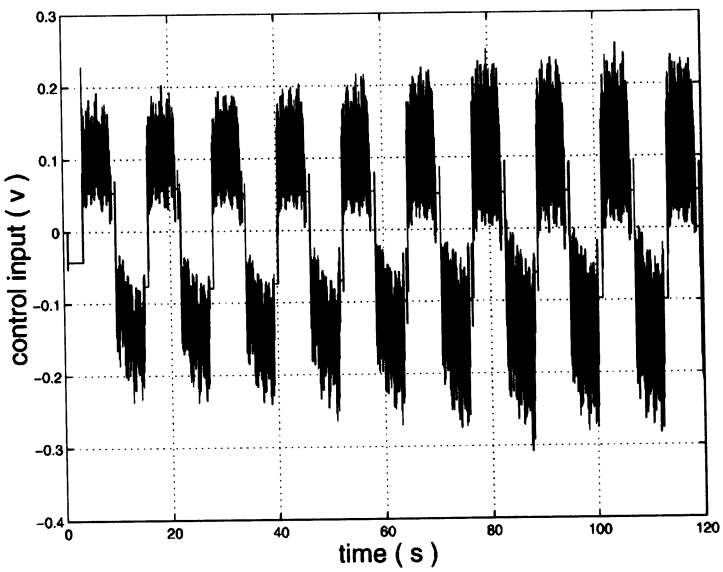

Figure 13. Control input under NNARC.

For completeness, the control inputs for both two controllers are given in Figures 12 and 13, which reveal that both controllers use almost the same level of control effort.

Overall, the proposed NNARC can have an excellent output tracking performance even with little knowledge of the system and parameters not converging to their true values.

5. CONCLUSION

In this paper, motivated by the precision motion control problem of linear motors, performance oriented NNARC control law has been constructed for a class of nth-order SISO uncertain non-linear systems. The proposed NNARC law takes full advantages of both neural networks and adaptive robust control (ARC) designs. The universal approximation capability of neural networks is utilized to construct multi-layer neural networks to approximate all unknown but repeatable non-linear 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 non-repeatable non-linearities 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 NNARC has the nice feature that if the unknown non-linear functions are in the functional ranges of the neural networks and the ideal weights fall within the prescribed range, asymptotic output tracking is also achieved—a performance that existing NN-based robust adaptive controllers cannot have. The proposed NNARC strategy is also applied to the precision motion control of a linear motor to deal with low speed control problem caused by the position dependent friction and electro-magnetic force ripples. Comparative experiments are carried out. Experimental results verify the high performance nature of the proposed NNARC strategy.

APPENDIX

A. Consider a positive definite function $V = \frac{1}{2}s^2$. In viewing (9), (26), and (48) as well as the fact that $\hat{\alpha}_{fi} \geqslant 0$, and $\hat{\alpha}_{bi} \geqslant 0$, the time derivative of V is

$$\begin{aligned} V = s\dot{s} &= -k\frac{b(\mathbf{x})}{b_l}s^2 - \frac{b(\mathbf{x})}{b_l}\hat{\boldsymbol{\alpha}}_f\mathbf{Y}_f|s| - \frac{b(\mathbf{x})}{b_l}\hat{\boldsymbol{\alpha}}_b\mathbf{Y}_b|u_a||s| \\ &\quad + s\{\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{\theta} + [f(\mathbf{x}) + a_r(t)] + b(\mathbf{x})u_a + \Delta + b(\mathbf{x})u_{s2}\} \\ &\leqslant -ks^2 + \varepsilon_s \end{aligned} \tag{A1}$$

which leads to (52).

B. Since $s(0) = 0$, from inequality (53), it is seen that

$$s^2(t) \leqslant \frac{\varepsilon_s}{k}[1 - \exp(-2kt)] \leqslant \frac{\varepsilon_s}{k} \tag{A2}$$

Hence, the inequality (53) is obtained [44].

C. Consider a positive definite function as follows

$$\begin{aligned} V = \tfrac{1}{2}[&s^2(t) + \tilde{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{\Gamma}_{\theta}^{-1}\tilde{\boldsymbol{\theta}} + \tilde{\mathbf{w}}_f^{\mathrm{T}}\boldsymbol{\Gamma}_{wf}^{-1}\tilde{\mathbf{w}}_f + \mathrm{Trace}\{\tilde{\mathbf{V}}_f\boldsymbol{\Gamma}_{vf}^{-1}\tilde{\mathbf{V}}_f^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_f^{\mathrm{T}}\boldsymbol{\Gamma}_{\alpha f}^{-1}\tilde{\boldsymbol{\alpha}}_f \\ &+ \tilde{\mathbf{w}}_b^{\mathrm{T}}\boldsymbol{\Gamma}_{wb}^{-1}\tilde{\mathbf{w}}_b + \mathrm{Trace}\{\tilde{\mathbf{V}}_b\boldsymbol{\Gamma}_{vb}^{-1}\tilde{\mathbf{V}}_b^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_b^{\mathrm{T}}\boldsymbol{\Gamma}_{\alpha_b}^{-1}\tilde{\boldsymbol{\alpha}}_b] \end{aligned} \tag{A3}$$

When $f(\mathbf{x}) = \mathbf{w}_f^{\mathrm{T}}\mathbf{g}_f$, $b(\mathbf{x}) = \mathbf{w}_b^{\mathrm{T}}\mathbf{g}_b$, and $\Delta = 0$, noting (26), the adaptation laws (27)–(33), the time derivative of V is derived as follows

$$
\begin{aligned}
V &= s\dot{s} + \tilde{\theta}^{\mathrm{T}}\Gamma_{\theta}^{-1}\dot{\hat{\theta}} + \tilde{\mathbf{w}}_f^{\mathrm{T}}\Gamma_{wf}^{-1}\dot{\hat{\mathbf{w}}}_f + \mathrm{Trace}\{\tilde{\mathbf{V}}_f\Gamma_{vf}^{-1}\dot{\hat{\mathbf{V}}}_f^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_f^{\mathrm{T}}\Gamma_{\alpha f}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_f \\
&\quad + \tilde{\mathbf{w}}_b^{\mathrm{T}}\Gamma_{wb}^{-1}\dot{\hat{\mathbf{w}}}_b + \mathrm{Trace}\{\tilde{\mathbf{V}}_b\Gamma_{vb}^{-1}\dot{\hat{\mathbf{V}}}_b^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_b^{\mathrm{T}}\Gamma_{\alpha_b}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_b \\
&= s\Big\{ - \phi^{\mathrm{T}}\tilde{\theta} + d_{fNN} + [- \tilde{\mathbf{w}}_f^{\mathrm{T}}(\hat{\mathbf{g}}_f - \hat{\mathbf{g}}_f'\hat{\mathbf{V}}_f\mathbf{x}_a) - \hat{\mathbf{w}}_f^{\mathrm{T}}\hat{\mathbf{g}}_f'\tilde{\mathbf{V}}_f\mathbf{x}_a] \\
&\quad + u_a d_{bNN} + [- \tilde{\mathbf{w}}_b^{\mathrm{T}}(\hat{\mathbf{g}}_b - \hat{\mathbf{g}}_b'\hat{\mathbf{V}}_b\mathbf{x}_a) - \hat{\mathbf{w}}_b^{\mathrm{T}}\hat{\mathbf{g}}_b'\tilde{\mathbf{V}}_b\mathbf{x}_a]u_a \\
&\quad - k\frac{b(\mathbf{x})}{b_l}s - \frac{b(\mathbf{x})}{b_l}\hat{\boldsymbol{\alpha}}_f^{\mathrm{T}}\mathbf{Y}_f\mathrm{sgn}(s) - \frac{b(\mathbf{x})}{b_l}\hat{\boldsymbol{\alpha}}_b^{\mathrm{T}}\mathbf{Y}_b|u_a|\mathrm{sgn}(s) + b(\mathbf{x})u_{s2}\Big\} + \tilde{\theta}^{\mathrm{T}}\Gamma_{\theta}^{-1}\dot{\hat{\theta}} \\
&\quad + \tilde{\mathbf{w}}_f^{\mathrm{T}}\Gamma_{wf}^{-1}\dot{\hat{\mathbf{w}}}_f + \mathrm{Trace}\{\tilde{\mathbf{V}}_f\Gamma_{vf}^{-1}\dot{\hat{\mathbf{V}}}_f^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_f^{\mathrm{T}}\Gamma_{\alpha f}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_f + \tilde{\mathbf{w}}_b^{\mathrm{T}}\Gamma_{wb}^{-1}\dot{\hat{\mathbf{w}}}_b \\
&\quad + \mathrm{Trace}\{\tilde{\mathbf{V}}_b\Gamma_{vb}^{-1}\dot{\hat{\mathbf{V}}}_b^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_b^{\mathrm{T}}\Gamma_{\alpha_b}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_b \\
&\leqslant - k\frac{b(\mathbf{x})}{b_l}s^2 + s\{d_{fNN} + [- \tilde{\mathbf{w}}_f^{\mathrm{T}}(\hat{\mathbf{g}}_f - \hat{\mathbf{g}}_f'\hat{\mathbf{V}}_f\mathbf{x}_a) - \hat{\mathbf{w}}_f^{\mathrm{T}}\hat{\mathbf{g}}_f'\tilde{\mathbf{V}}_f\mathbf{x}_a] + u_a d_{bNN} \\
&\quad - \tilde{\mathbf{w}}_b^{\mathrm{T}}(\hat{\mathbf{g}}_b - \hat{\mathbf{g}}_b'\hat{\mathbf{V}}_b\mathbf{x}_a)u_a - u_a\hat{\mathbf{w}}_b^{\mathrm{T}}\hat{\mathbf{g}}_b'\tilde{\mathbf{V}}_b\mathbf{x}_a\} - \frac{b(\mathbf{x})}{b_l}\hat{\boldsymbol{\alpha}}_f^{\mathrm{T}}\mathbf{Y}_f|s| - \frac{b(\mathbf{x})}{b_l}\hat{\boldsymbol{\alpha}}_b^{\mathrm{T}}\mathbf{Y}_b|u_a||s| \\
&\quad + \tilde{\mathbf{w}}_f^{\mathrm{T}}\Gamma_{wf}^{-1}\dot{\hat{\mathbf{w}}}_f + \mathrm{Trace}\{\tilde{\mathbf{V}}_f\Gamma_{vf}^{-1}\dot{\hat{\mathbf{V}}}_f^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_f^{\mathrm{T}}\Gamma_{\alpha f}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_f + \tilde{\mathbf{w}}_b^{\mathrm{T}}\Gamma_{wb}^{-1}\dot{\hat{\mathbf{w}}}_b \\
&\quad + \mathrm{Trace}\{\tilde{\mathbf{V}}_b\Gamma_{vb}^{-1}\dot{\hat{\mathbf{V}}}_b^{\mathrm{T}}\} + \tilde{\boldsymbol{\alpha}}_b^{\mathrm{T}}\Gamma_{\alpha_b}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_b \\
&\leqslant - ks^2 + \boldsymbol{\alpha}_f^{\mathrm{T}}\mathbf{Y}_f|s| + \boldsymbol{\alpha}_b^{\mathrm{T}}\mathbf{Y}_b|u_a||s| - \hat{\boldsymbol{\alpha}}_f^{\mathrm{T}}\mathbf{Y}_f|s| - \hat{\boldsymbol{\alpha}}_b^{\mathrm{T}}\mathbf{Y}_b|u_a||s| + \tilde{\boldsymbol{\alpha}}_f^{\mathrm{T}}\Gamma_{\alpha f}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_f + \tilde{\boldsymbol{\alpha}}_b^{\mathrm{T}}\Gamma_{\alpha_b}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_b \\
&\leqslant - ks^2
\end{aligned}
\tag{A4}
$$

in which the properties (41)–(47) have been used.

Since V is positive definite and $\dot{V}$ is negative semi-definite, V is bounded. Hence s and all the estimate errors are bounded. From Equation (26), it can be found that $\dot{s}$ is bounded. Subsequently, $ds^2/dt = 2s\dot{s}$ is also a bounded function. Hence, s^2 is a uniform continuous function. Furthermore, from inequality $\dot{V} \leqslant - ks^2$, it is obtained that $\int_0^t s(\tau)^2 d\tau \leqslant V(0)/k$. Hence $\int_0^t s(\tau)^2 d\tau$ is a non-decreasing upper-bounded function of time. Consequently, $\lim_{t\to\infty}\int_0^t s^2(\tau)\,d\tau$ exists and is finite. By Barbalat's lemma [45], $s^2 \to 0$ as $t \to \infty$, so does s. Therefore, the asymptotic tracking is achieved.

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

ACKNOWLEDGEMENTS

The authors would like to express their sincere thanks to Dr. Li Xu for his help in conducting the experiments. The authors are also grateful to the anonymous reviewers for their constructive suggestions and comments.

REFERENCES

1. Funahashi K-I. On the approximate realization of continuous mappings by neural networks. *Neural Networks* 1989; **2**:183–192.

2. Hornik K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* 1991; **4**:251–257.
3. Cybenko G. Approximation by superpositions of sigmoidal function. *Mathematics of Control, Signals and Systems* 1989; **2**:303–314.
4. Poggio T, Girosi F. Networks for approximation and learning. *Proceedings of the IEEE* 1990; **78**(9):1481–1497.
5. Park J, Sandberg IW. Universal approximation using radial-basis-function networks. *Neural Computation* 1991; **3**:246–257.
6. Kelly DG. Stability in contractive non-linear neural networks. *IEEE Transactions on Biomedical Engineering* 1990; **37**(3):231–242.
7. Liang XB, Yamaguchi T. On the analysis of global and absolute stability of nonlinear continuous neural networks. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences* 1997; **E80-A**(1):223–229.
8. Forti M, Manetti S, Marini M. Necessary and sufficient condition for absolut stability of neural networks. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications* 1994; **41**(7):491–494.
9. Matsuoka K. Stability conditions for nonlinear continuous neural networks with asymmetric connection weights. *Neural Networks* 1992; **5**:495–500.
10. Tanaka K. An approach to stability criteria of neural-network control systems. *IEEE Transactions on Neural Networks* 1996; **7**(3):629–642.
11. Sussmann HJ. Uniqueness of the weights for minimal feedforward nets with a given input–output map. *Neural Networks* 1992; **5**:589–593.
12. Hirsch MW. Convergent activation dynamics in continuous time networks. *Neural Networks* 1989; **2**:331–349.
13. Hunt KJ, Sbarbaro D, Zbikowski R, Gawthrop PJ. Neural networks for control systems—a survey. *Automatica* 1992; **28**(6):1083–1112.
14. Ge SS, Lee TH, Harris CJ. *Adaptive Neural Network Control of Robotic Manipulators*. World Scientific: London, 1998.
15. Lewis FL, Yesidirek A, Liu K. Neural net robot controller with guaranteed tracking performance. *IEEE Transcations on Neural Networks* 1995; **6**:703–715.
16. Polycarpou MM. Stable adptive neural control scheme for nonlinear systems. *IEEE Transcations on Automatic Control* 1996; **41**:447–451.
17. Zhang Y, Ioannou PA, Chien CC. Parameter convergence of a new class of adaptive controllers. *IEEE Transactions on Automatic Control* 1996; **41**(10):1489–1493.
18. Sanner RM, Slotine J-JE. Gaussian networks for direct adaptive control. *IEEE Transcations on Neural Networks* 1992; **3**(6):837–863.
19. Zhang T, Ge SS, Hang CC. Design and performance analysis of a direct adaptive controller for nonlinear systems. *Automatica* 1999; **35**:1809–1817.
20. Zhang T, Ge SS, Hang CC. Neural-based direct adaptive control for a class of general nonlinear systems. *International Journal of Systems Science* 1997; **28**(10):1011–1020.
21. Ge SS, Hang CC, Zhang T. Neural-based direct adaptive neural network control of nonlinear systems by state and output feedback. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics* 1999; **29**(6):818–828.
22. Yao B, Tomizuka M. Smooth robust adaptive sliding mode control of robot manipulators with guaranteed transient performance. *Transactions of ASME, Journal of Dynamic Systems, Measurement and Control* 1996; **118**(4):764–775. (Part of the paper also appeared in the *Proceedings of the 1994 American Control Conference*.)
23. Yao B. High performance adaptive robust control of nonlinear systems: a general framework and new schemes. *Proceedings of IEEE Conference on Decision and Control* 1997; 2489–2494.
24. Yao B, Tomizuka M. Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form. *Automatica* 1997; **33**(5):893–900. (Part of the paper appeared in *Procedings of the 1995 American Control Conference* 1995; 2500–2505).
25. Yao B, Tomizuka M. Adaptive robust control of MIMO nonlinear systems in semi-strict feedback forms. 1999. Submitted to *Automatica* (revised in 1999 and conditionally accepted) (Parts of the paper were presented in the *IEEE Conference on Decision and Control*, 1995; 2346–2351. and the *IFAC World Congress*, vol. F, 1996; 335–340.)

26. Gong JQ, Yao B. Adaptive robust control without knowing bounds of parameter variations. *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, AZ, USA, 7–10 December 1999; 3334–3339.
27. Alter DM, Tsao TC. Control of linear motors for machine tool feed drives: design and implementation of H_∞ optimal feedback control. *ASME Journal of Dynamic Systems, Measurement, and Control* 1996; **118**:649–656.
28. Alter DM, Tsao TC. Dynamic stiffness enhancement of direct linear motor feed drives for machining. *Proceedings of American Control Conference* 1994; 3303–3307.
29. Braembussche PV, Swevers J, Brussel HV, Vanherck P. Accurate tracking control of linear synchronous motor machine tool axes. *Mechatronics* 1996; **6**(5):507–521.
30. Komada S, Ishida M, Ohnishi K, Hori T. Disturbance observer-based motion control of direct drive motors. *IEEE Transactions on Energy Conversion* 1991; **6**(3):553–559.
31. Egami T, Tsuchiya T. Disturbance suppression control with preview action of linear DC brushless motor. *IEEE Transactions on Industrial Electronics* 1995; **42**(5):494–500.
32. Otten G, Vries T, Amerongen J, Rankers A, Gaal E. Linear motor motion control using a learning feedforward controller. *IEEE/ASME Transactions on Mechatronics* 1997; **2**(3):179–187.
33. Ohnishi K, Shibata M, Murakami T. Motion control for advanced mechatronics. *IEEE/ASME Transactions on Mechatronics* 1996; **1**(1):56–67.
34. Yao B, Al-Majed M, Tomizuka M. High performance robust motion control of machine tools: an adaptive robust control approach and comparative experiments. *IEEE/ASME Transactions on Mechatronics* 1997; **2**(2):63–76. (Part of the paper also appeared in *Proceedings of the 1997 American Control Conference.*)
35. Hanselman DC. *Brushless Permanent-Magnet Motor Design.* McGraw-Hill: New York, 1994.
36. de Wit CC, Olsson H, Astrom KJ, Lischinsky P. A new model for control of systems with friction. *IEEE Transactions on Automatic Control* 1995; **40**(3):419–425.
37. Brian A, Pierre D. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica* 1994; **30**(7):1083–1138.
38. Gong JQ, Yao B. Neural network-based adaptive robust control of a class of nonlinear systems in normal form. *Proceedings of the American Control Conference*, Chicago, IL, USA, 28–30 June 2000; 1491–1423.
39. Sastry S, Bodson M. *Adaptive Control: Stability, Convergence and Robustness.* Prentice-Hall: Englewood Cliffs, NJ, 1989.
40. Goodwin GC, Mayne DQ. A parameter estimation perspective of continuous time model reference adaptive control. *Automatica* 1989; **23**(1):57–70.
41. Fu L-C, Chang W-D, Yang J-H, Kuo T-S. Adaptive robust bank-to-turn missile autopilot design using neural networks. *Journal of Guidance, Control, and Dynamics* 1997; **20**(2):346–354.
42. Utkin VI. Variable structure systems with sliding modes. *IEEE Transactions on Automatic and Control* 1977; **22**(2):212–222.
43. Krstic M, Kanellakopoulos I, Kokotovic PV. *Non-linear and Adaptive Control Design.* New York: Wiley, 1995.
44. Slotine JJE, Li W. *Applied Nonlinear Control.* Prentice-Hall: Englewood Cliffs, NJ, 1991.
45. Khalil HK. *Nonlinear Systems* (2nd edn). Prentice-Hall: Englewood Cliffs, NJ, 1996.

