Asian Journal of Control, Vol. 3, No. 2, pp. 96-110, June 2001

NEURAL NETWORK ADAPTIVE ROBUST CONTROL OF SISO NONLINEAR SYSTEMS IN A NORMAL FORM

J.Q. Gong and Bin Yao

Asian Journal of Control, Vol. 3, No. 2, pp. 96-110, June 2001

NEURAL NETWORK ADAPTIVE ROBUST CONTROL OF SISO NONLINEAR SYSTEMS IN A NORMAL FORM

J.Q. Gong and Bin Yao

ABSTRACT

In this paper, performance oriented control laws are synthesized for a class of single-input-single-output (SISO) n-th order nonlinear systems in a normal form by integrating the neural networks (NNs) techniques and the adaptive robust control (ARC) design philosophy. All unknown but repeatable nonlinear functions in the system are approximated by the outputs of NNs to achieve a better model compensation for an improved performance. While all NN weights are tuned on-line, discontinuous projections with fictitious bounds are used in the tuning law to achieve a controlled learning. Robust control terms are then constructed to attenuate model uncertainties for a guaranteed output tracking transient performance and a guaranteed final tracking accuracy. Furthermore, if the unknown nonlinear functions are in the functional ranges of the NNs and the ideal NN weights fall within the fictitious bounds, asymptotic output tracking is achieved to retain the perfect learning capability of NNs. The precision motion control of a linear motor drive system is used as a case study to illustrate the proposed NNARC strategy.

KeyWords: Neural network, adaptive control, robust control, nonlinear systems.

I. INTRODUCTION

Nonlinearities in physical systems may appear in various forms. It is in general difficult to treat various nonlinearities under a unified framework. In some situations, 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. 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 [1-5]. It was shown in [2] that the standard multi-layer feedforward networks with only a single hidden layer and arbitrary bounded and nonconstant 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, sigmoid type of functions and radial basis function (RBF) are usually used in neural networks as activation functions. The approximation capabilities of networks with sigmoidal function being activation function were discussed in [3] while RBF networks were considered in [4,5]. 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.

In the research field of neural networks itself, the 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 of a variety of neural networks [6-10] and the evolution of the weights of neural networks [11, 12]. In [6,7], some criteria were proposed on how to choose the weights of neural networks to guarantee the stability of a class of nonlinear continuous neural networks. Existence and uniqueness of the equilibrium of the neural network were also discussed in [7]. Necessary and sufficient conditions were given in [8] for stability of a neural network of the Hopfield type with a symmetric weight matrix while the asymmetric case was studied in [9].

Manuscript received September 11, 2000; accepted December 4, 2000.

The authors are with School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA.

Part of the paper has been presented in the 2000 American Control Conference. The work is supported by the National Science Foundation under the CAREER grant CMS-9734345.

However, 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 in practice since weights can only be tuned in a relatively small region. As such, 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 large region. Consequently, the approximation range of a neural network becomes large, and a better approximation capability can be expected, which is helpful in the control of nonlinear systems when little is known about the nonlinearities 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 [14-17]. A survey of the application of neural networks to control field was given in [13], where modeling, identification, and control of nonlinear systems via neural networks were discussed. Two main issues have to be dealt with in the use of neural networks for nonlinear control design. Firstly, 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. Secondly, 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 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 [14], based on the assumption that the 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 [15], with the σ-modification type weight-tuning law, the adaptive neural network control schemes were proposed for nonlinear systems with uncertainties not satisfying matching conditions, where the input-hidden weights are also assumed to be known. The backstepping method was used in [16] to design a neural network controller to guarantee the semi-global stability of the closed system. RBF networks were used in [17] to adaptively compensate for the plant nonlinearities, 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. Furthermore, since the σ-modification type weight tuning method is used [14,15], an asymptotic output tracking cannot be achieved even when the unknown nonlinear function is in the functional range of the neural network. In other words, ideal perfect learning capability of neural networks is lost. In addition, the transient tracking performance is in general not known. The transient period may be long and large transient tracking errors may exhibit.

Recently, the adaptive robust control (ARC) approach has been proposed in [18-21] for nonlinear systems in the presence of both parametric uncertainties and non-repeatable 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 parametric uncertainties only, 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—have been considered in [18-21], 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 are integrated to synthesize performance oriented control laws for SISO nonlinear systems with matched model uncertainties in a normal form. The form allows unknown nonlinearities existing in both system model and input channel, and the unknown nonlinearities could include non-repeatable nonlinearities such as external disturbances as well. All unknown but repeatable nonlinear functions 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 [22] 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 nonlinearities 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 the existing NN based robust adaptive controllers [14-17] cannot achieve. In addition, if the unknown nonlinear 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 [14-17] cannot have. Furthermore, by choosing the prescribed range appropriately, the controller may have a built-in anti-integration windup mechanism to alleviate the effect of control saturation [22].

II. PROBLEM FORMULATION

The system to be considered in this paper has the following form [17,22]

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

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 i-th time derivative of the output x, $\phi(\boldsymbol{x}, t) = [\phi_1(\boldsymbol{x}, t), \ldots, \phi_r(\boldsymbol{x}, t)]^T$ is the vector of known basis functions, $\theta = [\theta_1, \ldots, \theta_r]^T$ is the constant unknown parameter vector, $\phi^T(\boldsymbol{x}, t)\theta$ denotes the structured nonlinearity [22], $f(\boldsymbol{x})$ 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 *non-repeatable* 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 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

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

where $\delta_f(\boldsymbol{x})$ is a known non-negative shape function, d_f is an unknown positive constant, $\boldsymbol{x}_a = [\boldsymbol{x}^T, -1]^T$ 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 $\boldsymbol{v}_{f1} \in \mathcal{R}^{(n+1)\times 1}$, r_f is the number of neurons, 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.

Remark 1. According to theorems in [1,2], nonlinearity f can be approximated by the output of a multi-layer 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.,

$$\left| f(\boldsymbol{x}) - \boldsymbol{w}_f^T \boldsymbol{g}_f(\boldsymbol{V}_f \boldsymbol{x}_a) \right| \le \eta_f, \ \forall \boldsymbol{x} \in \mathcal{A}_f \tag{3}$$

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.

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.,

$$\rho_{l,\theta_i} \le \theta_i \le \rho_{u,\theta_i}, \ i = 1, 2, \ldots, r \tag{4}$$

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

Remark 2. As a matter of fact, 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. However, since $\phi^T(\boldsymbol{x}, t)$ may explicitly depend on time t, the structured nonlinearity $\phi^T(\boldsymbol{x}, t)\theta$ is separated from the unstructured nonlinearity $f(\boldsymbol{x})$, which depends on the system state only.

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 3. The input gain $b(\boldsymbol{x})$ is nonzero with known sign. Without loss of generality, assume

$$b(\boldsymbol{x}) \ge b_l > 0, \ \forall \boldsymbol{x} \in \mathcal{R}^n \tag{5}$$

where b_l is a known positive constant.

Similar to Assumption 1, a neural network will be used to estimate the nonlinear input gain $b(\boldsymbol{x})$. Thus, the following assumption is made

Assumption 4. The nonlinear input gain $b(\boldsymbol{x})$ can be approximated by the output of a multi-layer neural network with

$$\left| b(\boldsymbol{x}) - \boldsymbol{w}_b^T \boldsymbol{g}_b(\boldsymbol{V}_b \boldsymbol{x}_a) \right| \le \delta_b(\boldsymbol{x}) d_b, \quad \forall \boldsymbol{x} \in \mathcal{R}^n \tag{6}$$

where $\delta_b(\boldsymbol{x})$ is a known non-negative shape funciton 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 $\boldsymbol{V}_b \in \mathcal{R}^{r_b \times (n+1)}$ is obtained. Similar to Remark 1, d_b can be made arbitrarily small with $\delta_b = 1$ 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 nonlinearity Δ is bounded by

$$\left| \Delta(\boldsymbol{x}, t) \right| \le \delta_\Delta(\boldsymbol{x}, t) d_\Delta(t) \tag{7}$$

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

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

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

Before leaving the section, the approximation properties of NNs and the discontinuous projection mapping are reviewed in the following two subsections to facilitate the controller designs.

2.1 Approximation properties of NN

By Taylor's expansion theory, the following theorem and lemma hold, which are proved in [23],

Theorem 2.1. [23] $\boldsymbol{w}_f^T \boldsymbol{g}_f(\boldsymbol{V}_f \boldsymbol{x}_a)$ can be approximated by its estimate $\hat{\boldsymbol{w}}_f^T \boldsymbol{g}_f(\hat{\boldsymbol{V}}_f \boldsymbol{x}_a)$ in the following form

$$\boldsymbol{w}_f^T \boldsymbol{g}_f(\boldsymbol{V}_f \boldsymbol{x}_a) = \hat{\boldsymbol{w}}_f^T \hat{\boldsymbol{g}}_f - \tilde{\boldsymbol{w}}_f^T(\hat{\boldsymbol{g}}_f - \hat{\boldsymbol{g}}_f' \hat{\boldsymbol{V}}_f \boldsymbol{x}_a) - \hat{\boldsymbol{w}}_f^T \hat{\boldsymbol{g}}_f' \tilde{\boldsymbol{V}}_f \boldsymbol{x}_a + d_{fNN} \tag{8}$$

where $\hat{\boldsymbol{g}}_f = \boldsymbol{g}_f(\hat{\boldsymbol{V}}_f \boldsymbol{x}_a)$, $\hat{\boldsymbol{g}}_f' = diag\{\hat{g}_{f1}', \ldots, \hat{g}_{fr_f}'\}$ with $\hat{g}_{fi}' = g_{fi}'(\hat{\boldsymbol{v}}_{fi}^T \boldsymbol{x}_a) = \left.\dfrac{d g_{fi}(z)}{dz}\right|_{z = \hat{\boldsymbol{v}}_{fi}^T \boldsymbol{x}_a}$, $i = 1, \ldots, r_f$, and the residual term $d_{fNN} = -\tilde{\boldsymbol{w}}_f^T \hat{\boldsymbol{g}}_f' \boldsymbol{V}_f \boldsymbol{x}_a + \boldsymbol{w}_f^T O(\tilde{\boldsymbol{V}}_f \boldsymbol{x}_a)$ with $O(\tilde{\boldsymbol{V}}_f \boldsymbol{x}_a)$ being the sum of the higher order terms.

Lemma 2.1. [23] The residual term d_{fNN} can be bounded by a linear-in-parameter function, i.e.,

$$\left| d_{fNN} \right| \le \boldsymbol{\alpha}_f^T \boldsymbol{Y}_f \tag{9}$$

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

$$\boldsymbol{Y}_f = \left[1, \|\boldsymbol{x}_a\|_2, \|\hat{\boldsymbol{w}}_f\|_2 \|\boldsymbol{x}_a\|_2, \|\hat{\boldsymbol{V}}_f\|_F \|\boldsymbol{x}_a\|_2 \right]^T \tag{10}$$

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

Proof. Please refer to [23].

Similar to Theorem 2.1 and Lemma 2.1, we have

Theorem 2.2. $\boldsymbol{w}_b^T \boldsymbol{g}_b(\boldsymbol{V}_b \boldsymbol{x}_a)$ can be approximated by its estimate $\hat{\boldsymbol{w}}_b^T \boldsymbol{g}_b(\hat{\boldsymbol{V}}_b \boldsymbol{x}_a)$ in the following way

$$\boldsymbol{w}_b^T \boldsymbol{g}_b(\boldsymbol{V}_b \boldsymbol{x}_a) = \hat{\boldsymbol{w}}_b^T \hat{\boldsymbol{g}}_b - \tilde{\boldsymbol{w}}_b^T(\hat{\boldsymbol{g}}_b - \hat{\boldsymbol{g}}_b' \hat{\boldsymbol{V}}_b \boldsymbol{x}_a) - \hat{\boldsymbol{w}}_b^T \hat{\boldsymbol{g}}_b' \tilde{\boldsymbol{V}}_b \boldsymbol{x}_a + d_{bNN} \tag{11}$$

where $\hat{\boldsymbol{g}}_b = \boldsymbol{g}(\hat{\boldsymbol{V}}_b \boldsymbol{x}_a)$, $\hat{\boldsymbol{g}}_b' = diag\ \{\hat{g}_{b1}', \ldots, \hat{g}_{br_b}'\}$ with $\hat{g}_{bi}' = g_{bi}'(\hat{\boldsymbol{v}}_{bi}^T \boldsymbol{x}_a) = \left.\dfrac{d g_{bi}(z)}{dz}\right|_{z = \hat{\boldsymbol{v}}_{bi}^T \boldsymbol{x}_a}$, $i = 1, \ldots, r_b$, and the residual term $d_{bNN} = -\tilde{\boldsymbol{w}}_b^T \hat{\boldsymbol{g}}_b' \boldsymbol{V}_b \boldsymbol{x}_a + \boldsymbol{w}_b O(\tilde{\boldsymbol{V}}_b \boldsymbol{x}_a)$ with $O(\tilde{\boldsymbol{V}}_b \boldsymbol{x}_a)$ being the sum of the higher order terms.

Lemma 2.2. The residual term d_{bNN} can be bounded by a linear-in-parameter function, i.e.,

$$\left| d_{bNN} \right| \le \boldsymbol{\alpha}_b^T \boldsymbol{Y}_b \tag{12}$$

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

$$\boldsymbol{Y}_b = \left[1, \|\boldsymbol{x}_a\|_2, \|\hat{\boldsymbol{w}}_b\|_2 \|\boldsymbol{x}_a\|_2, \|\hat{\boldsymbol{V}}_b\|_F \|\boldsymbol{x}_a\|_2 \right]^T \tag{13}$$

2.2 Discontinuous projection mapping

Since all the NN weights are constants, they are within certain bounded region. It is then reasonable to require that the estimates of the weights should be within the corresponding bounds. However, due to the fact that these bounds cannot be known *a prior*, certain fictitious bounds have to be used [22]. Discontinuous projection mapping [24,25] will be constructed based on these fictitious bounds. The following general notation is introduced

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

with its ij-th element defined as

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

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 $\rho_\star = \{\rho_{\star_{ij}}\}$.

With the above notations, the adaptation laws for the estimates of the unknown weights and bounds have the following form

$$\dot{\hat{\theta}} = \mathbf{Proj}_{\hat{\theta}}\{\Gamma_\theta \tau_\theta\} \tag{16}$$

$$\dot{\hat{\boldsymbol{w}}}_f = \mathbf{Proj}_{\hat{\boldsymbol{w}}_f}\{\Gamma_{wf}\tau_{wf}\} \tag{17}$$

$$\dot{\hat{\boldsymbol{V}}}_f = \mathbf{Proj}_{\hat{\boldsymbol{V}}_f}\{(\Gamma_{vf}\tau_{vf})^T\} \tag{18}$$

$$\dot{\hat{\boldsymbol{\alpha}}}_f = \mathbf{Proj}_{\hat{\boldsymbol{\alpha}}_f}\{\Gamma_{\alpha f}\tau_{\alpha f}\} \tag{19}$$

$$\dot{\hat{\boldsymbol{w}}}_b = \mathbf{Proj}_{\hat{\boldsymbol{w}}_b}\{\Gamma_{wb}\tau_{wb}\} \tag{20}$$

$$\dot{\hat{\boldsymbol{V}}}_b = \mathbf{Proj}_{\hat{\boldsymbol{V}}_b}\{(\Gamma_{vb}\tau_{vb})^T\} \tag{21}$$

$$\dot{\hat{\boldsymbol{\alpha}}}_b = \mathbf{Proj}_{\hat{\boldsymbol{\alpha}}_b}\{\Gamma_{\alpha b}\tau_{\alpha b}\} \tag{22}$$

where $\Gamma_\theta \in \mathcal{R}^{r\times r}$, $\Gamma_{wf} \in \mathcal{R}^{r_f\times r_f}$, $\Gamma_{vf} \in \mathcal{R}^{(n+1)\times(n+1)}$, $\Gamma_{\alpha f} \in \mathcal{R}^{4\times 4}$, Γ_{wb}, Γ_{vb}, and Γ_{α_b} are any diagonal positive definite (d.p.d.) adaptation rate matrices, and τ_θ, τ_{wf}, τ_{vf}, $\tau_{\alpha f}$, τ_{wb}, τ_{vb} and τ_{α_b} are adaptation functions to be specified later. Using similar arguments as in [18], it can be shown that the above projection-type updating laws have the following nice properties

P1. For any adaptation functions, the weight estimates are always within the corresponding known fictitious bounds used in defining the projection mappings, i.e.,

$$\text{P1 } \hat{\rho}_{l,\theta_i} \le \hat{\theta}_i \le \hat{\rho}_{u,\theta_i}, \quad i = 1, \ldots, r$$

$$\hat{\rho}_{l,w_{fi}} \le \hat{w}_{f_i} \le \hat{\rho}_{u,w_{f_i}}, \quad i = 1, \ldots, r_f$$

$$\hat{\rho}_{l,v_{f_{ij}}} \le \hat{v}_{f_{ij}} \le \hat{\rho}_{u,v_{f_{ij}}}, \quad i = 1, \ldots, r_f, j = 1, \ldots, n+1,$$

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

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

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

$$0 \le \hat{\alpha}_{bk} \le \hat{\rho}_{u,\alpha_{bk}}, \quad k = 1, \ldots, 4 \tag{23}$$

where the fictitious lower bounds for α_f and α_b are chosen to be zero since all their elements are positive as stated in Lemma 2.1 and 2.2. The fictitious lower and upper bounds $\hat{\rho}_{l,w_{bi}}$ and $\hat{\rho}_{u,w_{bi}}$ used in defining the projection mapping $\mathbf{Proj}_{\hat{\boldsymbol{w}}_b}$ for $\hat{w}_{bi}$, and $\hat{\rho}_{l,v_{bij}}$ and $\hat{\rho}_{u,v_{bij}}$ in $\mathbf{Proj}_{\hat{\boldsymbol{V}}_b}$ for $\hat{v}_{bij}$ should be chosen to guarantee that $\hat{\boldsymbol{w}}_b^T\hat{\boldsymbol{g}}_b \ne 0$, $\forall \hat{w}_{bi} \in [\hat{\rho}_{l,w_{bi}}, \hat{\rho}_{u,w_{bi}}]$, $\forall \hat{v}_{bij} \in [\hat{\rho}_{l,v_{bij}}, \hat{\rho}_{u,v_{bij}},]$; such a requirement may not be restrictive since the ideal value of $\boldsymbol{w}_b^T\boldsymbol{g}_b$, $\mathbf{b}(\boldsymbol{x})$, is positive as assumed in Assumption 3.

P2. In addition, if the true parameters θ, $\boldsymbol{w}_f$, $\boldsymbol{V}_f$, $\boldsymbol{\alpha}_f$, $\boldsymbol{w}_b$, $\boldsymbol{V}_b$, and $\boldsymbol{\alpha}_b$ are actually within the prescribed ranges, i.e., $\hat{\rho}_{l,\theta_i} \le \theta_i \le \hat{\rho}_{u,\theta_i}$, $\forall i = 1, \ldots, r$, $\hat{\rho}_{l,w_{fi}} \le w_{fi} \le \hat{\rho}_{u,w_{fi}}$, $\hat{\rho}_{l,v_{fij}} \le v_{fij} \le \hat{\rho}_{u,v_{fij}}$, $\forall i = 1, \ldots, r_f$, $\forall j = 1, \ldots, n+1$, $0 \le \alpha_{fk} \le \hat{\rho}_{u,\alpha_{fk}}$, $k = 1, \ldots, 4$, $\hat{\rho}_{l,w_{bi}} \le w_{bi} \le \hat{\rho}_{u,w_{bi}}$, $\hat{\rho}_{l,v_{bij}} \le v_{bij} \le \hat{\rho}_{u,v_{bij}}$, $\forall i = 1, \ldots, r_b$, $\forall j = 1, \ldots, n+1$, $0 \le \alpha_{bk} \le \hat{\rho}_{u,\alpha_{bk}}$, $k = 1, \ldots, 4$, then,

$$\text{P2 } \tilde{\theta}^T(\Gamma_\theta^{-1}\mathbf{Proj}_{\hat{\theta}}(\Gamma_\theta\bullet) - \bullet) \le 0, \qquad \forall\bullet$$

$$\tilde{\boldsymbol{w}}_f^T(\Gamma_{wf}^{-1}\mathbf{Proj}_{\hat{\boldsymbol{w}}_f}(\Gamma_{wf}\bullet) - \bullet) \le 0, \qquad \forall\bullet$$

$$\text{Trace}\{\tilde{\boldsymbol{V}}_f(\Gamma_{vf}^{-1}\mathbf{Proj}_{\hat{\boldsymbol{V}}_f}(\Gamma_{vf}\bullet) - \bullet)\} \le 0, \quad \forall\bullet$$

$$\tilde{\boldsymbol{\alpha}}_f^T(\Gamma_{\alpha f}^{-1}\mathbf{Proj}_{\hat{\boldsymbol{\alpha}}_f}(\Gamma_{\alpha f}\bullet) - \bullet) \le 0, \qquad \forall\bullet$$

$$\tilde{\boldsymbol{w}}_b^T(\Gamma_{wb}^{-1}\mathbf{Proj}_{\hat{\boldsymbol{w}}_b}(\Gamma_{wb}\bullet) - \bullet) \le 0, \qquad \forall\bullet$$

$$\text{Trace}\{\tilde{\boldsymbol{V}}_b(\Gamma_{vb}^{-1}\mathbf{Proj}_{\hat{\boldsymbol{V}}_b}(\Gamma_{vb}\bullet) - \bullet)\} \le 0, \quad \forall\bullet$$

$$\tilde{\boldsymbol{\alpha}}_b^T(\Gamma_{\alpha b}^{-1}\mathbf{Proj}_{\hat{\boldsymbol{\alpha}}_b}(\Gamma_{\alpha b}\bullet - \bullet) \le 0, \qquad \forall\bullet \tag{24}$$

III. NNARC WITH KNOWN INPUT-HIDDEN WEIGHTS

In order to explain the proposed NNARC clearly and compare it with other existing methodologies, we will solve the problem step by step. For this purpose, it is firstly assumed that the input-hidden weights of the network are known as done in [14,15,26,16] and the input gain is unity [14], i.e., $b \equiv 1$. In the subsequent sections, these assumptions will be removed, and on-line estimates of the input-hidden weights and NN approximation of $b(\boldsymbol{x})$ will be considered.

Since the control objective is to force $\boldsymbol{x}$ to track $\boldsymbol{x}_d$, a concise tracking error metric is defined by [17]:

$$s(t)=\left(\frac{d}{dt}+\lambda\right)^{n-1}\tilde{x}(t) \text{ with } \lambda>0 \tag{25}$$

where $\tilde{x}(t) = x(t) - x_d(t)$ is the output tracking error, λ is a positive constant. (25) can be rewritten as $s(t) = \boldsymbol{\lambda}^T\tilde{\boldsymbol{x}}(t)$ with the i-th element of vector $\boldsymbol{\lambda}$ 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. Thus, the perfect tracking can be asymptotically achieved by maintaining this condition [27].

Consider the control law

$$u = u_a + u_s \tag{26}$$

with

$$u_a=-a_r(t)-\phi^T(\boldsymbol{x},t)\hat{\theta}-\hat{\boldsymbol{w}}_f^T\boldsymbol{g}_f,$$

$$u_s = u_{s1} + u_{s2},\ u_{s1} = -ks \tag{27}$$

and adaptation laws

$$\dot{\hat{\theta}}=\mathbf{Proj}_{\hat{\theta}}(\Gamma_\theta s\phi(\boldsymbol{x},t)) \tag{28}$$

$$\dot{\hat{\boldsymbol{w}}}_f=\mathbf{Proj}_{\hat{\boldsymbol{w}}_f}(\Gamma_{wf}s\boldsymbol{g}_f) \tag{29}$$

where $k > 0$, and $a_r(t) = \boldsymbol{\lambda}_v^T\tilde{\boldsymbol{x}} - x_d^{(n)}$ with $\boldsymbol{\lambda}_v^T = [0, \lambda^{n-1}, \ldots, C_{n-1}^{i-2}\lambda^{n-i+1}, \ldots, (n-1)\lambda]$. In (27), the robust feedback term u_{s2} is synthesized to satisfy the following two conditions [22]

$$s\left(-\phi^T\tilde{\theta}+f(\boldsymbol{x})-\hat{\boldsymbol{w}}_f^T\boldsymbol{g}_f(V_f\boldsymbol{x}_a)+u_{s2}+\Delta\right)$$

$$\le\left|1+\frac{\left|\left(\|\rho_\theta\|_2-\|\hat{\rho}_\theta\|_2\right)\right|}{\|\hat{\rho}_\theta\|_2}\right|^2\epsilon_1+\left|1+\frac{\left|\left(\|\rho_{wf}\|_2-\|\hat{\rho}_{wf}\|_2\right)\right|}{\|\hat{\rho}_{wf}\|_2}\right|^2\epsilon_2$$

$$+\epsilon_3 d_f^2+\epsilon_4 d_\Delta^2(t) \tag{30}$$

$$su_{s2} \le 0 \tag{31}$$

in which ϵ_1, ϵ_2, ϵ_3, and ϵ_4 are four positive design constants, and $\|\bullet\|_2$ denotes the 2-norm of a vector $\bullet$. Control law (26) is referred to as NNARC-I in the following discussion.

Remark 3. The robust term u_{s2} in (30) may be chosen in the following way. Let

$$u_{s2}=-k_{s2}s,\ k_{s2}\ge\sum_{i=1}^{4}\frac{h_i^2}{4\epsilon_i} \tag{32}$$

where k_{s2} is a nonlinear gain and h_i's are sufficient smooth functions satisfying

$$h_1\ge\|\phi(\boldsymbol{x},t)\|\left(\|\hat{\theta}\|_2+\|\hat{\rho}_\theta\|_2\right), \tag{33}$$

$$h_2\ge\|\boldsymbol{g}_f\|\left(\|\hat{\boldsymbol{w}}_f\|_2+\|\hat{\rho}_{wf}\|_2\right), \tag{34}$$

$$h_3 \ge \delta_f(\boldsymbol{x}), \tag{35}$$

$$h_4 \ge \delta_\Delta(x, t) \tag{36}$$

and $\|\phi(\boldsymbol{x},t)\|=\sqrt{\Sigma_{i=1}^{i=r}|\phi_i(\boldsymbol{x},t)|^2}$, $\|\boldsymbol{g}_f\|=\sqrt{\Sigma_{i=1}^{i=r_f}|g_{fi}|_\infty^2}$. By doing so, (31) is satisfied, and

$$\text{Left hand side of (30)} \le -s\left(\phi^T(\boldsymbol{x},t)\tilde{\theta}+\frac{h_1^2}{4\epsilon_1}s\right)$$

$$-s\left(\tilde{\boldsymbol{w}}_f^T\boldsymbol{g}_f(V_f x_a)+\frac{h_2^2}{4\epsilon_2}s\right)$$

$$+s\left(f(\boldsymbol{x})-\boldsymbol{w}_f^T\boldsymbol{g}_f(V_f\boldsymbol{x}_a)-\frac{h_3^2}{4\epsilon_3}s\right)+s\left(\Delta-\frac{h_4^2}{4\epsilon_4}s\right) \tag{37}$$

It is thus easy to show that (30) is satisfied by using the completion of square, as in [22].

Remark 4. Usually, the activation functions g_{fi}'s are sigmoid functions; hence, $|g_{fi}|\le 1$ and $\|\boldsymbol{g}_f\|\le\sqrt{r_f}$. It is also clear that $\|\hat{\boldsymbol{w}}_f\|_2\le\|\hat{\rho}_{wf}\|_2$. Therefore, we can simply choose $h_2 = 2\sqrt{r_f}\|\hat{\rho}_{wf}\|_2$ in practice.

Substituting control law (26) into the system equation (1) with the assumption that $b(\boldsymbol{x}) = 1$ yields the following error equation

$$\dot{s}=-ks+u_{s2}-\phi^T\tilde{\theta}+f(\boldsymbol{x})-\hat{\boldsymbol{w}}_f^T\boldsymbol{g}_f(V_f\boldsymbol{x}_a)+\Delta \tag{38}$$

Theorem 3.1. With NNARC-I (26), the adaptation laws (28) and (29), 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

$$s^2(t) \le exp(-2kt)s^2(0) + \frac{\epsilon_s}{k}[1 - exp(-2kt)]$$
$$\le exp(-2kt)s^2(0) + \frac{\epsilon_s}{k} \quad (39)$$

where

$$\epsilon_s = \left|1 + \frac{\left|\left(\|\rho_\theta\|_2 - \|\hat{\rho}_\theta\|_2\right)\right|}{\|\hat{\rho}_\theta\|_2}\right|^2 \epsilon_1 + \left|1 + \frac{\left|\left(\|\rho_{wf}\|_2 - \|\hat{\rho}_{wf}\|_2\right)\right|}{\|\hat{\rho}_{wf}\|_2}\right|^2 \epsilon_2$$
$$+ \epsilon_3 d_f^2 + \epsilon_4 \|d_\Delta\|_\infty^2 \quad (40)$$

with $\|d_\Delta\|_\infty$ being L_∞ norm of $d_\Delta(t)$.

B. By the desired trajectory initialization [18], i.e., let $x_d(0) = x(0)$, we have, $s(0) = 0$. Consequently, the actual state tracking errors are asymptotically bounded by

$$\|\tilde{x}^{(i)}\|_\infty \le 2^i \lambda^{i-n+1} \sqrt{\frac{\epsilon_s}{k}} \quad (41)$$

C. If $f(x) = w_f^T g_f(V_f x_a)$ [3], i.e., the nonlinear function f is in the functional range of the neural network, then, in addition to the results in A and B, asymptotic output tracking is achieved provided that there is no external disturbance ($\Delta = 0$), and the actual parameter θ and the ideal weight w_f lie within their corresponding fictitious bounds (i.e., $\hat{\rho}_{l,\theta_i} \le \theta_i \le \hat{\rho}_{u,\theta_i}$, $\forall_i = 1, \ldots, r$, $\hat{\rho}_{l,w_{fi}} \le w_{fi} \le \hat{\rho}_{u,w_{fi}}$, $\forall i = 1, \ldots, r_f$).

Proof. See Appendix 1.

Remark 5. A and B of Theorem 3.1 show that the proposed NNARC-I achieves a guaranteed transient performance and final output tracking accuracy in general; it is seen that the exponentially converging rate $2k$ and the bounds of the final tracking error ($\|\tilde{x}\|_\infty \le \lambda^{-n+1}\sqrt{\epsilon_s/k}$) are related to the design parameters k, ϵ_1, ϵ_2, ϵ_3, and ϵ_4 in a known form, and can be adjusted freely by suitably choosing the design parameters. These results are thus much stronger than those in [14-16] where transient performance is not guaranteed.

Remark 6. C of Theorem 3.1 shows that the proposed NNARC-I is able to accomplish its learning goal (the assumptions in C of Theorem 3.1 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 [14-16] cannot attain this level of perfect learning capability.

Remark 7. From (26), it is clearly that the control law is composed of two parts, model compensation part u_a and robust control part u_s. For u_a, the key issue is how to adjust the estimate $\hat{w}_f$ to make $\hat{w}_f^T g_f$ as close to $f(x)$ as possible. For robust control part, the main focus is on the design of control gains to attenuate various model uncertainties effectively.

Remark 8. Same as in [22], the control saturation problem may be alleviated by choosing the fictitious bounds of θ and w_f appropriately.

IV. NNARC WITH UNKNOWN INPUT-HIDDEN WEIGHTS

Although the input-hidden weights may be obtained by the off-line training of neural networks, it will be more practical and beneficial if this assumption can be relaxed and input-hidden weights can be tuned on-line, which is the focus of this section. In the following, for simplicity, the sigmoid function will be used as activation functions. Other type of activation functions (e.g., RBF [17], bipolar sigmoid function [14]) can be worked out in the same way as long as the activation functions and their derivatives are bounded functions. It is still assumed that $b(x) \equiv 1$ in this section. This assumption will be removed in the next section.

The control law has the same structure as in (26), i.e.

$$u = u_a + u_s \quad (42)$$

with u_a and u_s given by

$$u_a = -a_r(t) - \phi^T(x, t)\hat{\theta} - \hat{w}_f^T \hat{g}_f,$$
$$u_s = u_{s1} + u_{s2},\ u_{s1} = -ks - \hat{\alpha}_f^T Y_f sgn(s) \quad (43)$$

Note that a sliding mode term is added in u_s by using the estimates of the residual bounds. In (43), the robust control term u_{s2} is synthesized to satisfy the following conditions

$$s\left(-\phi^T(x, t)\tilde{\theta} + f(x) - \hat{w}_f^T \hat{g}_f + u_{s2} + \Delta\right) \le \epsilon_s \quad (44)$$

$$su_{s2} \le 0 \quad (45)$$

where ϵ_s is given by (40). Control law (42) is referred to as NNARC-II in later discussion.

Since V_f is unknown, adaptation law (29) cannot be im-plemented. In viewing (8), it is changed to the following form

$$\dot{\hat{w}}_f = \mathbf{Proj}_{\hat{w}_f}(\Gamma_{wf} s(\hat{g}_f - \hat{g}_f' \hat{V}_f x_a)) \quad (46)$$

Furthermore, the following adaptation laws are employed to estimate the input-hidden weights V_f and the unknown parameter vector α_f.

$$\dot{\hat{V}}_f = \mathbf{Proj}_{\hat{V}_f}[(\Gamma_{vf}\tau_{vf})^T], \quad \tau_{vf} = \boldsymbol{x}_a s \hat{\boldsymbol{w}}_f^T \hat{\boldsymbol{g}}_f' \tag{47}$$

$$\dot{\hat{\boldsymbol{\alpha}}}_f = \mathbf{Proj}_{\hat{\alpha}_f}[(\Gamma_{\alpha f}\tau_{\alpha f})^T], \quad \tau_{\alpha f} = |s|\boldsymbol{Y}_f \tag{48}$$

Substituting control law (42) to the system equation (1) yields the following dynamic equation of $s(t)$

$$\dot{s} = -ks - \phi^T(\boldsymbol{x}, t)\tilde{\theta} + f(\boldsymbol{x}) - \hat{\boldsymbol{w}}_f^T \hat{\boldsymbol{g}}_f - \hat{\boldsymbol{\alpha}}_f^T \boldsymbol{Y}_f sgn(s) + \Delta + u_{s2} \tag{49}$$

Theorem 4.1. With NNARC-II (42), the adaptation laws (28), (46), (47), and (48), 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

$$s^2(t) \le exp(-2kt)s^2(0) + \frac{\epsilon_s}{k}[1 - exp(-2kt)]$$

$$\le exp(-2kt)s^2(0) + \frac{\epsilon_s}{k} \tag{50}$$

B. By desired trajectory initialization, i.e., let $\boldsymbol{x}_d(0) = \boldsymbol{x}(0)$, one has $s(0) = 0$. Consequently, the actual state tracking errors are asymptotically bounded by

$$\|\tilde{x}^{(i)}\|_\infty \le 2^i \lambda^{i-n+1} \sqrt{\frac{\epsilon_s}{k}} \tag{51}$$

C. If $f(\boldsymbol{x}) = \boldsymbol{w}_f^T \boldsymbol{g}_f(\boldsymbol{x})$ [3], i.e., the nonlinear function f is in the functional range of the neural network, then, in addition to the results in A and B, asymptotic output tracking is achieved provided that no disturbance exists, i.e., $\Delta = 0$, and all of the true parameters and weights lie within the corresponding fictitious bounds (i.e., $\hat{\rho}_{l,\theta_i} \le \theta_i \le \hat{\rho}_{u,\theta_i}$, $\forall i = 1, \ldots, r$, $\hat{\rho}_{l,w_{f_i}} \le w_{fi} \le \hat{\rho}_{u,w_{f_i}}$, $\hat{\rho}_{l,v_{f_{ij}}} \le v_{fij} \le \hat{\rho}_{u,v_{f_{ij}}}$, $\forall i = 1, \ldots, r_f$, $\forall j = 1, \ldots, n+1$, $0 \le \alpha_{fk} \le \hat{\rho}_{u,\alpha_{f_k}}$, $k = 1, \ldots, 4$).

D. When the discontinuous sign function $sgn(s)$ in (43) is dropped or replaced by any continuous function $\psi(s)$ satisfying $s\psi(s) \ge 0$, Results A and B still remain valid.

Proof. See Appendix 2.

Remark 9. In the above, although the input-hidden weights are unknown, by properly choosing control law and adaptation law, all the results in Section 3 remain valid.

Remark 10. In NNARC-II (42), the discontinuous term $\hat{\boldsymbol{\alpha}}_f^T \boldsymbol{Y}_f sgn(s)$ can be dropped from (43), and Results A and B in Theorem 4.1 still remain valid. The proof can be worked out in the same way as in the proofs of parts A and B of the Theorem in Appendix 2.

Remark 11. Although condition (44) is different from (30), u_{s2} in (44) can have the same form as (32) in Remark 3 with the same bounding functions h_1, h_2, h_3 and h_4 as in Remark 3. The reason is that $\|\hat{\boldsymbol{g}}_f\| = \|\boldsymbol{g}_f\|$ due to the use of sigmoid function. The details can be worked out in a similar way as in Remark 3 and are omitted.

V. NNARC FOR SYSTEMS WITH UNKNOWN INPUT GAIN

In the previous two sections, the case of unity input gain (or known input gain) has been considered. This section focuses on the case with unknown input gain. In viewing Assumption 6, the following control law is proposed

$$u = u_a + u_s \tag{52}$$

with

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

$$u_s = u_{s1} + u_{s2}, \quad u_{s1} = -\frac{1}{b_l}\left[ks + \left(\hat{\boldsymbol{\alpha}}_f^T \boldsymbol{Y}_f + \hat{\boldsymbol{\alpha}}_b^T \boldsymbol{Y}_b |u_a|\right) sgn(s)\right] \tag{53}$$

Using control law (52), the following dynamic equation can be obtained

$$\dot{s} = x^{(n)} + a_r(t) = \phi^T(\boldsymbol{x}, t)\theta + f(\boldsymbol{x}) + b(\boldsymbol{x})u + \Delta + a_r(t)$$

$$= \phi^T(\boldsymbol{x}, t)\theta + [f(\boldsymbol{x}) + a_r(t)] + b(\boldsymbol{x})u_a + b(\boldsymbol{x})u_s + \Delta$$

$$= \left[\phi^T(\boldsymbol{x}, t)\theta - \phi^T(\boldsymbol{x}, t)\hat{\theta}\right] + \left[f(\boldsymbol{x}) - \boldsymbol{w}_f^T \boldsymbol{g}_f\right] + \left[\boldsymbol{w}_f^T \boldsymbol{g}_f - \hat{\boldsymbol{w}}_f^T \hat{\boldsymbol{g}}_f\right]$$

$$+ \left[b(\boldsymbol{x}) - \boldsymbol{w}_b^T \boldsymbol{g}_b\right] u_a + \left[\boldsymbol{w}_b^T \boldsymbol{g}_b - \hat{\boldsymbol{w}}_b^T \hat{\boldsymbol{g}}_b\right] u_a$$

$$+ \Delta + b(\boldsymbol{x})u_{s1} + b(\boldsymbol{x})u_{s2}$$

$$= -\phi^T(\boldsymbol{x}, t)\tilde{\theta} + \left[f(\boldsymbol{x}) - \boldsymbol{w}_f^T \boldsymbol{g}_f + d_{fNN}\right]$$

$$+ \left[-\hat{\boldsymbol{w}}_f^T(\hat{\boldsymbol{g}}_f - \hat{\boldsymbol{g}}_f' \hat{\boldsymbol{V}}_f \boldsymbol{x}_a) - \hat{\boldsymbol{w}}_f^T \hat{\boldsymbol{g}}_f' \tilde{\boldsymbol{V}}_f \boldsymbol{x}_a\right]$$

$$+ \Big\{ [b(\boldsymbol{x}) - \boldsymbol{w}_b^T \boldsymbol{g}_b + d_{bNN}] + [-\hat{\boldsymbol{w}}_b^T(\hat{\boldsymbol{g}}_b - \hat{\boldsymbol{g}}_b' \hat{\boldsymbol{V}}_b \boldsymbol{x}_a) \cdot$$

$$- \hat{\boldsymbol{w}}_b^T \hat{\boldsymbol{g}}_b' \tilde{\boldsymbol{V}}_b \boldsymbol{x}_a] \Big\} u_a - k\frac{b(\boldsymbol{x})}{b_l}s - \frac{b(\boldsymbol{x})}{b_l}\hat{\boldsymbol{\alpha}}_f^T \boldsymbol{Y}_f sgn(s)$$

$$- \frac{b(\boldsymbol{x})}{b_l}\hat{\boldsymbol{\alpha}}_b^T \boldsymbol{Y}_b |u_a| sgn(s) + b(\boldsymbol{x})u_{s2} + \Delta \tag{54}$$

In viewing (54), the adaptation laws for $\hat{\boldsymbol{w}}_f$, $\hat{\boldsymbol{V}}_f$ and $\hat{\boldsymbol{\alpha}}_f$ are given by (46), (47), and (48), respectively. The adaptation laws for $\hat{\boldsymbol{w}}_b$, $\hat{\boldsymbol{V}}_b$, and $\hat{\boldsymbol{\alpha}}_b$ are as follows

$$\dot{\hat{\boldsymbol{w}}}_b = \mathbf{Proj}_{\hat{\boldsymbol{w}}_b}\left\{\Gamma_{wb}\left[u_a s(\hat{\boldsymbol{g}}_b - \hat{\boldsymbol{g}}_b' \hat{\boldsymbol{V}}_b \boldsymbol{x}_a)\right]\right\} \tag{55}$$

$$\dot{\hat{\boldsymbol{V}}}_b^T = \mathbf{Proj}_{\hat{\boldsymbol{V}}_b}\left\{\Gamma_{Vb}\left[u_a \boldsymbol{x}_a s \hat{\boldsymbol{w}}_b^T \hat{\boldsymbol{g}}_b'\right]\right\} \tag{56}$$

$$\dot{\hat{\boldsymbol{\alpha}}}_b = \mathbf{Proj}_{\hat{\boldsymbol{\alpha}}_b}\left\{\Gamma_{\alpha_b}\left[|s|\boldsymbol{Y}_b|u_a|\right]\right\} \tag{57}$$

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

$$s\left\{\phi^T\theta + [f(\boldsymbol{x}) + a_r(t)] + b(\boldsymbol{x})u_a + \Delta + b(\boldsymbol{x})u_{s2}\right\} \le \epsilon_{bs} \tag{58}$$

$$su_{s2} \le 0 \tag{59}$$

where

$$\epsilon_{bs} = \epsilon_s + \left|1 + \frac{\left|\left(\|\rho_{wb}\|_2 - \|\hat{\rho}_{wb}\|_2\right)\right|}{\|\hat{\rho}_{wb}\|_2}\right|^2 \epsilon_5 + d_b^2\epsilon_6 \tag{60}$$

in which ϵ_5 and ϵ_6 are positive design constants. The control law (52) is referred to as NNARC-III.

Theorem 5.1. With NNARC-III (52), the adaptation laws (28), (46), (47), (48), (55), (56), and (57), 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

$$s^2(t) \le exp(-2kt)s^2(0) + \frac{\epsilon_{bs}}{k}[1 - exp(-2kt)]$$
$$\le exp(-2kt)s^2(0) + \frac{\epsilon_{bs}}{k} \tag{61}$$

B. By desired trajectory initialization, i.e., letting $\boldsymbol{x}_d(0) = \boldsymbol{x}(0)$, one has $s(0) = 0$. Consequently, the actual tracking error is asymptotically bounded by

$$\left\|\tilde{x}^{(i)}\right\|_\infty \le 2^i\lambda^{i-n+1}\sqrt{\frac{\epsilon_{bs}}{k}} \tag{62}$$

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

D. When the discontinuous sign function $sgn(s)$ in (53) is replaced by the continuous saturation function $sat(\frac{s}{\psi})$ (ψ is the thickness of the boundary layer), Results A and B still remain valid.

Proof. See Appendix 3.

Remark 12. Similar to Remark 10, the discontinuous term $\frac{1}{b_l}[\hat{\boldsymbol{\alpha}}_f^T\boldsymbol{Y}_f + \hat{\boldsymbol{\alpha}}_b^T\boldsymbol{Y}_b|u_a|]sgn(s)$ can be dropped from (53) if only Results A and B in Theorem 5.1 are required. The proof is similar to parts A and B in Appendix 3.

Remark 13. In viewing the third and fourth line of (54), u_{s2} in (53) can be selected in the same form as (32) with $h_1 \ge \frac{1}{\sqrt{b_l}}\|\phi(\boldsymbol{x}, t)\|(\|\hat{\theta}\|_2 + \|\hat{\rho}_\theta\|_2)$, $h_2 \ge \frac{1}{\sqrt{b_l}}\|\hat{\boldsymbol{g}}_f\|(\|\hat{\boldsymbol{w}}_f\| + \|\hat{\rho}_{wf}\|)$, $h_3 \ge \frac{1}{\sqrt{b_l}}\delta_f$, $h_4 \ge \frac{1}{\sqrt{b_l}}\delta(x, t)$, and $h_5 \ge \frac{1}{\sqrt{b_l}}\|\hat{\boldsymbol{g}}_b\|(\|\hat{\boldsymbol{w}}_b\| + \|\hat{\rho}_{wb}\|)|u_a|$, and $h_6 \ge \frac{1}{\sqrt{b_l}}|u_a|\delta_b$.

VI. EXPERIMENTAL CASE STUDY

The proposed NNARC has been recently applied to the precision motion control of an expoxy core linear motor having the following dynamic model in [28]:

$$M\ddot{x} = -B\dot{x} - F_a(x, \dot{x}) + u + F_b(x, \dot{x}, t) \tag{63}$$

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, 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 the electr-magnetic force ripples, and F_d is the normalized external disturbance force (e.g. cutting force in machining), which may be state dependent and time-varying. In practice, M, B are unknown constants, and F_d is bounded. It is seen that the system (63) can be put in the form of (1) with $\boldsymbol{x} = [x, x^{(1)}]^T$, $\phi = \dot{x}$, $\theta = -\frac{B}{M}$, $f(\boldsymbol{x}) = -\frac{F_a}{M}$, $b(\boldsymbol{x}) = \frac{1}{M}$, $\Delta = \frac{F_d}{M}$.

---

* Normalized with respect to the unit input voltage.

The details of the experimental set-up and off-line identification can be found in [28] and only the sketch of the controller design and typical experimental results are shown below to illustrate the proposed NNARC strategy. The off-line identification in [28] indicates that $\boldsymbol{f}(\boldsymbol{x}) = -\frac{1}{M}F_a$ depends on both position x and velocity $\dot{x}$ and is hard to model. Hence, a three-layer neural network is employed to approximate this unknown nonlinearity with an input vector of $[x, \dot{x}, -1]^T$. Three types of neurons are used in the hidden layer to take into account the particular properties of $f(\boldsymbol{x})$:

1. Type-I: Five hidden neurons with a conventional sigmoid function $g(\bullet) = \frac{1-exp(-\bullet)}{1+exp(-\bullet)}$ 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 $\boldsymbol{g}(\bullet)$ 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 ($sin(\frac{2\pi}{0.03}x)$, $sin(\frac{6\pi}{0.03}x)$, and $sin(\frac{10\pi}{0.03}x)$) being activation functions are used to compensate for the electro-magnetic 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 other 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.

Since the input-hidden weights are fixed in the last two types of hidden neurons, the corresponding adaptation laws can be simplified to

$$\dot{\hat{\boldsymbol{w}}}_f = \mathbf{Proj}_{\hat{\boldsymbol{w}}_f}(\Gamma_{wf} s \boldsymbol{g}_f) \tag{64}$$

where $\boldsymbol{g}_f = [\sin(\frac{2\pi}{0.03}x), \sin(\frac{6\pi}{0.03}x), \sin(\frac{10\pi}{0.03}x)]^T$ for Type-II neurons, and $g_f = 1$ for Type-III neuron.

From (63), it is clear that the input gain $\frac{1}{M}$ 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 (56) and (57) are not needed, and adaptation law (55) becomes

$$\dot{\hat{w}}_b = \dot{\hat{b}} = \mathbf{Proj}_{\hat{b}}\{\Gamma_b u_a s\} \tag{65}$$

Since no hidden layer exists in NN for $b(\boldsymbol{x})$, the robust control term $\hat{\boldsymbol{\alpha}}_b^T \boldsymbol{Y}_b |u_a| sgn(s)$ in (53) is not needed. Furthermore, as stated in Result D of Theorem 5.1, $\frac{2}{\pi}arctan(\psi_a s)$ can be used to replace the sign function in (53) to avoid control input chattering. The resulting NNARC is

$$u_{nnarc} = -\frac{a_r(t) + \hat{\theta}\dot{x} + \hat{\boldsymbol{w}}_f^T \boldsymbol{g}_f}{\hat{b}} - \frac{1}{b_l}\left[ks + \frac{2}{\pi}\hat{\boldsymbol{\alpha}}_f^T \boldsymbol{Y}_f arctan(\psi_a s)\right] + u_{s2} \tag{66}$$

where u_{s2} assumes the form (32) with $k_{s2} = max(3, \Sigma_{i=1}^{i=5}\frac{h_i^2}{4\epsilon_i})$, and $\psi_a = 90$ is used in the experiments.

6.1 Parameters in control laws and adaptation laws

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 $0.025 \le M \le 0.1$, $b_l = 10$ results. The actual value of b is 37.037 with an initial estimate of $\hat{b}(0) = 20$. The adaptation rate is 5000. The fictitious lower and upper bounds of θ are assumed the values of -11 and 0 with an initial estimate of $\hat{\theta}(0) = -8.111$ and adaptation rate of 4×10^4. $\lambda = 200$ is used for the sliding plane and $k = 20$ is used as the constant gain. The final tracking accuracy indices are $\epsilon_i = 5000$, $i = 1, \ldots, 6$.

6.2 Experimental results

It is known that the main nonlinearity in the model (63) 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 nonlinearity, a low speed point-to-point desired trajectory with a maximum speed of 0.02 *m/s* is used.

The experimental results are given in Figs. 1-3. As shown in Fig. 1, the output tracks the desired trajectory very well. In fact, the tracking error during the entire run is within ±5.6 μm, and for most of the run, the tracking error stays within ±2 μm, almost within the encoder resolution of ±1 μm of the experimental systems. The appearance of small spikes when the direction of velocity changes is caused by the Coulomb friction, which is near discontinuous around those time instances. 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 not surprise to see these spikes. Even so, it can be seen that the magnitude of these spikes keep decreasing until reaching to a small value of ±2.5 μm. All these results indicate the high performance nature of the proposed NNARC.

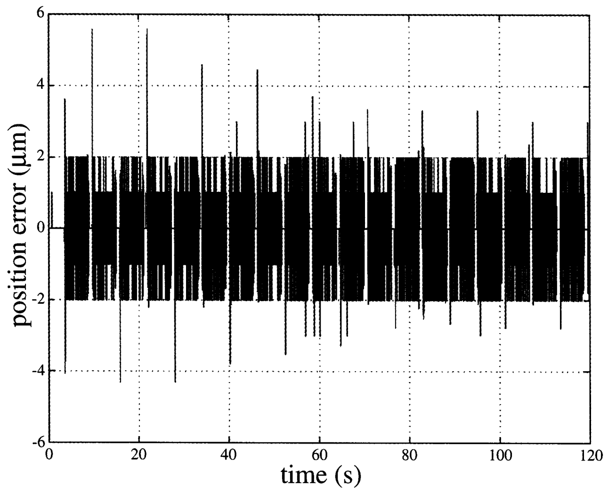


Fig. 1. Position error under NNARC.

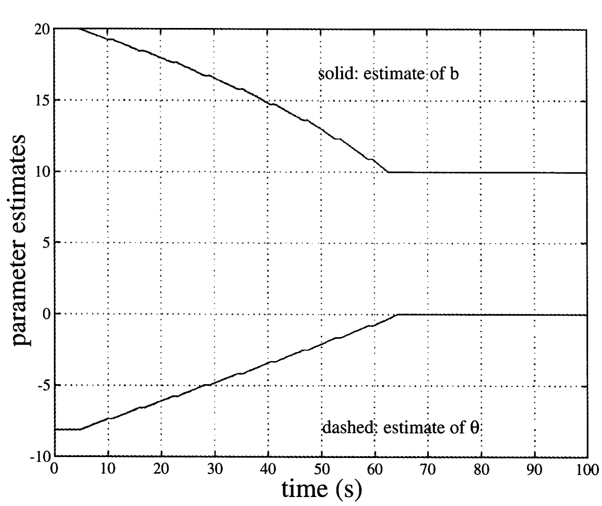


Fig. 2. Estimate of θ and b under NNARC.

The time histories of the estimate $\hat{\theta}$ and $\hat{b}$ are given in Fig. 2. Since the desired trajectory is not rich enough and the persistent excitation condition cannot be satisfied, the estimates "drift 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.

The output of the neural network (a stable filter with transfer function $\frac{50}{s+50}$ is used to get rid of the noise effects) is plotted versus the position as shown in Fig. 3. It can be seen that the filtered NN output tends to have a shape like the actual one in [28].

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.

VII. CONCLUSION

Performance oriented NNARC control laws have been constructed for a class of n-th order SISO nonlinear systems with matched uncertainties. 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 nonlinear functions to achieve a better model compensation for an improved performance. All NN weights are tuned on-line with no prior training

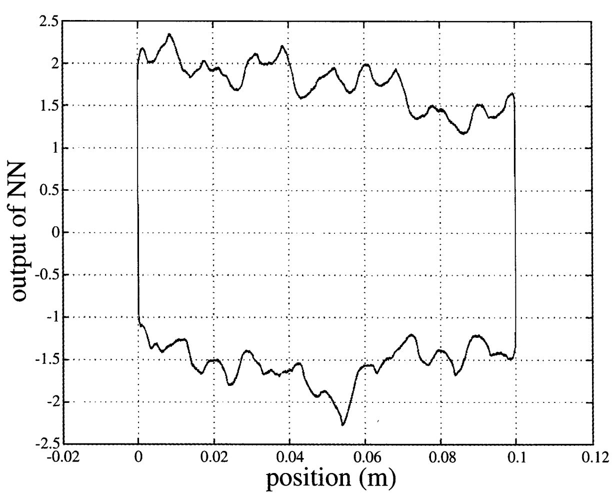


Fig. 3. Output of NN under NNARC (with filter $\frac{50}{s+50}$) from $t = 79.9996$ *sec* to $t = 92.1780$ sec.

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 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 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, asymptotic output tracking is also achieved-a performance that existing NN based robust adaptive controllers cannot have.

APPENDIX 1

A. Consider the following Lyapunov function candidate

$$V=\frac{1}{2}s^2(t) \tag{67}$$

Noting (38) and the condition (30), the time derivative of V is as follows

$$\dot{V}=s\dot{s}\le -ks^2+\epsilon_s=-2kV+\epsilon_s \tag{68}$$

which leads to inequality (39).

B. Since $s(0) = 0$, from inequality (39), we have

$$s^2(t)\le\frac{\epsilon_s}{k}[1-exp(-2kt)]\le\frac{\epsilon_s}{k} \tag{69}$$

Hence, the inequality (41) is obtained [29].

C. Consider the following Lyapunov function candidate

$$V=\frac{1}{2}\left[s^2(t)+\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}+\tilde{w}_f^T\Gamma_{wf}^{-1}\tilde{w}_f\right] \tag{70}$$

When $f(x) = w_f^T g_f(V_f x_a)$ and $\Delta = 0$,

$$\dot{V}=s\dot{s}+\tilde{\theta}^T\Gamma_\theta^{-1}\dot{\hat{\theta}}+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f$$

$$=-ks^2+su_{s2}-s\phi^T\tilde{\theta}+\tilde{\theta}^T\Gamma_\theta^{-1}\mathbf{Proj}_{\hat{\theta}}(\Gamma_\theta s\phi)-\tilde{w}_f^T s g_f(V_f x_a)$$

$$+\tilde{w}_f^T\Gamma_{wf}^{-1}\mathbf{Proj}_{\hat{w}_f}(\Gamma_{wf} s g_f)\le -ks^2+su_{s2}\le -ks^2 \tag{71}$$

in which the first two inequalities in properties (24) and condition (31) are utilized. By applying Barbalat's lemma [18], it can be proved that asymptotic output tracking is achieved. ■

APPENDIX 2

A. Consider the following Lyapunov function candidate

$$V=\frac{1}{2}s^2 \tag{72}$$

Noting the dynamic equation (49) and conditions (44), as well as the fact that $\hat{\alpha}_{fi}\ge 0$, we have

$$\dot{V}=s\dot{s}=-ks^2+s\left(-\phi^T\tilde{\theta}+f(x)-\hat{w}_f^T\hat{g}_f-\hat{\alpha}_f^T Y_f sgn(s)+\Delta+u_{s2}\right)$$

$$=-ks^2-\hat{\alpha}_f^T Y_f|s|+s\left[-\phi^T\tilde{\theta}+f(x)-\hat{w}_f^T\hat{g}_f+\Delta+u_{s2}\right]$$

$$\le -ks^2+\epsilon_s \tag{73}$$

which leads to the inequality (50).

B. Using the same method as in the proof B of Theorem 3.1, inequality (51) can be resulted [29].

C. Consider Lyapunov function candidate as follows

$$V=\frac{1}{2}\Big[s^2(t)+\tilde{\theta}^T\Gamma_\theta^{-1}\tilde{\theta}+\tilde{w}_f^T\Gamma_{wf}^{-1}\tilde{w}_f+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\tilde{V}_f^T\}$$

$$+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\tilde{\alpha}_f\Big] \tag{74}$$

When $f(x) = w_f^T g_f(V_f x_a)$ and $\Delta = 0$, using adaptation laws (28), (46), (47), (48) and condition (45), we have

$$\dot{V}=s\dot{s}+\tilde{\theta}^T\Gamma_\theta^{-1}\dot{\hat{\theta}}+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\dot{\hat{V}}_f^T\}+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\dot{\hat{\alpha}}_f$$

$$=-ks^2+s\left[-\phi^T\tilde{\theta}+f(x)-\hat{w}_f^T\hat{g}_f-\hat{\alpha}_f^T Y_f sgn(s)+u_{s2}\right]$$

$$+\tilde{\theta}^T\Gamma_\theta^{-1}\dot{\hat{\theta}}+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\dot{\hat{V}}_f^T\}+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\dot{\hat{\alpha}}_f$$

$$=-ks^2+su_{s2}-s\phi^T\tilde{\theta}+\tilde{\theta}^T\Gamma_\theta^{-1}\dot{\hat{\theta}}+s\left[w_f^T g_f-\hat{w}_f^T\hat{g}_f\right]$$

$$-|s|\hat{\alpha}_f^T Y_f+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\dot{\hat{V}}_f^T\}+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\dot{\hat{\alpha}}_f$$

$$=-ks^2+su_{s2}-s\phi^T\tilde{\theta}+\tilde{\theta}^T\Gamma_\theta^{-1}\dot{\hat{\theta}}-s\left[\tilde{w}_f^T(\hat{g}_f-\hat{g}_f'\hat{V}_f x_a)\right]$$

$$-s\left[\hat{w}_f^T\hat{g}_f'\tilde{V}_f x_a-d_{fNN}\right]-|s|\hat{\alpha}_f^T Y_f+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f$$

$$-s\left[\hat{w}_f^T\hat{g}_f'\tilde{V}_f x_a-d_{fNN}\right]-|s|\hat{\alpha}_f^T Y_f+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f$$

$$+\text{Trace}\{\tilde{V}^T\Gamma_{vf}^{-1}\dot{\hat{V}}_f^T\}+\tilde{\alpha}_f^T\Gamma_{\alpha}^{-1}\dot{\hat{\alpha}}_f$$

$$=-ks^2+su_{s2}-s\phi^T\tilde{\theta}+\tilde{\theta}^T\Gamma_{\theta}^{-1}\mathbf{Proj}_{\hat{\theta}}(\Gamma_{\theta}s\phi(x,t))$$

$$-s\left[\tilde{w}_f^T(\hat{g}_f-\hat{g}_f'\hat{V}_f x_a)\right]-s\left(\hat{w}_f^T\hat{g}_f'\tilde{V}_f x_a\right)$$

$$+sd_{fNN}-|s|\tilde{\alpha}_f^T Y_f+\tilde{w}_f^T\Gamma_{wf}^{-1}\mathbf{Proj}_{\hat{w}_f}[\Gamma_{wf}s(\hat{g}_f-\hat{g}_f'\hat{V}_f x_a)]$$

$$+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\mathbf{Proj}_{\hat{V}_f}(\Gamma_{vf}x_a s\hat{w}_f^T\hat{g}_f')\}$$

$$+\tilde{\alpha}_f^T\Gamma_{\alpha f}\mathbf{Proj}_{\hat{\alpha}_f}(\Gamma_{\alpha f}|s|Y_f)$$

$$\le -ks^2+|s|\alpha_f^T Y_f-|s|\hat{\alpha}_f^T Y_f+\tilde{\alpha}_f^T\Gamma_{\alpha f}\mathbf{Proj}_{\hat{\alpha}_f}(\Gamma_{\alpha f}|s|Y_f)$$

$$=-ks^2-|s|\tilde{\alpha}_f^T Y_f+\tilde{\alpha}_f^T\Gamma_{\alpha f}\mathbf{Proj}_{\hat{\alpha}_f}(\Gamma_{\alpha f}|s|Y_f)$$

$$\le -ks^2\le 0 \qquad (75)$$

in which the fourth equality uses Theorem 2.1, the first inequality uses Lemma 2.1, condition (45), and the first three inequalities in (24), and the second inequality uses the fourth inequality in (24). Using Barbalat's lemma, it can be proved that asymptotic output tracking is achieved.

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

APPENDIX 3

A. Consider a Lyapunov function candidate $V=\frac{1}{2}s^2$. In viewing (5) and (58) as well as the fact that $\hat{\alpha}_{fi}\ge 0$, $\hat{\alpha}_{bi}\ge 0$, and $b(x)\ge b_l$, the time derivative of V is

$$\dot{V}=s\dot{s}=-k\frac{b(x)}{b_l}s^2-\frac{b(x)}{b_l}\hat{\alpha}_f Y_f|s|-\frac{b(x)}{b_l}\hat{\alpha}_b Y_b|u_a||s|$$

$$+s\left\{\phi^T\theta+[f(x)+a_r(t)]+b(x)u_a+\Delta+b(x)u_{s2}\right\}$$

$$\le -ks^2+\epsilon_{bs} \qquad (76)$$

which leads to (61).

B. Using the same method in the proof B of Theorem 3.1, inequality (62) can be resulted [29].

C. Consider a Lyapunov function candidate as follows

$$V=\frac{1}{2}\Big[s^2(t)+\tilde{\theta}^T\Gamma_{\theta}^{-1}\tilde{\theta}+\tilde{w}_f^T\Gamma_{wf}^{-1}\tilde{w}_f+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\tilde{V}_f^T\}$$

$$+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\tilde{\alpha}_f+\tilde{w}_b^T\Gamma_{wb}^{-1}\tilde{w}_b+\text{Trace}\{\tilde{V}_b\Gamma_{vb}^{-1}\tilde{V}_b^T\}+\tilde{\alpha}_b^T\Gamma_{\alpha_b}^{-1}\tilde{\alpha}_b\Big] \qquad (77)$$

When $f(x)=w_f^T g_f$, $b(x)=w_b^T g_b$, and $\Delta=0$, noting (54), the adaptation laws (28), (46), (47), (48), (55), (56), and (57), the time derivative of V is derived as follows

$$\dot{V}=s\dot{s}+\tilde{\theta}^T\Gamma_{\theta}^{-1}\dot{\hat{\theta}}+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\dot{\hat{V}}_f^T\}+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\dot{\hat{\alpha}}_f$$

$$+\tilde{w}_b^T\Gamma_{wb}^{-1}\dot{\hat{w}}_b+\text{Trace}\{\tilde{V}_b\Gamma_{vf}^{-1}\dot{\hat{V}}_b^T\}+\tilde{\alpha}_b^T\Gamma_{\alpha_b}^{-1}\dot{\hat{\alpha}}_b$$

$$=s\Big\{-\phi^T\tilde{\theta}+\left[f(x)-w_f^T g_f+d_{fNN}\right]+\Big[-\tilde{w}_f^T(\hat{g}_f-\hat{g}_f'\hat{V}_f x_a)$$

$$-\hat{w}_f^T\hat{g}_f'\tilde{V}_f x_a\Big]+[b(x)-w_b^T g_b+d_{bNN}]u_a$$

$$+[-\tilde{w}_b^T(\hat{g}_b-\hat{g}_b'\hat{V}_b x_a)-\hat{w}_b^T\hat{g}_b'\tilde{V}_b x_a]u_a$$

$$-k\frac{b(x)}{b_l}s-\frac{b(x)}{b_l}\hat{\alpha}_f^T Y_f sgn(s)-\frac{b(x)}{b_l}\hat{\alpha}_b^T Y_b|u_a|sgn(s)$$

$$+b(x)u_{s2}+\Delta\Big\}+\tilde{\theta}^T\Gamma_{\theta}^{-1}\dot{\hat{\theta}}+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f$$

$$+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\dot{\hat{V}}_f^T\}+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\dot{\hat{\alpha}}_f+\tilde{w}_b^T\Gamma_{wb}^{-1}\dot{\hat{w}}_{fb}$$

$$+\text{Trace}\{\tilde{V}_b\Gamma_{vb}^{-1}\dot{\hat{V}}_b^T\}+\tilde{\alpha}_b^T\Gamma_{\alpha_b}^{-1}\dot{\hat{\alpha}}_b$$

$$\le -k\frac{b(x)}{b_l}s^2+s\Big\{d_{fNN}+\left[-\tilde{w}_f^T(\hat{g}_f-\hat{g}_f'\hat{V}_f x_a)-\hat{w}_f^T\hat{g}_f'\tilde{V}_f x_a\right]$$

$$+u_a d_{bNN}-\tilde{w}_b^T(\hat{g}_b-\hat{g}_b'\hat{V}_b x_a)u_a-u_a\hat{w}_b^T\hat{g}_b'\tilde{V}_b x_a\Big\}$$

$$-\frac{b(x)}{b_l}\hat{\alpha}_f^T Y_f|s|-\frac{b(x)}{b_l}\hat{\alpha}_b^T Y_b|u_a||s|$$

$$+\tilde{w}_f^T\Gamma_{wf}^{-1}\dot{\hat{w}}_f+\text{Trace}\{\tilde{V}_f\Gamma_{vf}^{-1}\dot{\hat{V}}_f^T\}+\tilde{\alpha}_f^T\Gamma_{\alpha f}^{-1}\dot{\hat{\alpha}}_f+\tilde{w}_b^T\Gamma_{wb}^{-1}\dot{\hat{w}}_b$$

$$+\text{Trace}\{\tilde{V}_b\Gamma_{vb}^{-1}\dot{\hat{V}}_b^T\}+\tilde{\alpha}_b^T\Gamma_{\alpha_b}^{-1}\dot{\hat{\alpha}}_b$$

$$\le -ks^2+\alpha_f^T Y_f|s|+s\left[-\tilde{w}_f^T(\hat{g}_f-\hat{g}_f'\hat{V}_f x_a)-\hat{w}_f^T\hat{g}_f'\tilde{V}_f x_a\right]$$

$$+\alpha_b^T Y_b|u_a||s|-su_a\tilde{w}_b^T(\hat{g}_b-\hat{g}_b'\hat{V}_b x_a)$$

$$
\begin{aligned}
&-su_a\hat{\boldsymbol{w}}_b^T\hat{\boldsymbol{g}}_b'\tilde{\boldsymbol{V}}_b\boldsymbol{x}_a-\hat{\boldsymbol{\alpha}}_f^T\boldsymbol{Y}_f|s|-\hat{\boldsymbol{\alpha}}_b^T\boldsymbol{Y}_b|u_a||s| \\
&+\tilde{\boldsymbol{w}}_f^T\Gamma_{wf}^{-1}\dot{\hat{\boldsymbol{w}}}_f+\mathrm{Trace}\{\tilde{\boldsymbol{V}}_f\Gamma_{vf}^{-1}\dot{\hat{\boldsymbol{V}}}_f^T\}+\tilde{\boldsymbol{\alpha}}_f^T\Gamma_{\alpha f}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_f \\
&+\tilde{\boldsymbol{w}}_b^T\Gamma_{wb}^{-1}\dot{\hat{\boldsymbol{w}}}_b+\mathrm{Trace}\{\tilde{\boldsymbol{V}}_b\Gamma_{vb}^{-1}\dot{\hat{\boldsymbol{V}}}_b^T\}+\tilde{\boldsymbol{\alpha}}_b^T\Gamma_{\alpha_b}^{-1}\dot{\hat{\boldsymbol{\alpha}}}_b \\
&\le -ks^2 \qquad (78)
\end{aligned}
$$

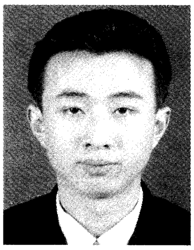
in which similar techniques as in the derivation of (75) have been used. Using the similar reasoning as in the proof of Theorem 4.1, the asymptotic output tracking is obtained.

D. Using the similar arguments as in the proof of Theorem 4.1, the result can be obtained. ■

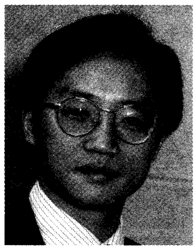
REFERENCES

1. Funahashi, K.-I., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, Vol. 2, pp. 183-192 (1989).
2. Hornik, K., "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, Vol, 4, pp. 251-257 (1991).
3. Cybenko, G., "Approximation by Superpositions of Sigmoidal Function," *Math. Contr. Signals Syst.*, Vol. 2, pp. 303-314 (1989).
4. Poggio, T. and G. Girosi, "Networks for Approximation and Learning," *Proc. IEEE*, Vol. 78, No. 9, pp. 1481-1497 (1990).
5. Park, J. and I.W. Sandberg, "Universal Approximation Using Radial-basis-function Networks," *Neural Comput.*, Vol. 3, pp. 246-257 (1991).
6. Kelly, D.G., "Stability in Contractive Nonlinear Neural Networks," *IEEE Trans. Biomed. Eng.*, Vol. 37, No. 3, pp. 231-242 (1990).
7. Liang, X.B. and T. Yamaguchi, "On the Analysis of Global and Absolute Stability of Nonlinear Continuous Neural Networks," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, Vol. E80-A, No. 1, pp. 223-229 (1997).
8. Forti, M., S, Manetti and M. Marini, "Necessary and Sufficient Condition for absolut stability of Neural Networks," *IEEE Trans. Circuits Syst.-I: Fundam. Theory Appl.*, Vol. 41, No. 7, pp. 491-494 (1994).
9. Matsuoka, K., "Stability Conditions for Nonlinear Continuous Neural Networks with Asymmetric Connection Weights," *Neural Networks*, Vol. 5, pp. 495-500 (1992).
10. Tanaka, K., "An Approach to Stability Criteria of Neural-Network Control Systems," *IEEE Trans. Neural Networks*, Vol. 7, No. 3, pp. 629-642 (1996).
11. Sussmann, H.J., "Uniqueness of the Weights for Minimal Feedforward Nets with a Given Input-output Map," *Neural Networks*, Vol. 5, pp. 589-593 (1992).
12. Hirsch, M.W., "Convergent Activation Dynamics in Continuous Time Networks," *Neural Networks*, Vol. 2, pp. 331-349 (1989).
13. Hunt, K.J., D. Sbarbaro, R. Zbikowski and P.J. Gawthrop, "Neural Networks for Control Systems-a Survey," *Automatica*, Vol. 28, No. 6, pp. 1083-1112 (1992).
14. Lewis, F.L, A. Yesidirek and K. Liu, "Neural Net Robot Controller with Guaranteed Tracking Performance," *IEEE Trans. Neural Networks*, Vol. 6, pp. 703-715 (1995).
15. Polycarpou, M.M., "Stable Adptive Neural Control Scheme for Nonlinear Systems," *IEEE Trans. Automat. Contr.*, Vol. 41, pp. 447-451 (1996).
16. Zhang, Y., P.A. Ioannou and C.C. Chien, "Parameter Convergence of a New Class of Adaptive Controllers," *IEEE Trans. Automat. Contr.*, Vol. 41, No. 10, pp. 1489-1493 (1996).
17. Sanner, R.M. and J.-J. E. Slotine, "Gaussian Networks for Direct Adaptive Control," *IEEE Trans. Neural Networks*, Vol. 3, No. 6, pp. 837-863 (1992).
18. Yao, B. and M. Tomizuka, "Smooth Robust Adaptive Sliding Mode Control of Robot Manipulators with Guaranteed Transient Performance," in *Proc. Amer. Contr. Conf.*, pp. 1176-1180 (1994) (Full paper appeared in *ASME J. Dyn. Syst., Meas. Contr.*, Vol. 118, No. 4, pp. 764-775 (1996)).
19. Yao, B., "High Performance Adaptive Robust Control of Nonlinear Systems: A General Framework and New Schemes," in *Proc. IEEE Conf. Decis. Contr.*, pp. 2489-2494 (1997).
20. Yao, B. and M. Tomizuka, "Adaptive Robust Control of SISO Nonlinear Systems in a Semi-strict Feedback Form," *Automatica*, Vol. 33, No. 5, pp. 893-900, (1997). (Part of the paper appeared in *Proc. Amer. Contr. Conf.*, pp. 2500-2505 (1995)).
21. Yao, B. and M. Tomizuka, "Adaptive Robust Control of MIMO Nonlinear Systems in Semi-strict Feedback Forms," *Automatica* (2000). (Accepted) (Parts of the paper were presented in *IEEE Conf. Decis. Contr.*, pp. 2346-2351 (1995), and *IFAC World Congr.*, Vol. F, pp. 335-340 (1996)).
22. Gong, J.Q. and B. Yao, "Adaptive Robust Control Without Knowing Bounds of Parameter Variations," in *Proc. 38th IEEE Conf. Decis.Contr.*, Phoenix, Arizona, USA, pp. 3334-3339 (1999).
23. Fu, L.-C., W.-D. Chang, J.-H. Yang and T.-S. Kuo, "Adaptive Robust Bank-to-turn Missile Sutopilot Design Using Neural Networks," *J. Guid., Contr., Dyn.*, Vol. 20, No. 2, pp. 346-354 (1997).
24. Sastry S. and M. Bodson, *Adaptive Control; Stability, Covergence and Robustness*, Prentice-Hall, Inc Englewood Cliffs, NJ, USA (1989).
25. Goodwin, G.C. and D.Q. Mayne, "A Parameter Estimation Perspective of Continuous Time Model Reference Adaptive Control," *Automatica*, Vol. 23, No. 1, pp. 57-70 (1989).

26. Polycarpou M.M. and P.A. Ioannou, "Neural Networks as On-line Approximators of Nonlinear Systems," in *Proc. 31st IEEE Conf. Decis. Contr.*, pp. 7-12 (1992).
27. Utkin, V.I., "Variable Structure Systems with Sliding Modes," *IEEE Trans. Automat. Contr.*, Vol. 22, No. 2, pp. 212-222 (1997).
28. Gong, J.Q. and B. Yao, "Neural Network Adaptive Robust Control with Application to Precision Motion Control of Linear Motors," *Int. J. Adaptive Contr. Signal Process.*, (2000) (Accepted).
29. Slotine J.J.E. and W. Li, *Applied Nonlinear Control.*, (Prentice-Hall) Englewood Cliffs, New Jersey (1991).



Jian Qin Gong received his B.E. and M.E. degrees from Xi'an Jiaotong University, China, in 1993 and 1996, in Electrical Engineering and Systems Engineering, respectively. He received another M.E. degree from National University of Singapore, Singapore in 1999 in Electrical Engineering. He is currently a Ph.D. student in the School of Mechanical Engineering at Purdue University, West Lafayette. His research interests include intelligent vehicle highway system (IVHS), control of rigid/flexible smart material robots, control of linear motor drive system, control of hard disk drive (HDD), adaptive control, robust control, and neural network control.



Bin Yao received the Ph.D. degree in Mechanical Engineering from the University of California at Berkeley in February 1996. Since 1996, he has been an Assistant Professor in the school of Mechanical Engineering at Purdue University. Dr. Yao's research focuses on the development of a general framework for the design of high performance adaptive robust control algorithms. Integration of the approach with actual sensor properties, physical properties, and control-oriented modeling has been carried out for different applications such as the control of electro-hydraulic systems, linear motors, machine tools, and robot manipulators. He experimentally verified the approach on the precision control of various electro-mechanical/hydraulic systems. Dr. Yao is the recipient of the 1997 Caterpillar Engineering Young Faculty Development Fund for his work on the electro-hydraulic control, and the 1998 National Science Foundation CAREER award for his work on the engineering synthesis of high performance adaptive robust controllers for mechanical systems and manufacturing processes. His research interests include design and coordinated control of intelligent high performance electro-mechanical/electro-hydraulic systems, optimal adaptive and robust control, nonlinear observer design and neural networks for virtual sensing, modeling, fault detection, diagnostics and adaptive fault-tolerant control, robotics, and data fusion.