Adaptive Control for Uncertain Nonlinear Systems Based on Multiple Neural Networks

Choon-Young Lee and Ju-Jang Lee, *Senior Member, IEEE*

***Abstract*—A new adaptive multiple neural network controller (AMNNC) with a supervisory controller for a class of uncertain nonlinear dynamic systems was developed in this paper. The AMNNC is a kind of adaptive feedback linearizing controller where nonlinearity terms are approximated with multiple neural networks. The weighted sum of the multiple neural networks was used to approximate system nonlinearity for the given task. Each neural network represents the system dynamics for each task. For a job where some tasks are repeated but information on the load is not defined and unknown or varying, the proposed controller is effective because of its capability to memorize control skill for each task with each neural network. For a new task, most similar existing control skills may be used as a starting point of adaptation. With the help of a supervisory controller, the resulting closed-loop system is globally stable in the sense that all signals involved are uniformly bounded. Simulation results on a cartpole system for the changing mass of the pole were illustrated to show the effectiveness of the proposed control scheme for the comparison with the conventional adaptive neural network controller (ANNC).**

***Index Terms*—Adaptive control, learning control, neural networks, supervisory control.**

I. Introduction

AN INTELLIGENT control system may have the ability to operate in multiple environments by understanding the current condition of operation and achieving the various tasks appropriately. The ability to adapt to any unknown operating condition is an important component to intelligent systems. Adaptive control is a promising technique to obtain a model of the plant and its environment from experimental data and to design a controller. Adaptive control for a feedback linearizable nonlinear system has attracted much interest among control system designers for over several decades. If we have exact knowledge of the system, we can transform a nonlinear adaptive control problem into a linear control problem by using a feedback linearization technique [1].

However, in many cases, the plant to be controlled is too complex to find the exact system dynamics, and the operating conditions in dynamic environments may be unexpected. Therefore, recently, an adaptive control technique has been combined with function approximators such as neural networks, fuzzy inference systems, and series expansion. These types of controllers take the capability of learning unknown nonlinear functions by universal approximation theorem [2], [3] and massive parallel computation. Based on the fact that universal approximators are capable of uniformly approximating a given nonlinear function over a compact set to any degree of accuracy, a globally stable adaptive controller has been developed with an adaptation algorithm [5]–[15]. Each study has its own advantages and disadvantages. An adaptive control scheme was used with a neural network to obtain a stable controller [6], [8], [10], [13]. An observer-based controller was derived for nonlinear systems with state estimation without measuring all the states [9], [11]. Off-line training of the robot manipulator was used in [7]. Although all these methodologies showed good performance in controlling uncertain dynamic systems, there are unavoidable large transient errors at the time of task variation. For example, if the robot manipulator has to perform task 1, task 2, task 1, task 2, repeatedly in this order, an adaptive controller will adapt itself to the newly changed task repeatedly, so the control skill having been acquired previously will be forgotten. Although task 1 is encountered for the second time, the adaptive controller recognizes the given task as a new task because the controller has been already adapted to task 2. In this case, if the dynamic parameters and control skills are stored, the former can be utilized to recognize the given task if it occurs at a later time, and the latter can be used to cope with the repeated task immediately.

Concepts for adaptive control with multiple models are not new in control theory. Multiple filter models were studied to improve the accuracy of the state estimation and control performance [16], [17]. In these studies, a combination of the control determined by different models was used, and no stability results were reported. Switching in the context of adaptive control, in which multiple models are used to determine to which controller one should switch, has been proposed [18]–[21]. The stability of linear switching and tuning systems was analyzed. The neuro-controller based on a set of fixed neural networks (NNs), that is, a bank of NNs trained off-line or on-line, with each NN representing some part of inverse dynamics, was also proposed [22], [23]. In these studies, multiple neural networks were used to improve the overall performance in identification and total computational requirement. They can be thought of as a single neural network in the sense that all the weights of the networks were adapted without any priority for each task.

In this paper, multiple neural networks were adopted to store different dynamics of the system for different tasks. An adaptive feedback linearization controller with these multiple neural networks was proposed with a supervisory controller. Global stability is guaranteed by the supervisory controller even for the new environment [12]. The proposed approach is different from

Manuscript received July 18, 2002; revised February 19, 2003. This work was supported in part by the Human-friendly Welfare Robot System Engineering Research Center (HWRS-ERC), Korea Science and Engineering Foundation. This paper was recommended by Associate Editor S. Phoha.

The authors are with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, 305-701, Republic of Korea (e-mail: cylee@odyssey.kaist.ac.kr).

Digital Object Identifier 10.1109/TSMCB.2003.811520

Narendras work in that our controller uses blending of multiple models, whereas only one model can be active at any given time in [20]. Weights for blending multiple models are calculated based on the predictive modeling error. Our algorithm uses one controller incorporating multiple models, whereas multiple inverse controllers are used in Kawatos work [26].

This paper is organized as follows. The problem formulation is described in Section II. The adaptive controller based on multiple neural networks with a supervisory controller is constructed in Section III. A simulation to demonstrate the performances of the proposed method is provided in Section IV. Finally, conclusions with further study issues are given in Section V.

II. Problem Formulation

Consider the nth order nonlinear dynamic system of the form

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= x_3\\ &\cdots\\ \dot{x}_n &= f(x_1, x_2, \ldots, x_n) + g(x_1, x_2, \ldots, x_n)u + d\\ y &= x_1\end{aligned} \tag{1}$$

or, equivalently, the form

$$\begin{aligned}x^{(n)} &= f\left(x, \dot{x}, \ldots, x^{(n-1)}\right)\\ &\quad + g\left(x, \dot{x}, \ldots, x^{(n-1)}\right)u + d\\ y &= x\end{aligned} \tag{2}$$

where f and g are unknown but bounded functions, $u \in \mathbb{R}$ and $y \in \mathbb{R}$ are the control input and output of the system respectively, and d is an external bounded disturbance.

If we represent (2) in the state space, we obtain

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}(f(\mathbf{x}) + g(\mathbf{x})u + d)\\ y &= \mathbf{C}^T\mathbf{x}\end{aligned} \tag{3}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 0 & 1 & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \cdots & 0 & 1\\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0\\ 0\\ \vdots\\ 0\\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} c1\\ 0\\ \vdots\\ 0\\ 0 \end{bmatrix} \tag{4}$$

and $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$ is a state vector where all x_i are assumed to be available for measurement. In order for (2) to be controllable, it is required that $g(\mathbf{x}) \neq 0$ for $\mathbf{x}$ in a certain controllability region $U_c \subset \mathbb{R}^n$.

Assumption 1: Without loss of generality, we can assume that $0 < g^L(\mathbf{x}) \leq g(\mathbf{x}) \leq g^U(\mathbf{x}) < \infty$ for $\mathbf{x} \in U_c$, and $|f(\mathbf{x})| \leq f^U(\mathbf{x}) < \infty$ for $\mathbf{x} \in U_c$. Furthermore, external disturbance is bounded, i.e., $|d| \leq d_m$, where d_m is the upper bound of noise d.

Let the reference signal vector $\mathbf{y}_d$ and the tracking error vector $\mathbf{e}$ be defined as

$$\mathbf{y}_d = \left[y_d, \dot{y}_d, \ldots, y_d^{(n-1)}\right]^T \in \mathbb{R}^n \tag{5}$$

$$\mathbf{e} = \left[e, \dot{e}, \ldots, e^{(n-1)}\right]^T \in \mathbb{R}^n \tag{6}$$

where $e = y_d - y = y_d - x_1 \in \mathbb{R}$. Then, the control objective is to generate an appropriate control signal such that the system output y follows a given bounded reference signal y_d under the stability constraint that all signals involved in the system must be bounded.

III. Adaptive Controller Based on Multiple Neural Networks

If the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ are known and there is no external disturbance d, then we can choose the following controller cancelling the nonlinearity of the system:

$$u^* = \frac{1}{g(\mathbf{x})}\left[-f(\mathbf{x}) + y_d^{(n)} + \lambda^T\mathbf{e}\right]. \tag{7}$$

If we apply the feedback linearizing controller (7) into the system (3), we obtain

$$\dot{\mathbf{e}} = \mathbf{\Lambda}\mathbf{e} \tag{8}$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 0 & 1 & 0 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \cdots & 0 & 1\\ -\lambda_1 & -\lambda_2 & -\lambda_3 & -\lambda_4 & \cdots & -\lambda_{n-1} & -\lambda_n \end{bmatrix}. \tag{9}$$

In particular, let $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T \in \mathbb{R}^n$ be chosen such that $\mathbf{\Lambda}$ is Hurwitz; then, $\lim_{t\to\infty} e(t) = 0$.

Since $f(\mathbf{x})$ and $g(\mathbf{x})$ are unknown, we replace the function $f(\mathbf{x})$ and $g(\mathbf{x})$ in (7) by the estimated function $\hat{f}(\mathbf{x})$ and $\hat{g}(\mathbf{x})$.

The radial basis function (RBF) network has been shown to have universal approximation ability to approximate any smooth function on a compact set S, simply connected set of $\mathbb{R}^m$. Let $f(\cdot)\colon S \to \mathbb{R}^m$ be a smooth function and $\{\phi(\mathbf{x})\}$ be a basis set, there exists a weight matrix $\mathbf{W}$ such that

$$f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{W} + \epsilon \tag{10}$$

with the estimation error bounded by $\|\epsilon\| < \epsilon_N$ for a given constant ϵ_N.

In light of the universal approximation capability of the RBF network, $f(\mathbf{x})$ and $g(\mathbf{x})$ may be identified using weighted sums of k multiple RBF networks with sufficiently high number of hidden-layer units such that

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i \phi_1(\mathbf{x})^T \mathbf{W}_{1i} + \epsilon_{1i}(\mathbf{x}) \tag{11}$$

$$\hat{g}(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i \phi_2(\mathbf{x})^T \mathbf{W}_{2i} + \epsilon_{2i}(\mathbf{x}) \tag{12}$$

where α_i is the significance parameter of the ith RBF network satisfying $0 \le \alpha_i \le 1$, and $\sum_{i=1}^{k} \alpha_i = 1$, $\mathbf{W}_{1i} \in \mathbb{R}^{n_1}$, and $\mathbf{W}_{2i} \in \mathbb{R}^{n_2}$ are unknown weights respectively, which are assumed to be bounded by

$$\|\mathbf{W}_{1i}\| \le W_{1B}, \qquad \|\mathbf{W}_{2i}\| \le W_{2B} \tag{13}$$

with W_{1B} and W_{2B} some known positive constants; n_1 and n_2 are the number of hidden-layer units of the two RBF networks, respectively; the approximation errors are bounded by $|\epsilon_{1i}| \le \epsilon_{1N}$ and $|\epsilon_{2i}| \le \epsilon_{2N}$, with ϵ_{1N} and ϵ_{2N} two positive constants; and $\phi_1(\mathbf{x})$ and $\phi_2(\mathbf{x})$ are properly chosen radial basis functions for the hidden-layer units of the two networks, with $\mathbf{x}$ as the input pattern of the input layers to the networks. The basis functions can be chosen as the Gaussian functions defined as [6]

$$\phi_{1i}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_{1i}\|^2}{\sigma_{1i}^2}\right), \qquad i = 1, 2, \ldots, n_1 \tag{14}$$

$$\phi_{2i}(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_{2i}\|^2}{\sigma_{2i}^2}\right), \qquad i = 1, 2, \ldots, n_2 \tag{15}$$

where $\mathbf{c}_{1i}$ and $\mathbf{c}_{2i}$ are centers of the basis functions, and σ_{1i}^2 and σ_{2i}^2 are widths, which are all chosen *a priori* and kept fixed throughout the experiment for simplicity.

If we select a Lyapunov function candidate as

$$V_1 = \tfrac{1}{2}\, \mathbf{e}^T \mathbf{P} \mathbf{e} \tag{16}$$

and let the overall control law be defined as follows:

$$u = u_f + u_s \tag{17}$$

$$u_f = \frac{1}{\hat{g}(\mathbf{x})} \left[-\hat{f}(\mathbf{x}) + y_d^{(n)} + \lambda^T \mathbf{e} \right] \tag{18}$$

$$= \frac{1}{\sum_{i=1}^{k} \alpha_i \phi_2(\mathbf{x})^T \mathbf{W}_{2i}} \times \left[-\sum_{i=1}^{k} \alpha_i \phi_1(\mathbf{x})^T \mathbf{W}_{1i} + y_d^{(n)} + \lambda^T \mathbf{e} \right] \tag{19}$$

$$u_s = I^* sgn(\mathbf{e}^T \mathbf{P} \mathbf{B}) \frac{1}{g^L(\mathbf{x})} \times \left[|\hat{f}(\mathbf{x})| + f^U(\mathbf{x}) + |\hat{g}(\mathbf{x}) u_f| + g^U(\mathbf{x}) |u_f| + d_m \right] \tag{20}$$

where $I^* = 1$ if $V_1 > V_c$ and $I^* = 0$ if $V_1 \le V_c$, and $sgn(a) = 1$ if $a \ge 0$ and $sgn(a) = 0$ if $a < 0$, and $V_c > 0$ is a constant designed by the user.

Theorem 1: Consider the nonlinear system described by (3) satisfying Assumption 1, and subject to the controller given in (17)–(20). Then $V_1 < V_c$ as $t \to \infty$, where V_c is a positive constant.

Proof: Applying (17) to the plant (3) we obtain

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}(f(\mathbf{x}) + g(\mathbf{x})(u_f + u_s) + d). \tag{21}$$

After simple manipulations, we can obtain the error dynamic equation

$$\begin{aligned} \dot{\mathbf{e}} &= \mathbf{A}\mathbf{e} - \mathbf{B}\lambda^T \mathbf{e} + \mathbf{B}\{\hat{f}(\mathbf{x}) - f(\mathbf{x}) \\ &\quad + (\hat{g}(\mathbf{x}) - g(\mathbf{x}))u_f - g(\mathbf{x})u_s - d\} \qquad (22) \\ &= \hat{\mathbf{A}}\mathbf{e} + \mathbf{B}\{\hat{f}(\mathbf{x}) - f(\mathbf{x}) \\ &\quad + (\hat{g}(\mathbf{x}) - g(\mathbf{x}))u_f - g(\mathbf{x})u_s - d\} \\ e_1 &= \mathbf{C}^T \mathbf{e} \\ \dot{\mathbf{e}} &= \hat{\mathbf{A}}\mathbf{e} + \mathbf{B}\{\tilde{f}(\mathbf{x}) + \tilde{g}(\mathbf{x})u_f - g(\mathbf{x})u_s - d\} \qquad (23) \end{aligned}$$

where $\hat{\mathbf{A}} = \mathbf{A} - \mathbf{B}\lambda^T$, $\tilde{f}(\mathbf{x}) = \hat{f}(\mathbf{x}) - f(\mathbf{x})$, $\tilde{g}(\mathbf{x}) = \hat{g}(\mathbf{x}) - g(\mathbf{x})$, and $e_1 = y_d - x_1$.

Since $\hat{\mathbf{A}}$ is Hurwitz, there exists a positive definite symmetric matrix $\mathbf{P}$, which satisfies the Lyapunov equation

$$\hat{\mathbf{A}}^T \mathbf{P} + \mathbf{P}\hat{\mathbf{A}} = -\mathbf{Q} \tag{24}$$

where $\mathbf{Q}$ is an arbitrary positive definite matrix.

From the Lyapunov function candidate of (16), if we take the time derivative of (16) along the trajectories of error dynamics (23), we obtain

$$\begin{aligned} \dot{V}_1 &= \tfrac{1}{2}\, \dot{\mathbf{e}}^T \mathbf{P}\mathbf{e} + \tfrac{1}{2}\, \mathbf{e}^T \mathbf{P}\dot{\mathbf{e}} \qquad (25) \\ &= \tfrac{1}{2}\, \mathbf{e}^T (\hat{\mathbf{A}}^T \mathbf{P} + \mathbf{P}\hat{\mathbf{A}})\mathbf{e} \\ &\quad + \mathbf{e}^T \mathbf{P}\mathbf{B}\{\tilde{f}(\mathbf{x}) + \tilde{g}(\mathbf{x})u_f - g(\mathbf{x})u_s - d\} \qquad (26) \\ &= -\tfrac{1}{2}\, \mathbf{e}^T \mathbf{Q}\mathbf{e} + \mathbf{e}^T \mathbf{P}\mathbf{B}\{\tilde{f}(\mathbf{x}) + \tilde{g}(\mathbf{x})u_f - g(\mathbf{x})u_s - d\}. \qquad (27) \end{aligned}$$

If we rewrite the above equation

$$\begin{aligned} \dot{V}_1 &= -\tfrac{1}{2}\, \mathbf{e}^T \mathbf{Q}\mathbf{e} + \mathbf{e}^T \mathbf{P}\mathbf{B}\{\tilde{f}(\mathbf{x}) + \tilde{g}(\mathbf{x})u_f - d\} \\ &\quad - \mathbf{e}^T \mathbf{P}\mathbf{B} g(\mathbf{x})u_s \qquad (28) \\ &\le -\tfrac{1}{2}\, \mathbf{e}^T \mathbf{Q}\mathbf{e} + |\mathbf{e}^T \mathbf{P}\mathbf{B}|\{|\hat{f}(\mathbf{x})| + |f(\mathbf{x})| + |\hat{g}(\mathbf{x})u_f| \\ &\quad + |g(\mathbf{x})u_f| + |d|\} - \mathbf{e}^T \mathbf{P}\mathbf{B} g(\mathbf{x})u_s. \qquad (29) \end{aligned}$$

Considering the case of $V_1 > V_c$ and substituting the supervisory controller (20) into (29), we obtain

$$\dot{V}_1 \le -\tfrac{1}{2}\, \mathbf{e}^T \mathbf{Q}\mathbf{e}. \tag{30}$$

Therefore, we always have $V_1 \le V_c$ by using the supervisory controller. ■

The bound of V_1 implies the bound of the magnitude of the error vector $\mathbf{e}$. Moreover, it also means that the state vector $\mathbf{x}$ is bounded. Therefore, the closed-loop system with the controller (17) operates well to stabilize the given system in the sense that the error is not divulged.

In order to derive adaptive laws for these weights of the RBF networks, optimal weights $\mathbf{W}_{1i}^*$ and $\mathbf{W}_{2i}^*$ are defined as

$$\mathbf{W}_{1i}^* = \arg \min_{\mathbf{W}_{1i} \in \Omega_{1i}} \left[\sup_{\mathbf{x} \in \Omega_{\mathbf{x}}} \left| \hat{f}(\mathbf{x}) - f(\mathbf{x}) \right| \right] \tag{31}$$

$$\mathbf{W}_{2i}^* = \arg \min_{\mathbf{W}_{2i} \in \Omega_{2i}} \left[\sup_{\mathbf{x} \in \Omega_{\mathbf{x}}} \left| \hat{g}(\mathbf{x}) - g(\mathbf{x}) \right| \right] \tag{32}$$

where Ω_{1i}, Ω_{2i}, and $\Omega_{\mathbf{x}}$ are compact sets of suitable bounds on $\mathbf{W}_{1i}$, $\mathbf{W}_{2i}$, and $\mathbf{x}$, respectively, and they are defined as

$$\Omega_{1i} = \{\mathbf{W}_{1i} \mid \quad |\mathbf{W}_{1i}| \le W_{1B}\} \tag{33}$$

$$\Omega_{2i} = \{\mathbf{W}_{2i} \mid \quad |\mathbf{W}_{2i}| \le W_{2B}\} \tag{34}$$

$$\Omega_{\mathbf{x}} = \{\mathbf{x} \mid \quad |\mathbf{x}| \le W_{\mathbf{x}}\} \tag{35}$$

where W_{1B}, W_{2B}, and $W_{\mathbf{x}}$ are positive constants.

Define the minimum approximation error as

$$\omega = \hat{f}^*(\mathbf{x}) - f(\mathbf{x}) + (\hat{g}^*(\mathbf{x}) - g(\mathbf{x}))u_f - d \tag{36}$$

where

$$\hat{f}^*(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i \phi_1(\mathbf{x})^T \mathbf{W}_{1i}^* \tag{37}$$

$$\hat{g}^*(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i \phi_2(\mathbf{x})^T \mathbf{W}_{2i}^*. \tag{38}$$

Assumption 2: The term ω cannot be equal to zero in real application even if there is no external disturbance because we cannot use infinitely large hidden-layer units to make $\omega \to 0$ due to the limitation of the computational power and the constraints on hardware equipment. As such, we can assume that $\left| \hat{f}^*(\mathbf{x}) - f(\mathbf{x}) \right| \le \omega_{1B}$ and $|\hat{g}^*(\mathbf{x}) - g(\mathbf{x})| \le \omega_{2B}$, and we may specify these bound values as small values by off-line experiment. We can set these bound values as the maximum approximation errors by performing a task of approximating a known nonlinear function similar to the given problem.

If we represent error dynamics (23) with the minimum approximation error (36), we obtain

$$\begin{aligned} \dot{\mathbf{e}} &= \hat{\mathbf{A}}\mathbf{e} + \mathbf{B}\{\hat{f}(\mathbf{x}) - f(\mathbf{x}) + (\hat{g}(\mathbf{x}) - g(\mathbf{x}))u_f - g(\mathbf{x})u_s - d\} \\ &= \hat{\mathbf{A}} + \mathbf{B}\{\hat{f}(\mathbf{x}) - \hat{f}^*(\mathbf{x}) + (\hat{g}(\mathbf{x}) - \hat{g}^*(\mathbf{x}))u_f - g(\mathbf{x})u_s + \omega\} \end{aligned} \tag{39}$$

$$\begin{aligned} &= \hat{\mathbf{A}}\mathbf{e} - \mathbf{B}g(\mathbf{x})u_s + \mathbf{B}\omega \\ &\quad + \mathbf{B}\left\{ \sum_{i=1}^{k} \alpha_i \phi_1(\mathbf{x})^T \tilde{\mathbf{W}}_{1i} + u_f \sum_{i=1}^{k} \alpha_i \phi_2(\mathbf{x})^T \tilde{\mathbf{W}}_{2i} \right\} \end{aligned} \tag{40}$$

where $\tilde{\mathbf{W}}_{1i} = \mathbf{W}_{1i} - \mathbf{W}_{1i}^*$, and $\tilde{\mathbf{W}}_{2i} = \mathbf{W}_{2i} - \mathbf{W}_{2i}^*$.

Theorem 2: Under Assumptions 1 and 2, if the following control law and adaptation laws are applied to the nonlinear system (3), then the closed-loop system is globally asymptotically stable and all the involved signals are bounded within specified regions.

Control Law:

$$u = u_f + u_s, \tag{41}$$

where u_f is described by (19), and u_s is given by the following equation:

$$\begin{aligned} u_s = I^* sgn(\mathbf{e}^T\mathbf{PB}) \frac{1}{g^L(\mathbf{x})} \Big[&|\hat{f}(\mathbf{x})| + f^U(\mathbf{x}) + d_m \\ &+ |\hat{g}(\mathbf{x})u_f| + g^U(\mathbf{x})|u_f| + \omega_{1B} + \omega_{2B}|u_f| \Big]. \end{aligned} \tag{42}$$

Adaptation Laws: If $|\mathbf{W}_{1i}| < W_{1B}$ or ($|\mathbf{W}_{1i}| = W_{1B}$ and $\mathbf{e}^T\mathbf{PB}\phi_1^T(\mathbf{x})\mathbf{W}_{1i} \ge 0$)

$$\dot{\mathbf{W}}_{1i} = -\gamma_1 \alpha_i \phi_1(\mathbf{x})\mathbf{B}^T\mathbf{Pe}. \tag{43}$$

If ($|\mathbf{W}_{1i}| = W_{1B}$ and $\mathbf{e}^T\mathbf{PB}\phi_1^T(\mathbf{x})\mathbf{W}_{1i} < 0$)

$$\dot{\mathbf{W}}_{1i} = -\gamma_1 \alpha_i \phi_1(\mathbf{x})\mathbf{B}^T\mathbf{Pe} + \gamma_1 \alpha_i \mathbf{e}^T\mathbf{PB} \frac{\mathbf{W}_{1i}\mathbf{W}_{1i}^T\phi_1(\mathbf{x})}{|\mathbf{W}_{1i}|^2}. \tag{44}$$

If $|\mathbf{W}_{2i}| < W_{2B}$ or ($|\mathbf{W}_{2i}| = W_{2B}$ and $\mathbf{e}^T\mathbf{PB}\phi_2^T(\mathbf{x})\mathbf{W}_{2i}u_f \ge 0$)

$$\dot{\mathbf{W}}_{2i} = -\gamma_2 \alpha_i \phi_2(\mathbf{x})\mathbf{B}^T\mathbf{Pe}u_f. \tag{45}$$

If ($|\mathbf{W}_{2i}| = W_{2B}$ and $\mathbf{e}^T\mathbf{PB}\phi_2^T(\mathbf{x})\mathbf{W}_{2i}u_f < 0$)

$$\begin{aligned} \dot{\mathbf{W}}2i = &-\gamma_2 \alpha_i \phi_2(\mathbf{x})\mathbf{B}^T\mathbf{Pe}u_f \\ &+ \gamma_2 \alpha_i \mathbf{e}^T\mathbf{PB} \frac{\mathbf{W}_{2i}\mathbf{W}_{2i}^T\phi_2(\mathbf{x})u_f}{|\mathbf{W}_{2i}|^2} \end{aligned} \tag{46}$$

where γ_1 and γ_2 are adaptation gains, respectively.

Proof: Let us define an overall Lyapunov function candidate as

$$V = \frac{1}{2}\mathbf{e}^T\mathbf{Pe} + \frac{1}{2\gamma_1} \sum_{i=1}^{k} \tilde{\mathbf{W}}_{1i}^T \tilde{\mathbf{W}}_{1i} + \frac{1}{2\gamma_2} \sum_{i=1}^{k} \tilde{\mathbf{W}}_{2i}^T \tilde{\mathbf{W}}_{2i}. \tag{47}$$

The time derivative of (47) is

$$\begin{aligned} \dot{V} = &\tfrac{1}{2}\dot{\mathbf{e}}^T\mathbf{Pe} + \tfrac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}} \\ &+ \frac{1}{\gamma_1} \sum_{i=1}^{k} \dot{\tilde{\mathbf{W}}}_{1i}^T \tilde{\mathbf{W}}_{1i} + \frac{1}{\gamma_2} \sum_{i=1}^{k} \dot{\tilde{\mathbf{W}}}_{2i}^T \tilde{\mathbf{W}}_{2i}. \end{aligned} \tag{48}$$

Using the fact that $\dot{\tilde{\mathbf{W}}}_{1i} = \dot{\mathbf{W}}_{1i}$ and $\dot{\tilde{\mathbf{W}}}_{2i} = \dot{\mathbf{W}}_{2i}$ with the error dynamics (40), we obtain

$$\begin{aligned} \dot{V} = &-\tfrac{1}{2}\mathbf{e}^T\mathbf{Qe} - \mathbf{e}^T\mathbf{PB}g(\mathbf{x})u_s + \mathbf{e}^T\mathbf{PB}\omega \\ &+ \mathbf{e}^T\mathbf{PB} \sum_{i=1}^{k} \alpha_i \phi_1(\mathbf{x})^T \tilde{\mathbf{W}}_{1i} \\ &+ \mathbf{e}^T\mathbf{PB}u_f \sum_{i=1}^{k} \alpha_i \phi_2(\mathbf{x})^T \tilde{\mathbf{W}}_{2i} \\ &+ \frac{1}{\gamma_1} \sum_{i=1}^{k} \dot{\mathbf{W}}_{1i}^T \tilde{\mathbf{W}}_{1i} + \frac{1}{\gamma_2} \sum_{i=1}^{k} \dot{\mathbf{W}}_{2i}^T \tilde{\mathbf{W}}_{2i} \end{aligned} \tag{49}$$

$$= -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \mathbf{e}^T\mathbf{P}\mathbf{B}g(\mathbf{x})u_s + \mathbf{e}^T\mathbf{P}\mathbf{B}\omega + \sum_{i=1}^{k}\left(\frac{1}{\gamma_1}\dot{\mathbf{W}}_{1i}^T + \mathbf{e}^T\mathbf{P}\mathbf{B}\alpha_i\phi_1^T(\mathbf{x})\right)\tilde{\mathbf{W}}_{1i} + \sum_{i=1}^{k}\left(\frac{1}{\gamma_2}\dot{\mathbf{W}}_{2i}^T + \mathbf{e}^T\mathbf{P}\mathbf{B}\alpha_i\phi_2^T(\mathbf{x})u_f\right)\tilde{\mathbf{W}}_{2i}. \quad (50)$$

If we substitute adaptation laws (43)–(46) into (50), we obtain

$$\dot{V} \le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \mathbf{e}^T\mathbf{P}\mathbf{B}g(\mathbf{x})u_s + \mathbf{e}^T\mathbf{P}\mathbf{B}\omega \quad (51)$$

$$\le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \mathbf{e}^T\mathbf{P}\mathbf{B}g(\mathbf{x})u_s + \left|\mathbf{e}^T\mathbf{P}\mathbf{B}\right||\omega|. \quad (52)$$

In the case of $V > V_c$, the supervisory controller (42) is activated, and we have

$$\dot{V} \le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \mathbf{e}^T\mathbf{P}\mathbf{B}g(\mathbf{x})u_s + \left|\mathbf{e}^T\mathbf{P}\mathbf{B}\right||\omega|$$

$$\le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \mathbf{e}^T\mathbf{P}\mathbf{B}g(\mathbf{x})u_s + \left|\mathbf{e}^T\mathbf{P}\mathbf{B}\right|\{|f^*| + |f| + |g^*u_f| + |g||u_f| + |d|\} \quad (53)$$

$$\le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} - \mathbf{e}^T\mathbf{P}\mathbf{B}g(\mathbf{x})u_s + \left|\mathbf{e}^T\mathbf{P}\mathbf{B}\right| \times \left\{\left|\hat{f}\right| + |f| + |\hat{g}u_f| + |g||u_f| + |\omega_{1B}| + |\omega_{2B}||u_f| + |d|\right\} \quad (54)$$

$$\le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e}. \quad (55)$$

Therefore, we always have $V \le V_c$ by using the supervisory controller. If $V \le V_c$, the supervisory controller is deactivated. Since $g(\mathbf{x}) > 0$, we have $g(\mathbf{x})\mathbf{e}^T\mathbf{P}\mathbf{B}u_s \ge 0$ from (42). Hence, from Rayleighs inequality of $(1/2)\lambda_{\min}(\mathbf{Q})|\mathbf{e}|^2 \le (1/2)\mathbf{e}^T\mathbf{Q}\mathbf{e} \le (1/2)\lambda_{\max}(\mathbf{Q})|\mathbf{e}|^2$, we obtain

$$\dot{V} \le -\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e} + \mathbf{e}^T\mathbf{P}\mathbf{B}\omega$$

$$\le -\frac{1}{2}\lambda_{\min}(\mathbf{Q})|\mathbf{e}|^2 + \mathbf{e}^T\mathbf{P}\mathbf{B}\omega \quad (56)$$

$$= -\frac{1}{2}\lambda_{\min}(\mathbf{Q})|\mathbf{e}|^2 + \frac{1}{2}|\mathbf{e}|^2 - \frac{1}{2}|\mathbf{e}|^2 + \mathbf{e}^T\mathbf{P}\mathbf{B}\omega - \frac{1}{2}|\mathbf{P}\mathbf{B}\omega|^2 + \frac{1}{2}|\mathbf{P}\mathbf{B}\omega|^2 \quad (57)$$

$$= -\left(\frac{1}{2}\lambda_{\min}(\mathbf{Q}) - 1\right)|\mathbf{e}|^2 + \frac{1}{2}|\mathbf{P}\mathbf{B}\omega|^2 - \frac{1}{2}\left(|\mathbf{e}|^2 - 2\mathbf{e}^T\mathbf{P}\mathbf{B}\omega + |\mathbf{P}\mathbf{B}\omega|^2\right) \quad (58)$$

$$\le -\left(\frac{1}{2}\lambda_{\min}(\mathbf{Q}) - 1\right)|\mathbf{e}|^2 + \frac{1}{2}|\mathbf{P}\mathbf{B}|^2|\omega|^2 \quad (59)$$

where $\lambda_{\min}(\mathbf{Q})$ is the minimum eigenvalue of $\mathbf{Q}$. By integrating both sides of (59) with specifying $\mathbf{Q}$ such that $\lambda_{\min}(\mathbf{Q}) > 1$ and some manipulation, we obtain

$$\int_0^t |\mathbf{e}(\tau)|^2\, d\tau \le \frac{2}{\lambda_{\min}(\mathbf{Q}) - 1}(V(0) - V(t)) + \frac{|\mathbf{P}\mathbf{B}|^2}{\lambda_{\min}(\mathbf{Q}) - 1}\int_0^t |\omega(\tau)|^2\, d\tau. \quad (60)$$

Therefore, if $\omega(\tau) \in L_2$, we have $\mathbf{e} \in L_2$, where L_2 means squared integrable [25].

Now, to prove $|\mathbf{W}_{1i}| \le W_{1B}$, let $V_{\mathbf{W}_{1i}} = (1/2)\mathbf{W}_{1i}^T\mathbf{W}_{1i}$. If (43) is true, we have either $|\mathbf{W}_{1i}| \le W_{1B}$ or $\dot{V}_{\mathbf{W}_{1i}} = -\gamma_1\alpha_i\mathbf{W}_{1i}^T\phi_1(\mathbf{x})\mathbf{B}^T\mathbf{P}\mathbf{e} \le 0$ for $|\mathbf{W}_{1i}| = W_{1B}$, i.e., we always have $|\mathbf{W}_{1i}| \le W_{1B}$. If (44) is true, we have $|\mathbf{W}_{1i}| = W_{1B}$, and $\dot{V}_{\mathbf{W}_{1i}} = -\gamma_1\alpha_i\mathbf{W}_{1i}^T\phi_1(\mathbf{x})\mathbf{B}^T\mathbf{P}\mathbf{e} + \gamma_1\alpha_i\mathbf{e}^T\mathbf{P}\mathbf{B}(\mathbf{W}_{1i}^T\mathbf{W}_{1i}\mathbf{W}_{1i}^T\phi_1(\mathbf{x})/|\mathbf{W}_{1i}|^2) = 0$. Therefore, we prove that $|\mathbf{W}_{1i}| \le W_{1B}$, $t \ge 0$.

With the similar technique, we can show that $|\mathbf{W}_{2i}| \le W_{2B}$, $t \ge 0$.

To prove the boundedness of the system state, we use the definition of the error vector, $\mathbf{e} = \mathbf{y}_d - \mathbf{x}$, and we have $|\mathbf{x}| \le |\mathbf{y}_d| + |\mathbf{e}|$. Using Rayleighs inequality and Theorem 1, we have $(1/2)\lambda_{\min}(\mathbf{P})|\mathbf{e}|^2 \le (1/2)\mathbf{e}^T\mathbf{P}\mathbf{e} \le V_c$ and $|\mathbf{e}| \le \sqrt{2V_c/\lambda_{\min}(\mathbf{P})}$. Therefore, the magnitude of the system state vector is bounded as $|\mathbf{x}| \le |\mathbf{y}_d| + \sqrt{2V_c/\lambda_{\min}(\mathbf{P})} = W_{\mathbf{x}}$.

The boundedness of control signals can be shown easily since all the terms in the control inputs are bounded from the previous discussion.

Since $\hat{f}(\mathbf{x})$ and $\hat{g}(\mathbf{x})$ are weighted averages of the elements of $\mathbf{W}_{1i}$ and $\mathbf{W}_{2i}$, respectively, we have

$$\left|\hat{f}(\mathbf{x})\right| = \left|\sum_{i=1}^{k}\alpha_i\phi_1(\mathbf{x})^T\mathbf{W}_{1i}\right| \le \sum_{i=1}^{k}\alpha_i\left|\phi_1(\mathbf{x})^T\mathbf{W}_{1i}\right| \le \sum_{i=1}^{k}\alpha_i|\phi_1(\mathbf{x})|W_{1B} = W_{\hat{f}} \quad (61)$$

and similarly

$$|\hat{g}(\mathbf{x})| = \left|\sum_{i=1}^{k}\alpha_i\phi_2(\mathbf{x})^T\mathbf{W}_{2i}\right| \le \sum_{i=1}^{k}\alpha_i\left|\phi_2(\mathbf{x})^T\mathbf{W}_{2i}\right| \le \sum_{i=1}^{k}\alpha_i|\phi_2(\mathbf{x})|W_{2B} = W_{\hat{g}} \quad (62)$$

and

$$|\hat{g}(\mathbf{x})| \ge \epsilon_g > 0. \quad (63)$$

From (19), we obtain

$$|u_f| \le \frac{1}{|\hat{g}(\mathbf{x})|}\left[\left|\hat{f}(\mathbf{x})\right| + \left|y_d^{(n)}\right| + |\lambda||\mathbf{e}|\right] \le \frac{1}{|\hat{g}(\mathbf{x})|}\left[\left|\hat{f}(\mathbf{x})\right| + \left|y_d^{(n)}\right| + |\lambda|\sqrt{\frac{2V_c}{\lambda_{\min}(\mathbf{P})}}\right] \le \frac{1}{\epsilon_g}\left[W_{\hat{f}} + \left|y_d^{(n)}\right| + |\lambda|\sqrt{\frac{2V_c}{\lambda_{\min}(\mathbf{P})}}\right]. \quad (64)$$

Therefore, u_f is bounded.

For the supervisory controller (42), we obtain the following relationship:

$$|u_s| \le \frac{1}{g^L(\mathbf{x})}\left[\left|\hat{f}(\mathbf{x})\right| + f^U(\mathbf{x}) + |\hat{g}(\mathbf{x})||u_f| + g^U(\mathbf{x})|u_f| + d_m + \omega_{1B} + \omega_{2B}|u_f|\right]$$

$$\le \frac{1}{g^L(\mathbf{x})}\left\{W_{\hat{f}} + f^U(\mathbf{x}) + d_m + \omega_{1B} + \frac{W_{\hat{g}} + g^U(\mathbf{x}) + \omega_{2B}}{\epsilon_g} \times \left[W_{\hat{f}} + \left|y_d^{(n)}\right| + |\lambda|\sqrt{\frac{2V_c}{\lambda_{\min}(\mathbf{P})}}\right]\right\}. \quad (65)$$

Therefore, u_s is bounded.

From (64) and (65), we can prove boundedness of the overall control signal of $u = u_f + u_s$, (41) such that

$$|u| \leq |u_f| + |u_s|. \tag{66}$$

To use Barbalat's lemma, let us check the uniform continuity of $\dot{V}$. The derivative of $\dot{V}$ for $V < V_c$ can be represented as

$$\ddot{V} \leq -\mathbf{e}^T \mathbf{Q}\dot{\mathbf{e}} + \dot{\mathbf{e}}^T \mathbf{PB}\omega. \tag{67}$$

This shows that $\ddot{V}$ is bounded. Hence, $\dot{V}$ is uniformly continuous. Application of Barbalat's lemma indicates that $\dot{V}$ converges to zero as time goes to infinity. Therefore, $\mathbf{e}$ is globally asymptotically stable. Thus the proof is complete. ∎

From Theorem 2, switching between multiple models does not affect the closed-loop stability, i.e., under any permissible switching or blending rule, the proposed control algorithm results in an overall system stability that all signals remain bounded.

One choice of the scheme for blending multiple models from the predicted errors [26] may be used. Let the predicted output of the ith neural network model $\hat{y}_i$ be calculated by the following model:

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_i &= \mathbf{A}\mathbf{x} + \mathbf{B}\left(\phi_1(\mathbf{x})^T \mathbf{W}_{1i} + \phi_2(\mathbf{x})^T \mathbf{W}_{2i} u\right) \\ \hat{y}_i &= \mathbf{C}^T \hat{\mathbf{x}}_i \end{aligned} \tag{68}$$

where $u = u_f + u_s$. If we define $\hat{e}_i = y - \hat{y}_i$ as the difference between the predicted output and the actual output, then we may use the following blending coefficient, α_i, i.e., significance parameter

$$\alpha_i = \frac{e^{-|\hat{e}_i|^2/\sigma^2}}{\sum_{j=1}^{k} e^{-|\hat{e}_j|^2/\sigma^2}}. \tag{69}$$

The soft-max transforms the errors using the exponential function and then normalizes these values so that α_i has the property that they lie between 0 and 1, and $\sum_{i=1}^{k} \alpha_i = 1$.

In this paper, we used a hybrid algorithm as shown in the following procedure. One is to switch between multiple models in learning each model for memorizing control skill, and the other is to blend existing multiple models for a new task.

- **Step 0:** `Initialize all the RBF network weights randomly.`
- **Step 1:** [**Switching**] `With user interaction, train each RBF network to learn specific tasks.`
- **Step 2:** [**Blending**] `For the new task, blending of multiple models based on the predicted error using (69).`

IV. Simulation

In this section, we will apply our proposed control scheme to control a cartpole system for the pole to track a given trajectory. We will compare an adaptive multiple neural network controller with an adaptive neural network controller for the cartpole system with changing mass of the pole.

The dynamic equations of the cartpole system [27] are

$$y^{(2)} = f(y, \dot{y}) + g(y, \dot{y})u + d \tag{70}$$

where y is the angle of the pole. Let $x_1 = y$ and $x_2 = \dot{y}$; then, the nonlinear terms are

$$f = \frac{G \sin x_1 - \frac{m l x_2^2 \cos x_1 \sin x_1}{m_c + m}}{l\left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m}\right)} \tag{71}$$

$$g = \frac{\frac{\cos x_1}{m_c + m}}{l\left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m}\right)} \tag{72}$$

where $G = 9.8$ m/s^2 is the acceleration due to gravity; m_c is the mass of the cart; l is the half length of the pole; m is the mass of the pole; and u is the control input. Here, we assumed that $m_c = 1.0$ kg, $l = 0.5$ m and that m can be changed between 0.1 and 1.0 kg. In addition, we assumed that the external disturbance is $0.05 \sin t$.

The control objective is to control the state x_1 of the system to track the reference trajectory $y_d = (\pi/30) \sin t$, and during the operation, the mass of the pole changes as follows:

$$m = \begin{cases} 0.1 \text{ kg}, & \text{if } 0 \leq t \leq t_1, \quad \text{Task 1} \\ 1.0 \text{ kg}, & \text{if } t_1 < t \leq t_2, \quad \text{Task 2} \\ 0.1 \text{ kg}, & \text{if } t_2 < t \leq t_3, \quad \text{Task 1} \\ 0.5 \text{ kg}, & \text{if } t_3 < t \leq t_4, \quad \text{Task 3} \end{cases}$$

where $t_1 = 80.0$, $t_2 = 150.0$, $t_3 = 220.0$, and $t_4 = 300.0$ s. Tasks 1 and 2 are assumed to be known tasks before operation, and Task 3 is assumed to be an unknown task for which a model is not specified.

From Assumption 1, we assumed that the bounds f^U, g^U, and g^L are known *a priori*. By checking the system dynamics for $|x_1| \leq \pi/30$, $|x_2| \leq \pi/30$, and $m \in [0.1, 1.0]$, we can obtain the approximate exact bounds as $f^U = 5.8$, $g^U = 1.464$, and $g^L = 1.174$. Therefore, in our simulation, we assumed empirically that $f^U = 10.0$, $g^U = 2.0$, and $g^L = 0.3$. In addition, we used $\lambda = [4, 4]^T$ and selected $\mathbf{Q} = \begin{bmatrix} 10 & 13 \\ 13 & 28 \end{bmatrix}$ of which the minimum eigenvalue is greater than 1. Since $\hat{\mathbf{A}} = \mathbf{A} - \mathbf{B}\lambda^T = \begin{bmatrix} 0 & 1 \\ -4 & -4 \end{bmatrix}$, we obtain $\mathbf{P} = \begin{bmatrix} 7.25 & 1.25 \\ 1.25 & 3.8125 \end{bmatrix}$ by solving the Lyapunov equation. Since there are two known tasks, we designed two RBF network pairs. Each network pair consists of two RBF networks, where one approximates f and the other approximates g. We placed 121 basis functions evenly on the input region with $\sigma_{ji} = 0.2$, $j = 1, 2$, $i = 1, 2, \ldots, 121$. We used adaptation gains as $\gamma_1 = \gamma_2 = 10.0$ through the simulation. We set the bound of external disturbance as $d_m = 0.05$, and the approximation errors between the optimal networks and the concerned nonlinear functions are set as small numbers $\omega_{1B} = \omega_{2B} = 0.001$.

Two groups of experiments were performed: 1) adaptive neural network controller (ANNC), which has a single RBF network pair, and 2) the proposed adaptive multiple neural network controller (AMNNC), which has two RBF network pairs.

TABLE I
PERFORMANCE COMPARISON BETWEEN ANNC AND AMNNC

	ANNC		AMNNC	
Interval	Max. $\|\mathbf{e}\|$	IAE	Max. $\|\mathbf{e}\|$	IAE
$t \in [0, 330]$	0.3114	2.7341	0.3125	2.5540
$t \in [150, 220]$	0.0189	0.3286	0.0091	0.1775
$t \in [220, 300]$	0.0151	0.3692	0.0142	0.2481

TABLE II
PERCENTAGE IMPROVEMENT IN IAE PERFORMANCE OF AMNNC OVER ANNC

Interval	$\left(\frac{\mathrm{IAE}_{\mathrm{ANNC}}-\mathrm{IAE}_{\mathrm{AMNNC}}}{\mathrm{IAE}_{\mathrm{ANNC}}}\right) \times 100\%$
$t \in [0, 330]$	6.587 %
$t \in [150, 220]$	45.983 %
$t \in [220, 300]$	32.801 %

We compared the control performance for the two groups of experiments (ten trials for each controller) by averaging the integrals of the absolute magnitude of the error (IAE), which is written as

$$\mathrm{IAE} = \int_{T_0}^{T_1} |\mathbf{e}(t)| \, dt. \tag{73}$$

Table I presents the comparison of error measurements between ANNC and AMNNC. Table II gives the percentage improvements of the IAE provided by AMNNC over ANNC, which proves that AMNNC is much more effective in tracking performance when some tasks are repeated.

Figs. 1–3 show the tracking errors when ANNC is applied. Initially, the neural network does not have dynamic knowledge of the system and the adaptive scheme changes network weights when Task 1 is given for $0 \leq t \leq 80.0$. As time passes, the performance is improved since the neural network approximates nonlinearity of the system for Task 1. When Task 2 is given for $80.0 < t \leq 150.0$, the system dynamics for this task is different from that of the learned neural network at the time. Therefore, some transient errors appeared around $t = 80.0$, and the neural network is adapted for this new task minimizing tracking errors. At $t = 150.0$, the task is changed into the old Task 1. There are also some transient errors around at that time as shown in Fig. 2 since the neural network forgot control skill for Task 1 while it learned the new control skill for Task 2. The conventional adaptive neural network controller repeats its adaptation for the given task whether it is a new task or an old one. As shown in Fig. 3, for the new Task 3—around $t = 150.0$—tracking errors are slowly minimized since the neural network has to be adapted from the previous dynamic knowledge. If there are large task variations, the conventional ANNC will give large transient errors when the task is changed.

Figs. 4–6 show the tracking errors when AMNNC is applied. For $0 \leq t \leq 150.0$, the performance is similar to that of ANNC. Since Tasks 1 and 2 are known, significance parameters are set by the user (see Fig. 7). Control skill for each task is stored

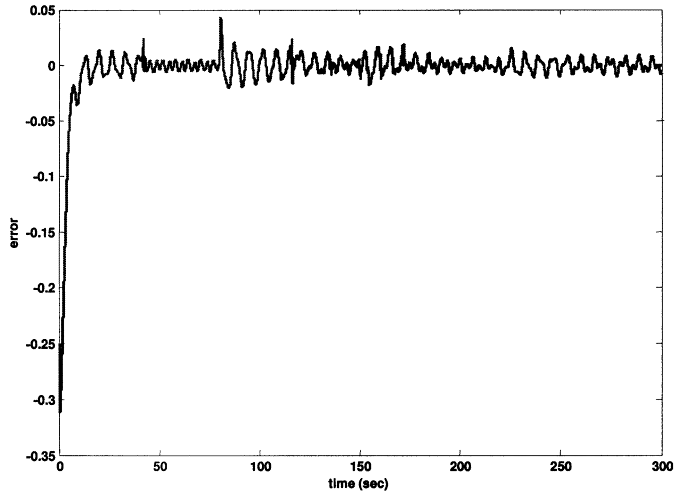

Fig. 1. Tracking error for desired trajectories (ANNC).

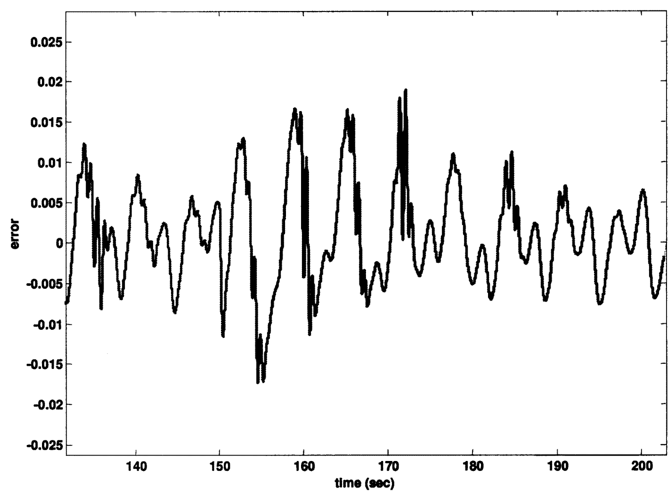

Fig. 2. Tracking error for repeated Task 1 (ANNC).

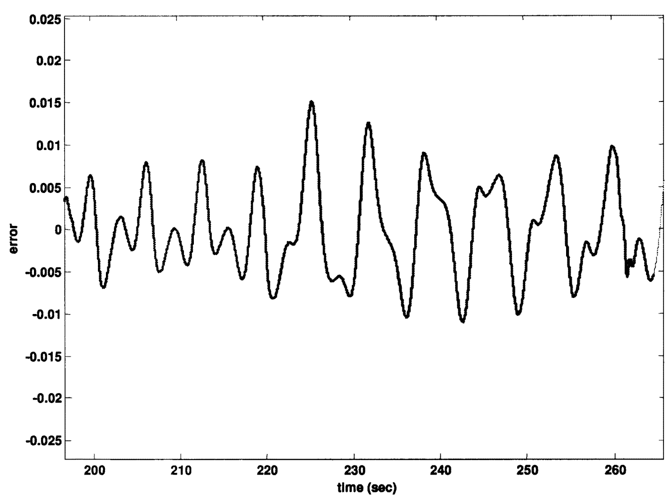

Fig. 3. Tracking error for Task 3 (ANNC).

for each neural network pair, and appropriate control skill can be selected for the repeated task. As shown in Fig. 5, there are small transient errors around $t = 150.0$ when the task is changed into the old Task 1, for which dynamic information is memorized with one neural network pair. For the unknown

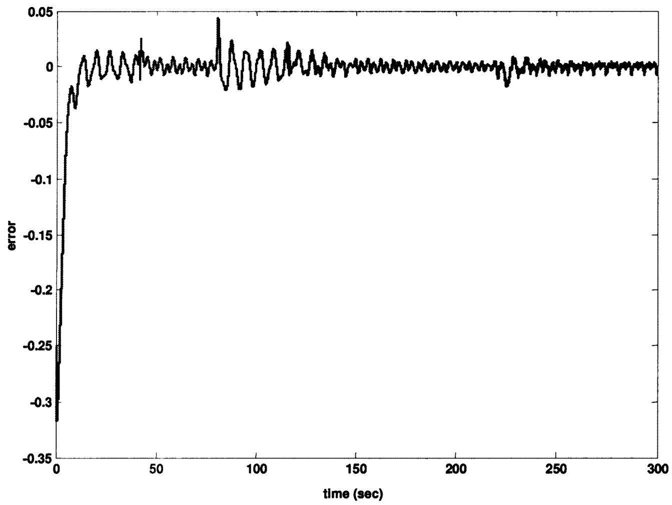

Fig. 4. Tracking error for desired trajectories (AMNNC).

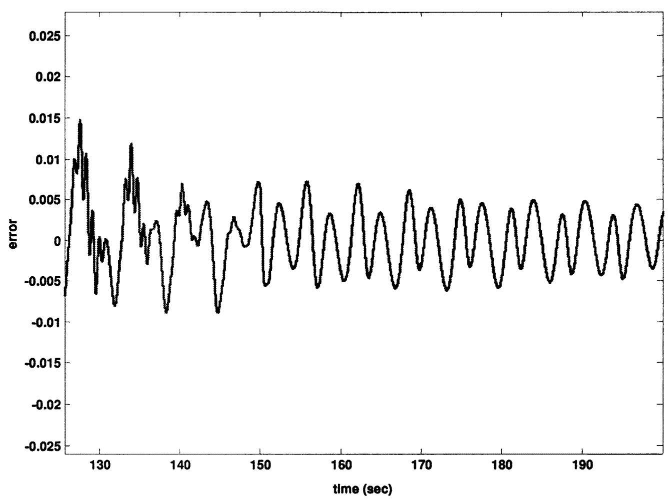

Fig. 5. Tracking error for repeated Task 1 (AMNNC).

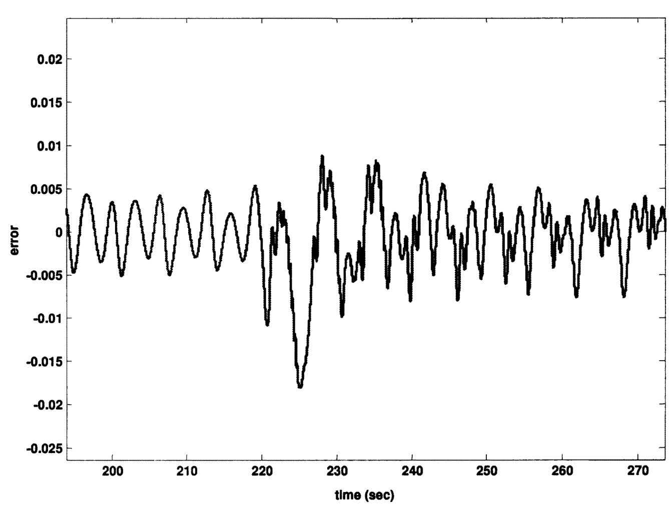

Fig. 6. Tracking error for Task 3 (new task) (AMNNC).

new Task 3, the proposed blending of existing neural networks using predicted errors is shown in Figs. 6 and 8. From these experiments, the proposed AMNNC shows better performance over the conventional ANNC for the repeated tasks, which is

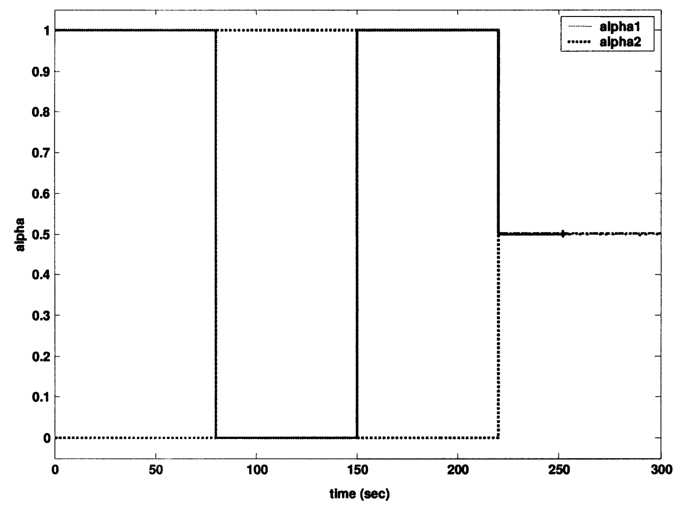

Fig. 7. Significance parameter (AMNNC).

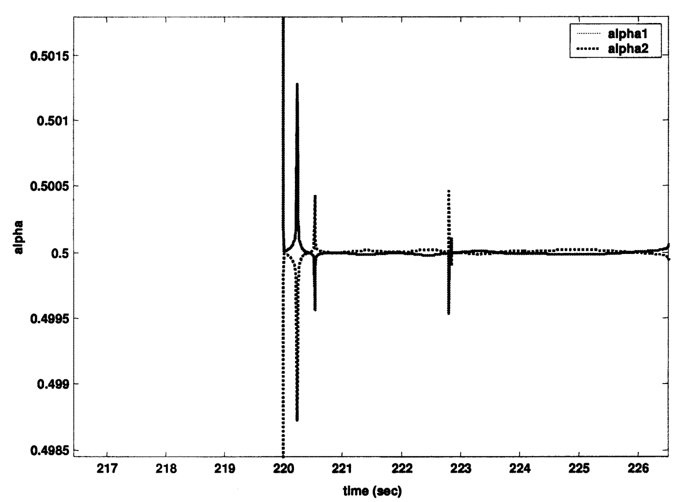

Fig. 8. Significance parameter for Task 3 (new task) (AMNNC).

needed in industrial applications where the load changes from time to time.

V. Conclusion

In this paper, an adaptive multiple neural network control method was developed for the control of a certain unknown nonlinear dynamic system with a supervisory controller. Multiple neural networks are used to approximate the changing system dynamics for various tasks. For the case of repeated jobs, the proposed control scheme is effective because of its capability to memorize control skill for each task with the neural network. Transient errors at the time of changing tasks will be attenuated significantly when the new task occurs repeatedly as shown in the experiment. The Lyapunov function-based design of adaptation laws guarantees the global stability of the closed-loop system. The implementation with the feedforward neural network instead of the RBF network is possible by following Lewis' work [28]

In this paper, we have fixed the number of models and we assumed that we can train each model for each task by user interaction. The on-line generation and pruning of models will be interesting further studies.

REFERENCES

[1] S. S. Sastry and A. Isidori, "Adaptive control of linearization systems," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 1123–1131, Nov. 1989.

[2] K. Funahashi, "On the approximate realization of continuous mapping by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 359–366, 1989.

[3] J. L. Castro, "Fuzzy logic controllers are universal approximators," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 629–635, Apr. 1995.

[4] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, U.K.: Taylor and Francis, 1999.

[5] J. S. Albus, "A new approach to manipulator control: The cerebella model articulation controller (CMAC)," *Trans. ASME, J. Dyna. Syst., Measurement, Contr.*, vol. 63, no. 3, pp. 220–227, 1975.

[6] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837–863, Nov. 1992.

[7] R. Carelli, E. F. Camacho, and D. Patiño, "A neural network based feedforward adaptive controller for robot," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 1281–1288, Sept. 1995.

[8] G. A. Rovithakis and M. A. Christodulou, "Neural adaptive regulation of unknown nonlinear dynamical systems," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 810–822, Oct. 1997.

[9] Y. G. Leu, T. T. Lee, and W. Y. Wang, "Observer-based adaptive fuzzy neural control for unknown nonlinear dynamical systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, pp. 583–591, Oct. 1999.

[10] S. Seshagiri and H. K. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *IEEE Trans. Neural Networks*, vol. 11, pp. 69–79, Jan. 2000.

[11] F. Sun, Z. Sun, and P.-Y. Woo, "Neural network-based adaptive controller design of robotic manipulators with an observer," *IEEE Trans. Neural Networks*, vol. 12, pp. 54–67, Jan. 2001.

[12] L. X. Wang, "Stable adaptive fuzzy control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 146–155, May 1993.

[13] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 339–359, Aug. 1996.

[14] M. C. M. Teixeira and S. H. Zak, "Stabilizing controller design for uncertain nonlinear systems using fuzzy model," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 133–142, Apr. 1999.

[15] C.-H. Wang, H.-L. Liu, and T.-C. Lin, "Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems," *IEEE Trans. Fuzzy Syst.*, vol. 10, pp. 39–49, Feb. 2002.

[16] M. Athans *et al.*, "The stochastic control of the F-8C aircraft using a multiple model adaptive control (MMAC) method—Part I: Equilibrium flight," *IEEE Trans. Automat. Contr.*, vol. 22, pp. 768–780, 1977.

[17] C. Yu, R. J. Roy, H. Kaufman, and B. W. Bequette, "Multiple-model adaptive predictive control of mean arterial pressure and cardiac output," *IEEE Trans. Biomed. Eng.*, vol. 39, pp. 765–778, Aug. 1992.

[18] M. Fu and B. R. Barmish, "Adaptive stabilization of linear systems via switching control," *IEEE Trans. Automat. Contr.*, vol. AC-31, pp. 1097–1103, 1986.

[19] A. S. Morse, D. Q. Mayne, and G. C. Goodwin, "Applications of hysteresis switching in parameter adaptive control," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1343–1354, Sept. 1992.

[20] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 171–187, Feb. 1997.

[21] D. J. Leith and W. E. Leithead, "Analytic framework for blended multiple model systems using linear local models," *Int. J. Contr.*, vol. 72, pp. 605–619, 1999.

[22] D. Shukla, D. M. Dawson, and F. W. Paul, "Multiple neural-network-based adaptive controller using orthonormal activation function neural networks," *IEEE Trans. Neural Networks*, vol. 10, pp. 1494–1501, Nov. 1999.

[23] H. D. Patiño, R. Carelli, and B. R. Kuchen, "Neural networks for advanced control of robot manipulators," *IEEE Trans. Neural Networks*, vol. 13, pp. 343–354, Mar. 2002.

[24] M. Vidyasagar, *Nonlinear Systems Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[25] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[26] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural Networks*, vol. 11, pp. 1317–1329, 1998.

[27] S. Geva and J. Sitte, "A cartpole experiment benchmark for trainable controller," *IEEE Control Syst. Mag.*, pp. 40–51, Oct. 1993.

[28] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 589–600, July 1998.

Choon-Young Lee was born in Korea on September 1, 1973. He received B.S. degree in electronic engineering from Hanyang University, Seoul, Korea, in 1996 and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 1998, where he is currently pursuing the Ph.D. degree.

From 1998 to 2000, he had developed a gait training robot system as an researcher on the project sponsored by the Ministry of Health and Welfare, Korea. He also participated in the development of an emotional robot from 2001 to 2002. His research interests are in the field of neural network application on the design of control system, ITS, sensor fusion, and robotics.

Mr. Lee is a member KSAE.

Ju-Jang Lee (M'86–SM'98) was born in Seoul, Korea, on November 14, 1948. He received the B.S. and M.S. degrees, both in electrical engineering, from Seoul National University in 1973 and 1977, respectively, and the Ph.D. degree in electrical engineering from the University of Wisconsin, Madison, in 1984.

From 1977 to 1978, he was a research engineer at the Korean Electric Research and Testing Institute, Seoul. From 1978 to 1979, he was a design and processing engineer at G. T. E. Automatic Electric Co., Waukesha, WI. In 1983, he was briefly the project engineer with the Research and Development Department of the Wisconsin Electric Power Co., Milwaukee, WI. He joined the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 1984, where he is currently a Professor. In 1987, he was a visiting professor at the Robotics Laboratory of Imperial College Science and Technology, London, U.K. From 1991 to 1992, he was a visiting scientist at the Robotics of Carnegie Mellon University, Pittsburgh, PA. His research interests are in the areas of intelligent control of mobile robot, service robotics for the disabled, space robotics, evolutionary computation, variable structure control, chaotic control system, electronic control units for automobiles, and power system stabilizers.

Dr. Lee is a member of the IEEE Robotics and Automation Society, the IEEE Evolutionary Computation Society, the IEEE Industrial Electronics Society, KIEE, KITE, and KISS. He is a vice president of ICASE and a director of SICE in Japan.