Neural Networks, Vol. 6, pp. 933–946, 1993
Printed in the USA.

0893-6080/93 $6.00 + .00

ORIGINAL CONTRIBUTION

Neural Network Control for a Closed-Loop System Using *Feedback-Error-Learning*

HIROAKI GOMI* AND MITSUO KAWATO†

*†ATR Human Information Processing Research Laboratories
†Hokkaido University

(*Received* 3 *August* 1992; *revised and accepted* 14 *January* 1993)

Abstract—*This paper presents new learning schemes using* feedback-error-learning *for a neural network model applied to adaptive nonlinear* feedback *control.* Feedback-error-learning *was proposed as a learning method for forming a* feedforward *controller that uses the output of a feedback controller as the error for training a neural network model. Using new schemes for nonlinear feedback control, the actual responses after learning correspond to the desired responses which are defined by an inverse reference model implemented as a conventional feedback controller. In this respect, these methods are similar to Model Reference Adaptive Control (MRAC) applied to linear or linearized systems. It is shown that* learning impedance control *is derived when one proposed scheme is used in Cartesian space. We show the results of applying these learning schemes to an inverted pendulum and a 2-link manipulator. We also discuss the convergence properties of the neural network models employed in these learning schemes by applying the Lyapunov method to the averaged equations associated with the stochastic differential equations which describe the system dynamics.*

Keywords—Neural network control, Adaptive control, Feedback-control learning, Reference model, Impedance control, Feedback-error-learning, Cerebellum motor control learning.

1. INTRODUCTION

Adaptive control theory has been applied to nonlinear robot control as a method of adjusting linear parameters. Dubowsky (Dubowsky & DesForges, 1979) applied Model Reference Adaptive Control (MRAC) to a robot manipulator control using the local linearization technique. Craig, Hsu, and Sastry (1986) and Slotine and Li (1987) proposed an adaptive control method based on the Lyapunov function for a manipulator whose nonlinear characteristics were known in advance.

In previous studies of adaptive learning control using a neural network model, Barto (Barto, Sutton, & Anderson, 1983), Jordan (Jordan, 1988), and Psaltis (Psaltis, Sideris, & Yamamura, 1987) addressed the problem of how to obtain the error signal for a neural network controller. In supervised learning (Barto, 1989), the difference between the desired response and the actual response (i.e., plant performance error) cannot directly be used as the error for controller adaptation. The error for controller adaptation should not be the trajectory-error (i.e., plant performance error) but the command-error (i.e., plant input error). Thus, Jordan proposed *forward-inverse-modeling* (Jordan, 1988) and Albus (1975), Miller (Miller, Hewes, Glanz, & Kraft, 1990), Atkeson (Atkeson & Reinkensmeyer, 1988), Psaltis (Psaltis et al., 1987), Kuperstein (Kuperstein & Rubinstein, 1989) and Martinets (Martinets, Ritter, & Schulten, 1990) used *direct-inverse-modeling* to obtain command-error for forming the inverse dynamics model as a feedforward controller. In *reinforcement learning* (Barto et al., 1983), it is possible to improve plant performance over time by means of online learning methods in less structured situations. As an example of *reinforcement learning,* Barto proposed using ASE (associative search element) and ACE (adaptive critic element) techniques (Barto et al., 1983).

Kawato, Furukawa, and Suzuki (1987) proposed a learning method to acquire a *feedforward* controller,

Acknowledgments: The main part of this work was done while the authors were at the ATR Auditory and Visual Perception Research Laboratories. We would like to thank Dr. E. Yodogawa of the ATR Auditory and Visual Perception Research Laboratories for his continuing encouragement. This work was supported by a Human Frontier Science Project Grant to M.K.

A part of this work has been presented at the IEEE Conference on Decision and Control 1990.

Requests for reprints should be sent to Hiroaki Gomi, ATR Human Information Processing Research Laboratories, 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan.

which uses the output of a feedback controller as the error for training a neural network model. They called this learning method *feedback-error-learning*. Using this method, the neural network model for feedforward control acquires the inverse dynamics model of a controlled object. They successfully applied feedback-error-learning to trajectory-learning and force-control of a PUMA robot (Kawato, 1990a; Miyamoto, Kawato, Setoyana, & Suzuki, 1988), a kinematically redundant manipulator (Kano, Kawato, Uno, & Suzuki, 1990), and a manipulator driven by rubber actuators with dynamical redundancy (Katayama & Kawato, 1991a, 1991b, 1991c). This method was originally proposed as a model of voluntary movement learning in the cerebellum (Tsukahara & Kawato, 1982). However, no methods of learning for an adaptive *feedback* controller using feedback-error-learning had yet been examined.

In this paper, we propose learning schemes using feedback-error-learning for a neural network model applied to an adaptive nonlinear *feedback* controller. In these learning schemes, a conventional feedback controller (CFC) is provided both as an ordinary feedback controller to guarantee global asymptotic stability in a compact space and as an inverse reference model of the response of the controlled object. If a CFC is prepared as an inverse reference model of the response in Cartesian space, *impedance control* (Hogan, 1985) is derived after learning as shown in Section 2.1. In Section 3, we introduce simulated results of proposed learning schemes. In Section 4, we also point out the relationship of these learning schemes to the posture and locomotion adaptive control mechanisms in the cerebellum which receives proprioceptive feedback signals as the inputs. Finally, in the Appendixes, we discuss the convergence properties of the neural network models by applying the Lyapunov method to the averaged equations associated with the stochastic differential equations that describe the system dynamics.

2. ADAPTIVE NONLINEAR FEEDBACK CONTROLLER

In this section, we propose two adaptive learning control schemes using feedback-error-learning for neural network feedback controllers.

In the first learning scheme, the neural network feedback controller ultimately acquires an inverse dynamics model of the controlled object. Thus, we call this learning scheme Inverse Dynamics Model Learning (IDML). In the second learning scheme, the neural network model is trained to become a nonlinear regulator to compensate for the nonlinearity of the controlled object (except for the inertia term) through learning. Accordingly, this is called Nonlinear Regulator Learning (NRL). These two learning schemes are described in detail below. Convergence properties of these two learning schemes will be examined in the Appendixes using procedures similar to those used by Kawato (Kawato, 1990b) for adaptive *feedforward* control.

2.1. Inverse Dynamics Model Learning (IDML)

We explain the configuration and the behavior of this learning system using Figure 1. Subsequently, we show *learning impedance control* as an application of IDML in Cartesian space.

The components, a CFC, a neural network applied to an adaptive nonlinear feedback controller (NNFC), and a controlled object, are connected as shown in Figure 1. The CFC is used both as an ordinary feedback controller to guarantee global asymptotic stability during the learning period, and as a reference model for the responses of the controlled object. The sum of the output of the CFC and the external input for a controlled object is fed to the NNFC as the error signal. This corresponds to the inputs for parameter modification. The NNFC also receives $\theta, \dot{\theta}$, and $\ddot{\theta}$ as ordinary

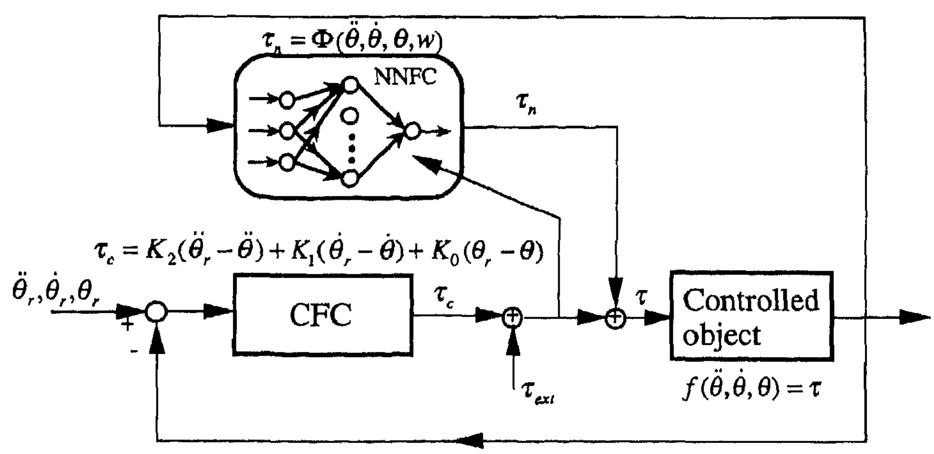

FIGURE 1. Block diagram of Inverse Dynamics Model Learning (IDML).

inputs. Here, θ, $\dot{\theta}$, and $\ddot{\theta}$ are the state vectors of the controlled object: its position, velocity, and acceleration. As the neural network acquires the inverse dynamics model of the controlled object through learning with sufficiently rich external inputs or sufficiently rich reference responses, the output responses of the controlled object are governed by the inverse reference model implemented as the CFC. Thus, in the absence of external inputs after learning, the actual responses finally coincide with the reference responses.

Following are the equations representing the dynamics of each component and this learning scheme. The controlled object dynamics f is expressed as:

$$f(\ddot{\theta}, \dot{\theta}, \theta) = \tau. \tag{1}$$

For simplicity, a linear controller is used as the CFC in the following explanation:

$$\tau_c = K_2(\ddot{\theta}_r - \ddot{\theta}) + K_1(\dot{\theta}_r - \dot{\theta}) + K_0(\theta_r - \theta). \tag{2}$$

The NNFC output is expressed as:

$$\tau_n = \Phi(\ddot{\theta}, \dot{\theta}, \theta, w). \tag{3}$$

Here, θ is the actual position vector, θ_r is the reference position vector, and w is the matrix set of synaptic weights (i.e., adaptive parameters) of the NNFC. The NNFC can be one of several types of neural network models in which the error of the output vector will decrease by changing the internal adaptive parameters [e.g., Multi-Layer Perceptron (MLP) (Irie & Miyake, 1988; Rumelhart & McClelland, 1986), Cerebellar Model Articulator Controller (CMAC) (Albus, 1975), associative content addressable memory (Atkeson & Reinkensmeyer, 1988), Memory Based Reasoning (MBR) (Stanfill & Waltz, 1986), or Radial Based Function (RBF) network (Poggio & Girosi, 1990)]. The optimum structure and the size of the neural network model depend on the target function f to be learnt. This problem has been frequently discussed in contexts such as functional approximation or the learning generalization problem, and is beyond the scope of this paper. Our assumption about the neural network model is that the nonlinear function of the controlled object, f, can be arbitrarily modeled closely by Φ with an appropriate w within a compact set. This approximation property has been validated for several neural network models (e.g., Irie & Miyake, 1988; Funahashi, 1989). For more satisfactory treatment of the functional approximation problem, readers are encouraged to refer to Sanner and Slotine (1992) in which the RBF network is used to discuss this problem.

The equation of the system is

$$\tau = \tau_n + \tau_c + \tau_{ext}, \tag{4}$$

where τ_{ext} is the external input to the controlled object. The difference between τ and τ_n is expressed as τ_{imag}.

$$\tau_{imag} = \tau - \tau_n = \tau_c + \tau_{ext}. \tag{5}$$

Equation 5 can be rewritten as the following equation using eqn 2.

$$K_2(\ddot{\theta} - \ddot{\theta}_r) + K_1(\dot{\theta} - \dot{\theta}_r) + K_0(\theta - \theta_r) = \tau_{ext} - \tau_{imag}. \tag{6}$$

The learning rule of the feedback-error-learning scheme is:

$$\frac{dw}{dt} = \eta\left(\frac{\partial\Phi}{\partial w}\right)^T(\tau_c + \tau_{ext}) = \eta\left(\frac{\partial\Phi}{\partial w}\right)^T\tau_{imag}, \tag{7}$$

where η is a positive-definite matrix which determines the learning rate. The convergence property of this leaning scheme is examined in Appendix A. After learning, the NNFC acquires an arbitrary close model of the inverse dynamics of the controlled object, and the response of the controlled object is governed by:

$$K_2\ddot{\xi} + K_1\dot{\xi} + K_0\xi \cong \tau_{ext}, \tag{8}$$

where $\xi = \theta - \theta_r$. That is to say, the tracking-error, $\theta_r - \theta$, converges to zero in accordance with the reference model represented by eqn (8) while $\tau_{ext} = 0$. When the external input, τ_{ext}, is served, the tracking-error is given by eqn (8).

The above describes the learning scheme in joint space. This can also be applied in Cartesian space. Using this learning scheme in Cartesian space allows for the *impedance control* proposed by Hogan (1985). To apply impedance control to a robot manipulator or some mechanical system, it is necessary to fully know the dynamics of the controlled object in advance. By using IDML scheme in Cartesian space, however, the controlled object dynamics will be obtained by learning. Thus, we call this learning scheme as applied to Cartesian space *learning impedance control.* Next, learning impedance control is explained briefly as an application of the IDML. The manipulator dynamics can be expressed as:

$$R(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) = \tau. \tag{9}$$

The CFC is represented by the following equation using the Cartesian coordinates

$$F_c = M\ddot{x}_e + B\dot{x}_e + Kx_e, \tag{10}$$

where $x_e = x_r - x$. Equation (10) defines the reference model of the manipulator responses. M, B, and K respectively represent the virtual mass, virtual viscosity, and virtual stiffness of the manipulator as derived by impedance control. The function Φ of the neural network is the same as in eqn (3). The total system is shown in Figure 2. Similarly to the joint-space scheme, if the NFC acquires an approximate model of the inverse dynamics of the controlled object by learning, the total input to the controlled object, τ, is given by the following equation.

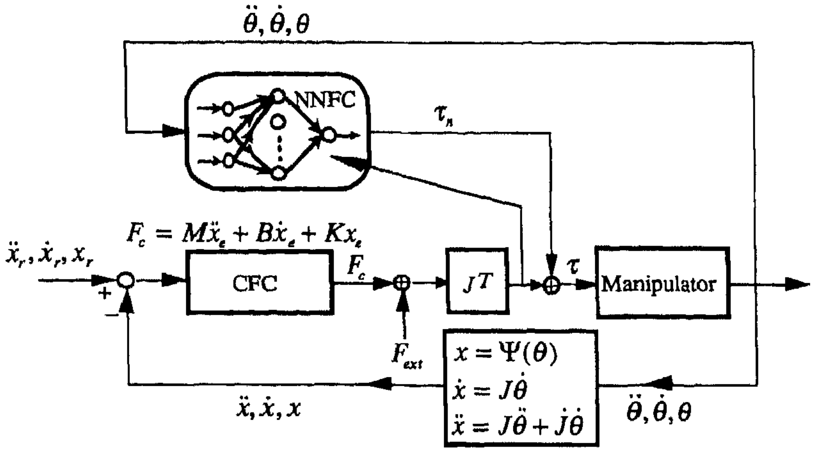

FIGURE 2. Block diagram of Learning Impedance Control.

$$\tau = \tau_n + J^T(F_c + F_{ext}) \cong R(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) + J^T(M(\ddot{x}_r - \ddot{x}) + B(\dot{x}_r - \dot{x}) + K(x_r - x) + F_{ext}). \quad (11)$$

Hence, the response of the end-point of the manipulator is governed by:

$$M(\ddot{x} - \ddot{x}_r) + B(\dot{x} - \dot{x}_r) + K(x - x_r) \cong F_{ext}. \quad (12)$$

Thus, this learning method derives impedance control for a manipulator that has unknown nonlinear characteristics. Moreover, we can change the virtual impedance of the manipulator simply by changing the parameters, M, B, and K, of the CFC without relearning, because the NNFC learns only the inverse dynamics model of the controlled object.

2.2. Nonlinear Regulator Learning (NRL)

In this learning scheme, the actual acceleration is not used as the input of the NNFC. Instead, we feed the reference trajectory (position, velocity, and acceleration) to the NNFC in order to acquire the feedforward controller and to obtain desired responses. We explain here the configuration of this learning scheme and the results of learning. In Appendix B, we will discuss the convergence property briefly.

The dynamics of the controlled object is represented here as:

$$R(\theta)\ddot{\theta} + N(\theta, \dot{\theta}) = \tau. \quad (13)$$

R is an inertia matrix that is nonlinear. N is another term of the controlled object dynamics (i.e., Corioris-force, Centrifugal-force, Viscosity, and Stiffness). As with the above method, IDML, and CFC serves the same two purposes (see Section 2.1.). We use only the output of the CFC, τ_c, as the error signal (i.e., adaptive modification input) for the NNFC as shown in Figure 3. This is expressed as:

$$\frac{dw}{dt} = \eta\left(\frac{\partial \Phi}{\partial w}\right)^T \tau_c. \quad (14)$$

In this learning scheme, the external input is considered to be absent during learning.

In this case, we feed the reference values, $\ddot{\theta}_r$, $\dot{\theta}_r$, θ_r, and the tracking-errors of the controlled object, $\theta_r - \theta$, $\dot{\theta}_r - \dot{\theta}$, to the NNFC as ordinary inputs. Hence, con-

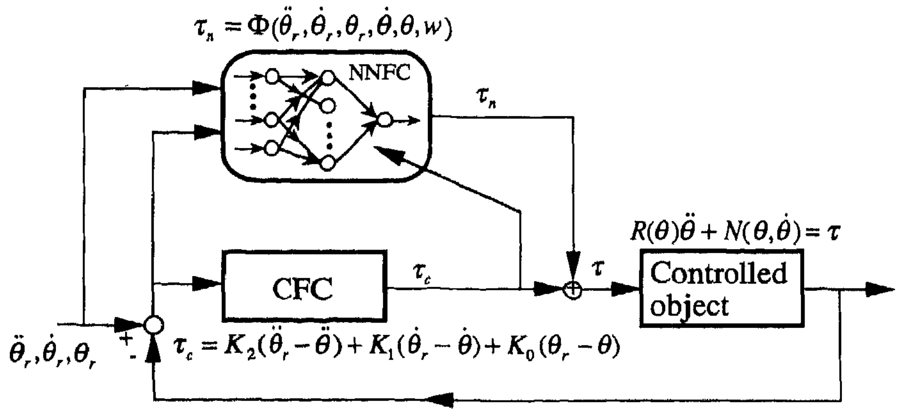

FIGURE 3. Block diagram of Nonlinear Regulator Learning (NRL).

sidering that θ, $\dot{\theta}$ is reproduced by using the reference values and the tracking-errors, the function of the NNFC is:

$$\tau_n = \Omega(\ddot{\theta}_r, \dot{\theta}_r, \theta_r, \theta_r - \theta, \dot{\theta}_r - \dot{\theta}, w) = \Phi(\ddot{\theta}_r, \dot{\theta}_r, \theta_r, \dot{\theta}, \theta, w). \quad (15)$$

The total dynamics is given by the following equation.

$$\tau = \tau_c + \tau_n \Leftrightarrow (R(\theta) + K_2)(\ddot{\theta}_r - \ddot{\theta}) + K_1(\dot{\theta}_r - \dot{\theta}) + K_0(\theta_r - \theta) + \Phi - N(\theta, \dot{\theta}) - R(\theta)\ddot{\theta}_r = 0. \quad (16)$$

If Φ is represented as:

$$\Phi \cong \Phi_d = N(\theta, \dot{\theta}) + R(\theta)\ddot{\theta}_r + R(\theta)K_2^{-1}(K_1(\dot{\theta}_r - \dot{\theta}) + K_0(\theta_r - \theta)), \quad (17)$$

Equation (16) can be expressed as:

$$(R(\theta) + K_2)(\ddot{\theta}_r - \ddot{\theta}) + K_1(\dot{\theta}_r - \dot{\theta}) + K_0(\theta_r - \theta) + R(\theta)K_2^{-1}(K_1(\dot{\theta}_r - \dot{\theta}) + K_0(\theta_r - \theta)) \cong 0. \quad (18)$$

This gives:

$$(I + R(\theta)K_2^{-1})(K_2(\ddot{\theta}_r - \ddot{\theta}) + K_1(\dot{\theta}_r - \dot{\theta}) + K_0(\theta_r - \theta) \cong 0. \quad (19)$$

Consequently, this equation yields the tracking-error responses without an external input to the controlled objects on the condition, $I + R(\theta)K_2^{-1} \neq 0$, as:

$$K_2\ddot{\xi} + K_1\dot{\xi} + K_0\xi \cong 0. \quad (20)$$

Here $\xi = \theta - \theta_r$.

Roughly speaking, to obtain the desired response during tracking-error convergence movement without external inputs [i.e., eqn (20)] by compensating for the nonlinear object dynamics, the neural network model is trained to become a nonlinear regulator expressed in eqn (17). As with the IDML method, we can also apply this learning scheme in Cartesian space. However, perfect impedance control cannot be derived by using NRL because the NNFC does not have the feedback acceleration signal, $\ddot{\theta}$. Therefore, it is impossible to make the inertia term of virtual impedance smaller than the inertia term of inherent impedance of the controlled object by using NRL in Cartesian space.

3. SIMULATION RESULTS

In this section, we show the simulation results to the learning schemes described in the previous section applied to particular control problems. In the first experiment, an inverted pendulum is used as the controlled object for IDML and NRL in joint space. In the second experiment, a two-link manipulator is used as the controlled object for IDML in Cartesian space (i.e., learning impedance control). All simulations used a fourth order, fixed-step-size Runge-Kutta method (with $\Delta t = 2 \times 10^{-5}$ (s) for inverted pendulum simulations, and $\Delta t = 2 \times 10^{-4}$ (s) for manipulator simulations).

3.1. Inverted Pendulum Controlled by IDML

The dynamics of the controlled object, an inverted pendulum, is expressed as:

$$-(J + ml^2)\ddot{\theta} - \left(\frac{2}{1 + \exp(-a\dot{\theta})} - 1\right) + mgl \sin\theta = ml\tau\cos\theta. \quad (21)$$

where

m: pendulum mass 0.0411 [kg],
l: pendulum length 1.26 [m],
J: pendulum inertia 0.0647 [kg·m/s^2],
θ: pendulum angle [rad],
τ: input acceleration for the cart [m/s^2],
g: gravity acceleration 9.806 [m/s^2].

The second term on the left side of eqn (21) corresponds to Coulomb friction. This is a typical regulator problem because the reference values $\ddot{\theta}_r$, $\dot{\theta}_r$, θ_r are always zero.

The following nonlinear functions, p_i, are used as the input to the two-layered neural network in the NNFC. This is because we can show the convergence of the adaptive parameters to the ideal values, and the learning problem can be made considerably easier by using known nonlinearity, as shown by Khosla & Kanade (1985), and others (Atkeson & Reinkensmeyer, 1988; Craig et al., 1986; Kawato et al., 1987; Miyamoto et al., 1988; Slotine & Li, 1987).

$$p_0 = \tan\theta$$
$$p_1 = (2/(1 + \exp(-a\dot{\theta})) - 1)/\cos\theta$$
$$p_2 = \ddot{\theta}/\cos\theta. \quad (22)$$

The output of the NNFC, τ_n, is

$$\tau_n = \sum_{k=0}^{2} w_k p_k. \quad (23)$$

The ideal values of the weights in IDML are represented as follows for the controlled object expressed as eqn (21).

$$\hat{w}_0 = g, \quad \hat{w}_1 = -\frac{1}{ml}, \quad \hat{w}_2 = -\frac{J + ml^2}{ml}. \quad (24)$$

Using these known nonlinearities and linear parameter adaptation, the desired function with ideal weights for the adaptive controller are exactly obtained by the *feedback-error-learning* rule.

Figure 4 shows the time courses of the weights during

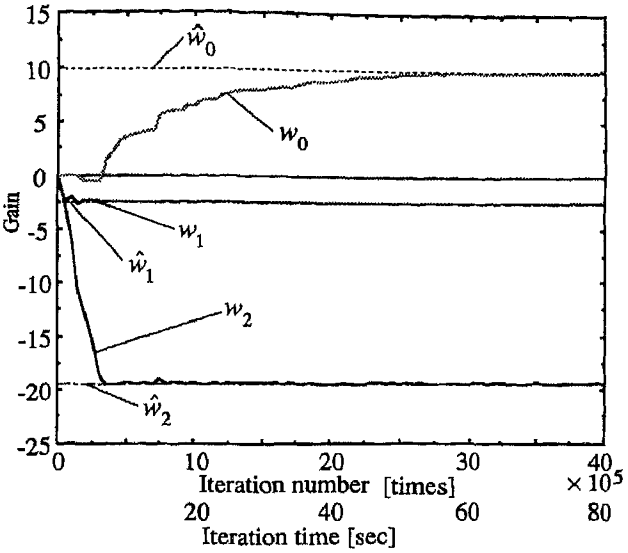

FIGURE 4. Time courses of weights (adaptive parameters) during IDML.

IDML. We used the Ornstein–Uhlenbeck Process, which is a strongly mixing process (i.e., past and future becomes asymptotically independent), as the external input, τ_{ext}, in this learning period. During the leaning period, the feedback gains are $K_2 = 1$, $K_1 = 3.5$, $K_0 = 20$, and $\hat{w}_2$, $\hat{w}_1$, $\hat{w}_0$ are the ideal values for the actual values, w_2, w_1, w_0. Each actual value gradually approached the ideal value. Figure 5 shows the moving average of the square of the NNFC output and the error for the NNFC during IDML. The NNFC output increased to compensate for the controlled object dynamics. This adaptation decreased the error for the NNFC.

The responses of the pendulum to each condition are compared in Figure 6. The response after learning

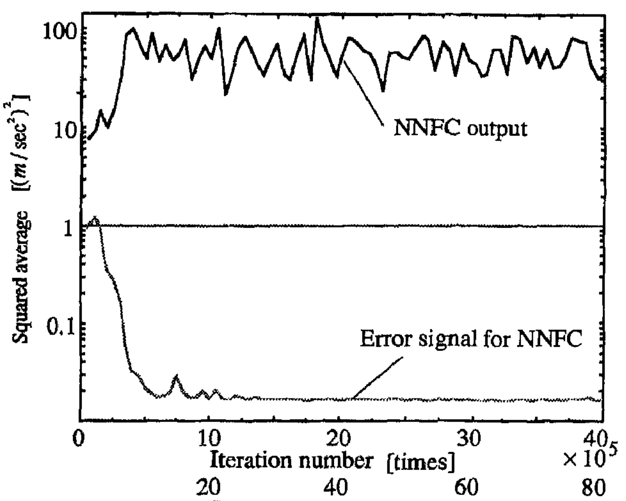

FIGURE 5. Moving averages of the square of the NNFC output and the error for NNFC during IDML.

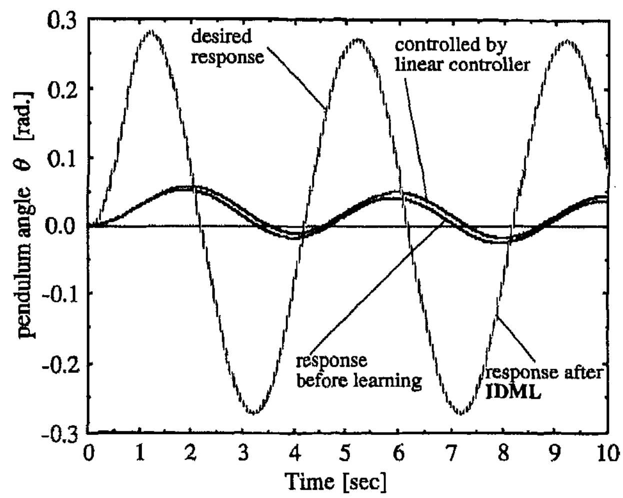

FIGURE 6. The responses of the controlled object for external input. $\tau_{ext} = -5\sin(\pi t/2)$ [m/s²].

corresponded perfectly to the desired response governed by the reference model:

$$K_2(\ddot{\theta} - \ddot{\theta}_r) + K_1(\dot{\theta} - \dot{\theta}_r) + K_0(\theta - \theta_r) = \tau_{ext}. \quad (25)$$

(In this case, $\ddot{\theta}_r = \dot{\theta}_r = \theta_r = 0$, $K_2 = 1$, $K_1 = 3.5$, $K_0 = 20$.)

To compare with a conventional method, we show the response controlled by the linear controller which was designed to obtain the desired response for a linearized controlled object without the friction term:

$$-(J + ml^2)\ddot{\theta} + mgl\theta = ml\tau. \quad (26)$$

The response driven only by the linear controller did not correspond to the desired response because of the nonlinearity of the controlled object and the unexpected factor in the controller design (i.e., the friction term of the actual object dynamics).

3.2. Inverted Pendulum Controlled by NRL

We show here a result of the second learning scheme, NRL, using the same controlled object, an inverted pendulum. by using the prepared nonlinearities, we have already confirmed that the actual weights in this learning scheme approached the ideal weights. Accordingly, the following result was derived by using a 3-layered neural network (2 input units, 5 hidden units, 1 output unit) with direct connections from the input units to the output unit to confirm the ability of this learning scheme.

Figure 7 shows the desired response, the response using the linear controller, the response before learning, and the response after learning. The linear controller was designed to perform the desired response for the controlled object expressed in eqn (26). However, we did not obtain the desired response using a linear con-

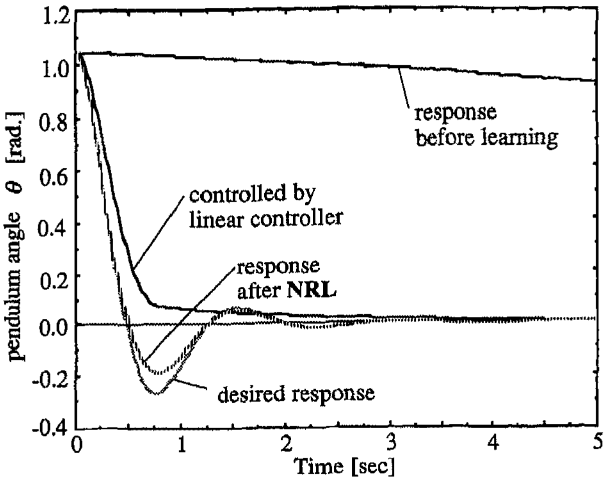

FIGURE 7. The responses of the controlled object from initial position of 60°. The parameters in feedback controller, K_2, K_1, K_0, are 1.0, 3.5, 20.0, respectively.

troller because of the nonlinearity and friction of the actual controlled object. The actual response after learning approached the desired response because the nonlinear adaptive controller compensated for the nonlinearity of the controlled object through learning.

3.3. Learning Impedance Control of a 2-link Manipulator

In this section, we show some results of learning impedance control using a 2-link manipulator in the horizontal plane. A 3-layered neural network model with 9 input units, 13 hidden units, and 2 output units was used in this simulation. To demonstrate the efficiency of learning impedance control, we compared the responses of the end-point of the manipulator to the external force input in each condition (before learning, after 1000-s learning, and the ideal response), as shown in Figure 8. The virtual impedance which is decoupled along the x and y axes (i.e., nondiagonal components of the inertia-, viscosity-, stiffness-matrix are zero), is required for the manipulator. For a step external force input along the x or y axis, the response of the ideal system which has such decoupled impedance, traces a straight path along each axis as shown in Figure 8. When an ideal inverse model is supplied to the NNFC, the responses will be ideal, even though the inherent dynamics of the controlled object's components interference each other, such as a 2-link manipulator. As shown in Figure 8, the responses after learning were closer to the ideal response than before learning. Thus, we can conclude that the NNFC acquired an approximate model of the inverse dynamics model in the work space which was explored during learning.

Next, we examined the responses in the contact task to the wall, which was simulated using a strong spring. The parameters of the CFC are shown below for each condition.

$$F_c = -[M\ddot{x} + B\dot{x} + K(x - x_r)] \quad x = [x, y]^T.$$

Before learning

$$M = \mathrm{diag}[0.1, 0.1]\ [\mathrm{kg}]$$
$$B = \mathrm{diag}[7.0, 7.0]\ [\mathrm{N/(m/s)}]$$
$$K = \mathrm{diag}[100, 100]\ [\mathrm{N/m}],$$

After learning

$$M = \mathrm{diag}[0.1, 0.1]\ [\mathrm{kg}]$$
$$B = \mathrm{diag}[7.0, 7.0]\ [\mathrm{N/(m/s)}]\ (\text{noncontact phase})$$
$$B = \mathrm{diag}[700, 7.0]\ [\mathrm{N/(m/s)}]\ (\text{contact phase})$$
$$K = \mathrm{diag}[100, 100]\ [\mathrm{N/m}].$$

Before learning, the actual trajectory differed from the reference trajectory in free movement, and the performance of the contact task was much poorer, as shown in Figures 9 and 10. After learning, the actual trajectory in free movement was much closer to the reference trajectory as shown in Figure 11. The impact force at the time of collision was much lower than before learning as shown in Figure 12, because the virtual impedance of the manipulator was lowered. Moreover, the contact task was so smooth that the external force was proportional to the magnitude of the difference between the actual trajectory and the reference trajectory as shown in Figure 12. This is because the NNFC acquired approximate model of the inverse dynamics, and the virtual impedance settled in the CFC was changed to allow a stable contact in this phase.

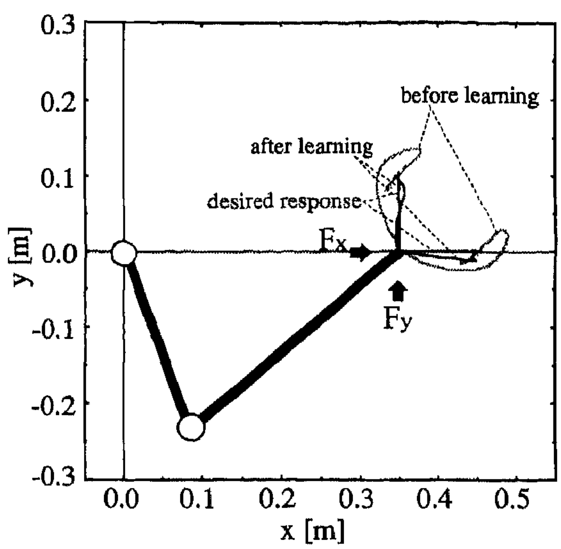

FIGURE 8. Improved responses for the external force inputs at the end effector. The parameters in feedback controller, M, B, K, are 0.1, 2.0, 20.0, respectively.

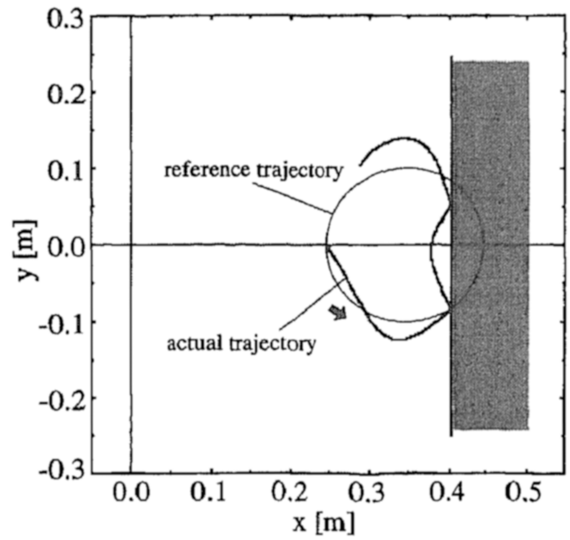

FIGURE 9. Trajectory during the contact task before learning.

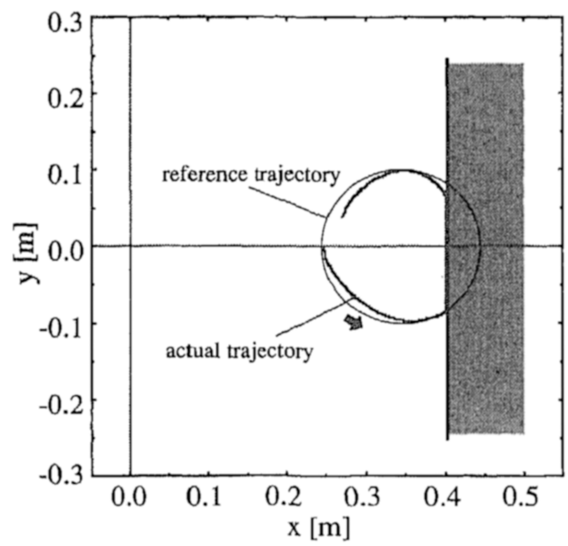

FIGURE 11. Trajectory during the contact task after learning.

4. DISCUSSION

4.1. Differences Between the Two Learning Schemes, IDML and NRL

Comparing the results of IDML and NRL, the same linear autonomous responses ($\tau_{ext} = 0$) can be obtained after learning as shown in eqn (8) and eqn (20). However, the two methods give different responses for the same external input (i.e., constrained movement). As shown by eqn (19), NRL cannot provide the motor command to satisfy a desired response for the external input, τ_{ext}. This is because the dynamics of the controlled object cannot be fully compensated for using an adaptive feedback controller, whose inputs are only actual position and velocity. When the desired response to the continuous external input should be satisfied in a particular control task, it is necessary to use IDML, and when only the convergence response without external input (i.e., the autonomous response) should be improved, NRL is preferable to use because it is not necessary to measure the external input, and we do not need the acceleration input to the adaptive controller, NNFC. Therefore, the choice of learning scheme depends on the object to be controlled.

4.2. Relation to Previous Work

4.2.1. *Adaptive Control Methods.*. As we noted in Section 1, previous adaptive control methods mainly use the local linearization technique (Dubowsky & DesForges, 1979) or a priori knowledge of the nonlinearity of the controlled object to regulate linear parameters (Craig et al., 1986; Slotine & Li, 1991). On the other hand, some nonlinear adaptive controllers, such as artificial neural networks, can be directly utilized for nonlinear problems (Sanner & Slotine, 1992). A sufficiently large capacity of a neural network and a suf-

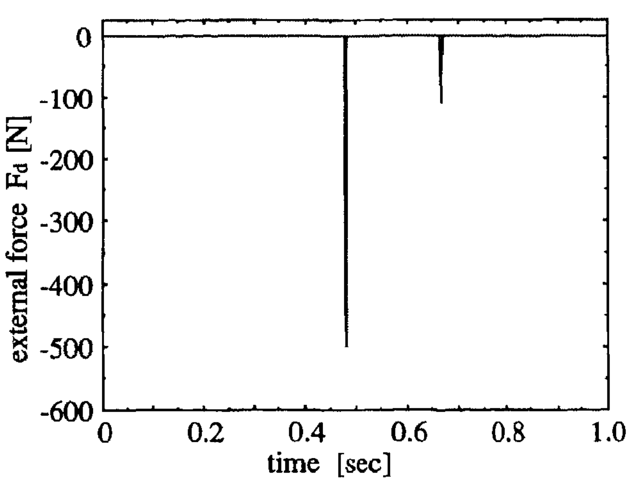

FIGURE 10. External force at the end effector during the contact task before learning.

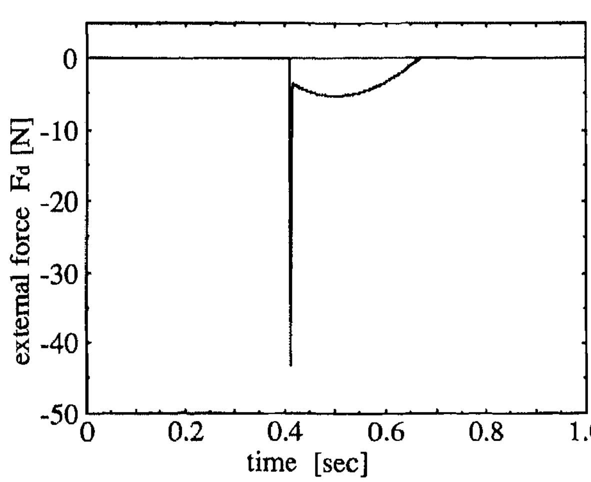

FIGURE 12. External force at the end effector during the contact task after learning.

ficiently powerful training method in the work space make it unnecessary to know the exact nonlinearity of the controlled object in advance. In practice, however, it is difficult to achieve these ideal conditions. Thus, it is preferable to partially use neural network controllers for controlled objects that have unknown characteristics.

4.2.2. *Neural Network Controllers.*. Robot control has been studied using neural networks which employ *direct inverse modeling* (Barto, 1989; Jordan, 1988). This research can be categorized into two areas: creating an inverse kinematics model (Kuperstein & Rubinstein, 1989; Martinets et al., 1990) and creating an inverse dynamics model (Miller et al., 1990; Psaltis et al., 1987).

As we pointed out in the introduction, the fundamental problem of training a nonlinear adaptive controller such as a neural network is how to obtain the error signal for the controller. The forward-inverse-modeling method (Jordan, 1988) uses back propagation (Rumelhart & McClelland, 1986) through a forward dynamics model (direct dynamics model) to obtain the input-error (command-error) from the output-error (trajectory-error). Mathematically, this corresponds to a steepest-descent method. On the other hand, it has been pointed out (Kawato, 1990b) that feedback-error-learning corresponds to a Newton-like method. These two methods, steepest descent and the Newton-like method, are well known numerical solutions for nonlinear optimization problems. Thus, we can employ not only feedback-error-learning but also back propagation through a forward model in the proposed schemes, IDML and NRL, as shown in Figure 13. In this case, the reference model of tracking-error convergence is prepared separately from the CFC as shown. In the IDML case, the desired acceleration, $\ddot{\theta}_d$, is calculated by the reference model using the reference response, θ_r, the actual response, θ, and the external input, τ_{ext}, such as:

$$\ddot{\theta}_d = K_2^{-1}(\tau_{ext} + K_1(\dot{\theta}_r - \dot{\theta}) + K_0(\theta_r - \theta)) + \ddot{\theta}_r.$$

Narendra and Parthasarathy (1990) examined in detail neural network controllers that use another kind of reference model and error-back propagation through a forward dynamics model.

Sanner and Slotine showed a learning scheme (Sanner & Slotine, 1992) similar to the NRL scheme proposed above and before Gomi and Kawato (1990). They examined the convergence of their learning schemes based on the adaptive control method proposed by Slotine and Li (1987) and the functional approximation by the Fourier series to translate nonlinear function to linear parameters with basis functions. They also discussed how to actually combine neural network learning with conventional sliding control (Slotine & Li, 1991). Their scheme has the merit of not using the current acceleration of the controlled object. However the applications are restricted to trajectory tracking control. If external inputs are provided, the adaptive mechanism changes the adaptive parameters to improve the trajectory tracking performance, which is lowered by the external inputs. Thus, the adaptive parameters will change continuously because of the varying external inputs, and parameter convergence cannot be obtained. When the changing rate of the external inputs is faster

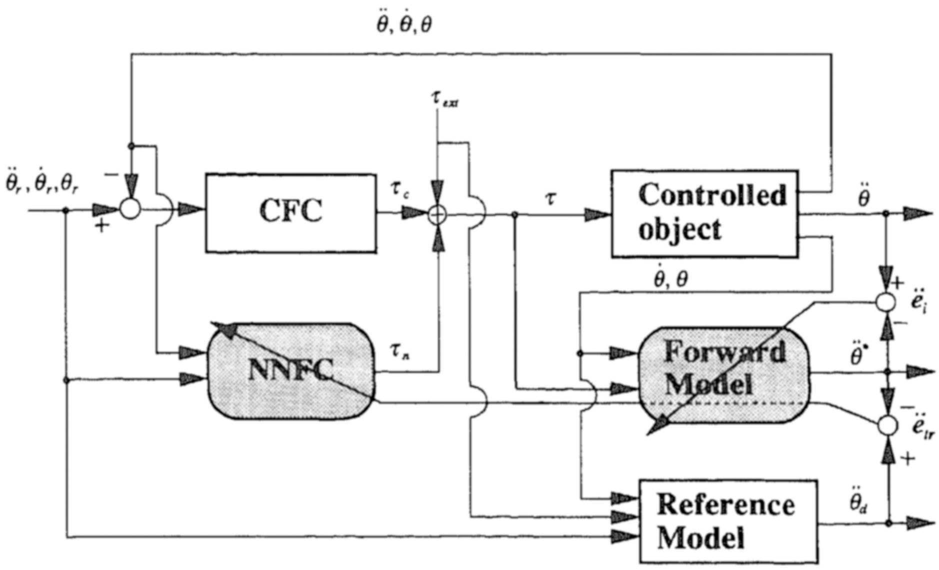

FIGURE 13. IDML and NRL scheme by using back propagation through a forward model. The error for forming the forward model is the difference between the actual output of the controlled object and the output of the forward model, e_i. The error for training the NNFC is produced from the tracking-error, e_{tr}, by using back propagation through the forward model of a controlled object (i.e., partial derivative of the output error with respect to the input of the forward model).

than the adjusting rate of the parameters (which is relatively slow in usual neural network learning), accurate trajectory tracking cannot be obtained. Moreover, when the controlled object is modeled as a second order mechanical model, it might not be possible to directly apply their scheme to regulation control in which the reference acceleration is always zero, such as positional regulation control or velocity regulation control. For example, if their scheme is applied to control an inverted pendulum without the boundary of the parameter adaptation, the adaptive parameters will diverge because of unlimited positional regulation. Additionally, even if adaptation is stopped after the trajectory-error and the parameters both converge, the desired autonomous response cannot be supplied, because the design of their adaptive control is focused only on the trajectory tracking control rather than on obtaining the desired response during the tracking-error convergence. In contrast to their scheme, although the current acceleration is required, the adaptation mechanisms in IDML and NRL work not only for tracking control but also for regulation control as shown in the inverted pendulum simulations above, and the desired autonomous response can be obtained after learning.

4.3. Adaptive Motor Control in the Central Nervous System (CNS)

Originally, this research was launched to reveal the computational schemes of motor learning in the central nervous system (CNS). Hence, we discuss here the relationships between the learning schemes proposed above and the actual animal motor learning ability.

Humans can stand upright stably and walk smoothly, because various sensors observe the present state, and the CNS controls posture with a real-time feedback loop. This ability is gradually learnt during growth. It has been pointed out that proprioceptors such as the muscle spindles and the tendon organs of Golgi, the vestibulo organ and vision all play a major part as sensors for observing the present state. The proprioceptors observe the position, velocity, and force of the limbs and the body trunk. The vestibulo organs observe the head velocity and acceleration, and the visual sensor also detects the head velocity. Even if one sensor is seriously damaged, the other sensors can compensate in order to maintain the upright posture. It is known that there are some tracts descending in the spinal cord from the brain stem and cerebellum to the limb and body trunk muscles which serve to harmoniously control posture and locomotion (Carew, 1985).

Ito (1984) has revealed physiologically that the cerebellum plays an important role in adaptation for the vestibulo-ocular reflex. There are four main parts of the cerebellum: the flocculus, the vermis, the hemisphere intermediate part, and the hemisphere lateral part. Fujita (1982) gave a hypothesis for the adaptive mechanism regulating the vestibulo-ocular reflex in the flocculus, and Kawato et al. (1987) gave a hypothesis for the adaptive mechanism regulating voluntary movement in the hemisphere lateral part. Nashner found that the adaptation of posture control is severely impaired in patients with cerebellar disease (Nashner, 1976). Thus, we assume that the cerebellum also plays an important role in the adaptation of posture control.

The inputs and outputs of these four parts have already been investigated in anatomical studies (Ghez & Fahn, 1985). The principal afferent inputs of the vermis in the cerebellum are from the vestibular labyrinth, the proximal body parts, and the visual organs. The principal outputs of the vermis are to the medial brain stem and the axial regions of the motor cortex. The function of the vermis is thought to be axial and proximal motor control. The hemisphere intermediate part receives the afferent information from the spinal cord, and sends the output to the red nucleus and the distal region of the motor cortex. The function of the hemisphere intermediate part is thought to be distal motor control.

These physiological and anatomical observations support the hypotheses that the vermis and the hemisphere intermediate part have an adaptive mechanism for posture control and locomotion control. In addition to the previous two models by Fujita and Kawato, we propose two other models of adaptive control in the cerebellum for posture and locomotion control, as shown in Figure 14. These adaptive mechanisms are based on the feedback-error-learning mechanism which has some similarities with the Adaptive Vector Integration to Endpoint (VITE) model studied by Gaudiano and Grossberg (1991). These two adaptive models of the vermis and the hemisphere intermediate part correspond to the learning scheme that we proposed in this paper. The details of the physiological aspects for these models have been examined in Gomi and Kawato (1992) and Kawato and Gomi (1992). The physiological evidences for creating these precise models are not yet adequate, but these models might play an important role in revealing the computational schemes for motor learning by the CNS.

5. CONCLUSIONS

We have proposed new learning schemes using feedback-error-learning for a neural network model applied to adaptive nonlinear feedback control. After the neural network has compensated perfectly or partially for the nonlinearity of the controlled object through learning, the responses of the controlled object follow the desired responses supplied by inverse reference model implemented in the conventional feedback controller. Each learning scheme does not require perfect knowledge of the nonlinearity of a controlled object in advance. The learning schemes proposed here can also be used together with previous adaptive control methods to use

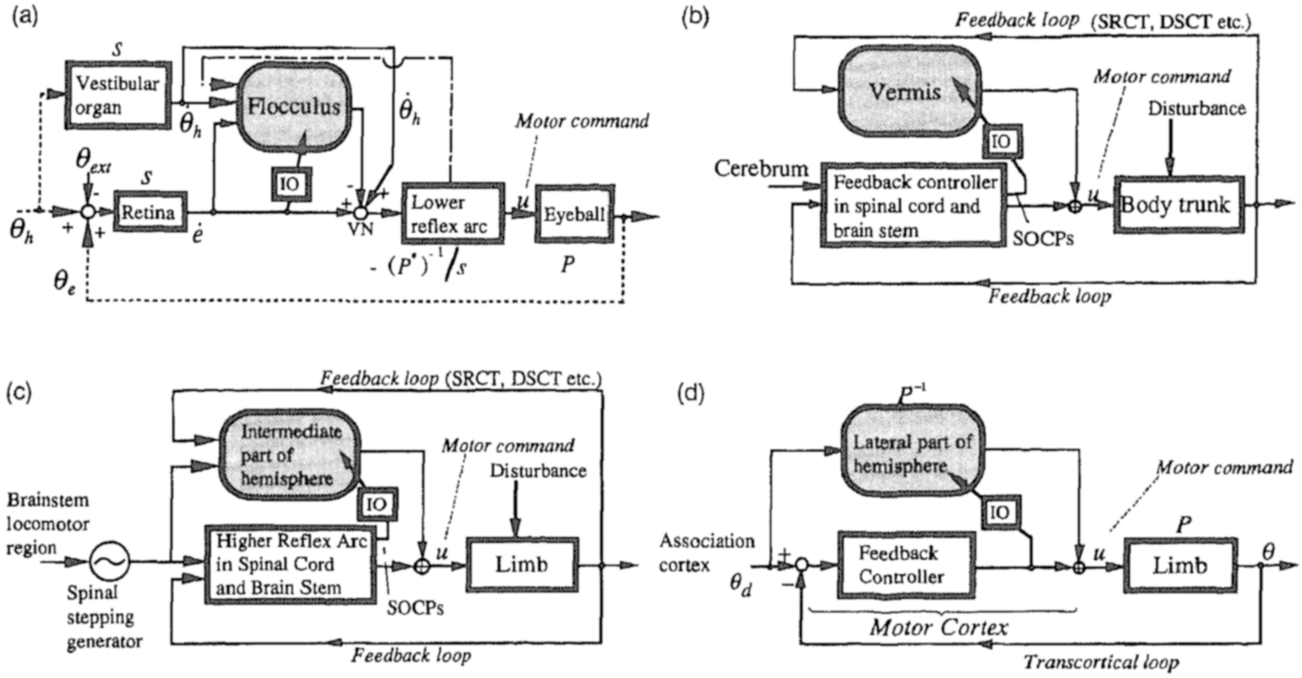

FIGURE 14. Schematic block diagrams of the adaptive motor control in each region of cerebellum. (a) Adaptive modification of vestibulo-ocular reflex and optokinetic response; (b) adaptive control for posture; (c) adaptive control for locomotion; (d) learning control for voluntary movement. IO: inferior olivary nucleus, VN: vestibular nuclei, SRCT: spino-reticulo-cerebellar tracts, DSCT: dorsal spino-cerebellar tracts, SOCPs: spino-olivo-cerebellar paths. θ_h: head position, θ_e: eye position, θ_{ext}: external visual world position, $\dot{e}$: retinal error velocity, s: derivative operator, P: controlled object dynamics.

a priori knowledge. Therefore, these learning schemes can be used to control many kinds of objects, such as chemical plants, mechanical systems, robots, etc.

REFERENCES

Albus, J. S. (1975). A new approach to manipulator control: The cerebellar model articulation controller. *Transactions of The American Society of Mechanical Engineering*, **97**, 220–227.

Atkeson, C. G., & Reinkensmeyer, D. J. (1988). Using associative content-addressable memories to control robots. *Proceedings IEEE Conference on Decision and Control*, 792–797.

Barto, A. G. (1989). Connectionist learning for control. In T. Miller III, R. S. Sutton, & P. J. Werbos (Eds.), *Neural networks for control* (pp. 5–58). Cambridge, MA: The MIT Press.

Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transaction on System Man and Cybernetics*, **13**, 834–846.

Carew, T. J. (1985). Posture and locomotion. In E. R. Kandel & J. H. Schwartz (Eds.), *Section* 37 *principles of neural science* (pp. 478–486). New York: Elsevier.

Craig, J. J., Hsu, P., & Sastry, S. S. (1986). Adaptive control of mechanical manipulators. *IEEE International Conference on Robotics and Automation*, 190–195. San Francisco.

Dubowsky, S., & DesForges, D. T. (1979). The application of model referenced adaptive control to robotic manipulators. *Journal of Dynamic System, Measurement and Control*, **101**, 193–200.

Fujita, M. (1982). Simulation of adaptive modification of the vestibulo-ocular reflex with an adaptive filter model of the cerebellum. *Biological Cybernetics*, **45**, 207–214.

Funahashi, K. (1989). On the approximate realization of continuous mapping by neural network. *Neural Networks*, **2**, 183–192.

Gaudiano, P., & Grossberg, S. (1991). Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. *Neural Networks*, **4**, 147–183.

Geman, S. (1979). Some averaging and stability results for random differential equations. *SIAM Journal of Applied Mathematics*, **36**(1), 86–105.

Ghez, C., & Fahn, S. (1985). The cerebellum. In E. R. Kandel & J. H. Schwartz (Eds.), *Section* 39 *principles of neural science* (pp. 502–521). New York: Elsevier.

Gomi, H., & Kawato, M. (1990). Learning control for a closed loop system using feedback-error-learning. *Proceedings IEEE Conference on Decision and Control*, 3289–3294. Hawaii.

Gomi, H., & Kawato, M. (1992). Adaptive feedback control models of the vestibulocerebellum and spinocerebellum. *Biological Cybernetics*, **68**(2), 105–114.

Hogan, N. (1985). Impedance control: An approach to manipulation: Part I—Theory, Part II—Implementation, Part III—Applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, **107**, 1–24.

Irie, B., & Miyake, S. (1988). Capabilities of three-layered perceptrons. *Proceedings International Conference on Neural Networks*, **1**, 641–648. San Diego.

Ito, K. (1953). *Probability theory* (p. 234). Tokyo: Iwanami.

Ito, M. (1984). *The cerebellum and neural control*. New York: Raven Press.

Jordan, M. I. (1988). *Supervised learning and systems with excess degree of freedom* (COINS Tech. Rep. No. 88–27). University of Massachusetts at Amherst.

Kano, M., Kawato, M., Uno, Y., & Suzuki, R. (1990). Learning trajectory control of a redundant arm by feedback-error-learning. *IEICE Technical Report*, **89**(463), 1–6.

Katayama, M., & Kawato, M. (1991a). Learning trajectory and force control of an artificial muscle arm by parallel-hierarchical neural network model. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in neural information processing systems*, Vol. 3 (pp. 436–442). San Mateo, CA: Morgan Kaufmann.

Katayama, M., & Kawato, M. (1991b). A parallel-hierarchical neural network model for motor control of musculo-skeletal system. *System & Computers in Japan*, **22**, 95–105.

Katayama, M., & Kawato, M. (1991c). Virtual trajectory and stiffness ellipse during force-trajectory control using a parallel-hierarchical neural network model. *Proceedings of Fifth International Conference on Advanced Robotics*, 1187–1194. Pisa, Italy.

Kawato, M. (1990a). Computational schemes and neural network models for formation and control of multijoint arm trajectory. In T. Miller III, R. S. Sutton, & P. J. Werbos (Eds.), *Neural networks for control* (pp. 192–228). Cambridge, MA: The MIT Press.

Kawato, M. (1990b). The feedback-error-learning neural network for supervised motor learning. In R. Eckmiller (Ed.), *Advanced neural computers* (pp. 365–372). Amsterdam: Elsevier.

Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural network model for control and learning of voluntary movement. *Biological Cybernetics*, **57**, 169–185.

Kawato, M., & Gomi, H. (1992). A computational model of four regions of the cerebellum based on feedback-error learning. *Biological Cybernetics*, **68**(2), 95–103.

Khosla, P. K., & Kanade, T. (1985). Parameter identification of robot dynamics. *Proceedings IEEE Conference Decision and Control*, 1754–1760. Ft. Lauderdale, FL.

Kuperstein, M., & Rubinstein, J. (1989). Implementation of an adaptive neural controller for sensory-motor coordination. *IEEE Control Systems Magazine*, **8**(2), 25–30.

Martinets, T. M., Ritter, H. J., & Schulten, K. J. (1990). Three-dimensional neural net for learning visuomotor coordination of a robot arm. *IEEE Transactions on Neural Networks*, **1**(1), 131–136.

Miller, W. T., III, Hewes, R. P., Glanz, F. H., & Kraft, L. G., III (1990). Real-time dynamic control of an industrial manipulator using a neural-network-based learning controller. *IEEE Transactions on Robotics & Automation*, **6**(1), 1–9.

Miyamoto, H., Kawato, M., Setoyana, T., & Suzuki, R. (1988). Feedback-error-learning neural network for trajectory control of a robotic manipulator. *Neural Networks*, **1**, 251–265.

Narendra, K. S., & Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, **1**(1), 4–27.

Nashner, L. M. (1976). Adapting reflexes controlling the human posture. *Experimental Brain Research*, **26**, 59–72.

Poggio, T., & Girosi, F. (1990). *Extensions of a theory of networks for approximation and learning: dimensionality reduction and clustering* (No. 1167). MIT A.I. Memo.

Psaltis, D., Sideris, A., & Yamamura, A. (1987). Neural controllers. *IEEE International Conference Neural Networks*, **4**, 551–557.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing*. Cambridge, MA: The MIT Press.

Sanner, R. M., & Slotine, J. J. E. (1992). Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6), 837–863.

Slotine, J. E., & Li, W. (1987). On the adaptive control of robot manipulators. *International Journal Robotics Research*, **6**(3).

Slotine, J. J. E., & Li, W. (1991). *Applied nonlinear control*. New Jersey: Prentice-Hall.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the Association for Computing Machinery*, **29**, 1213–1228.

Tsukahara, N., & Kawato, M. (1982). Dynamic and plastic properties of the brain stem neuronal networks as the possible neuronal basis of learning and memory. In S. Amari & M. A. Arbib (Eds.), *Competition and cooperation in neural nets*, (pp. 430–441). New York: Springer-Verlag.

APPENDIX A

Stochastic space is used in the discussion of the convergence of the NNFC below, because the nonlinearity of the controlled object is not specified [compare with Slotine and Li (1987) in which the nonlinearity of the object dynamics is perfectly known]. Geman's Theorem (1979) and Lyapunov's second method are used under the following three assumptions.

ASSUMPTION 1. *The learning rate, η, is very small and positive-definite.*

ASSUMPTION 2. *The inputs, τ_{ext} and $\ddot{\theta}_r$, $\dot{\theta}_r$, θ_r, are strongly mixing and strongly stationary stochastic processes.*

ASSUMPTION 3. *The CFC is designed to guarantee the asymptotic convergence of θ in a compact set during learning when τ_{ext} equal zero and θ_r is constant.*

Using the above assumptions, the system dynamics can be represented as the following random differential equations instead of the deterministic equations, eqn (4) and eqn (7).

$$\begin{aligned} f(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega)) - \Phi(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), w(t,\omega)) \\ = K_2(\ddot{\theta}_r(t,\omega) - \ddot{\theta}(t,\omega)) + K_1(\dot{\theta}_r(t,\omega) - \dot{\theta}(t,\omega)) \\ + K_0(\theta_r(t,\omega) - \theta(t,\omega)) + \tau_{ext}(t,\omega), \end{aligned} \quad (A.1)$$

$$\begin{aligned} \frac{dw(t,\omega)}{dt} = \eta\left(\frac{\partial\Phi(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), w(t,\omega))}{\partial w}\right)^T \\ \times \{\tau_c(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), \ddot{\theta}_r(t,\omega), \\ \dot{\theta}_r(t,\omega), \theta_r(t,\omega)) + \tau_{ext}(t,\omega)\}. \end{aligned} \quad (A.2)$$

Here, ω is a sample point in probability space. The solution of the following averaged equation is a good approximation to the solution of eqn (A.2) for small η, as proven by Geman's theorem (1979).

$$\begin{aligned} \frac{dM}{dt} = \eta E\Bigg[\left(\frac{\partial\Phi(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), w(t,\omega))}{\partial w}\right)^T \\ \times \{\tau_c(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), \ddot{\theta}_r(t,\omega), \\ \dot{\theta}_r(t,\omega)\theta_r(t,\omega)) + \tau_{ext}(t,\omega)\}\Bigg]_{w=M}. \end{aligned} \quad (A.3)$$

Thus, we consider the following function V as a possible Lyapunov function for the averaged eqn (A.3) by using the following function L:

$$\begin{aligned} L = \tfrac{1}{2}\{\tau_c(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), \ddot{\theta}_r(t,\omega), \dot{\theta}_r(t,\omega), \theta_r(t,\omega)) \\ + \tau_{ext}(t,\omega)\}^T \times \{\tau_c(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), \\ \ddot{\theta}_r(t,\omega), \dot{\theta}_r(t,\omega), \theta_r(t,\omega)) + \tau_{ext}(t,\omega)\} \end{aligned} \quad (A.4)$$

$$\begin{aligned} V(M,t) = E[L(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), \\ \ddot{\theta}_r(t,\omega), \dot{\theta}_r(t,\omega), \theta_r(t,\omega), \tau_{ext}(t,\omega))]_{w=M} \geq 0. \end{aligned} \quad (A.5)$$

The time derivative of V is calculated as follows:

$$\begin{aligned} \frac{dV(M,t)}{dt} = E\Bigg[\{\tau_c(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), \ddot{\theta}_r(t,\omega), \dot{\theta}_r(t,\omega), \\ \theta_r(t,\omega)) + \tau_{ext}(t,\omega)\}^T \times \frac{d}{dt}\{\tau_c(\ddot{\theta}(t,\omega), \dot{\theta}(t,\omega), \theta(t,\omega), \\ \ddot{\theta}_r(t,\omega), \dot{\theta}_r(t,\omega), \theta_r(t,\omega)) + \tau_{ext}(t,\omega)\}\Bigg]_{w=M}. \end{aligned} \quad (A.6)$$

To simplify the expression, we shorten each variable by omitting (t, ω).

$$\begin{aligned} \frac{dV}{dt} = E\Bigg[(\tau_c + \tau_{ext})^T\Bigg\{\frac{\partial\tau_c}{\partial\ddot{\theta}}\frac{d\ddot{\theta}}{dt} + \frac{\partial\tau_c}{\partial\dot{\theta}}\frac{d\dot{\theta}}{dt} + \frac{\partial\tau_c}{\partial\theta}\frac{d\theta}{dt} + \frac{\partial\tau_c}{\partial\ddot{\theta}_r}\frac{d\ddot{\theta}_r}{dt} \\ + \frac{\partial\tau_c}{\partial\dot{\theta}_r}\frac{d\dot{\theta}_r}{dt} + \frac{\partial\tau_c}{\partial\theta_r}\frac{\partial\theta_r}{dt} + \frac{d\tau_{ext}}{dt}\Bigg\}\Bigg]_{w=M}. \end{aligned} \quad (A.7)$$

Equation (A.1) can be expressed in explicit form with respect to $\ddot{\theta}$ as follows:

$$\ddot{\theta}(t,\omega) = h(\dot{\theta}(t,\omega), \theta(t,\omega), w(t,\omega), \ddot{\theta}_r(t,\omega), \dot{\theta}_r(t,\omega), \theta_r(t,\omega), \tau_{ext}(t,\omega)). \quad (A.8)$$

By differentiating eqn (A.8) with respect to t, and then substituting it for $d\ddot{\theta}/dt$ in eqn (A.7), the following equation is obtained:

$$\frac{dV}{dt} = E\left[(\tau_c + \tau_{ext})^T\left\{-K_2\left(\frac{\partial\ddot{\theta}}{\partial w}\frac{dw}{dt} + \frac{\partial\ddot{\theta}}{\partial\dot{\theta}}\frac{d\dot{\theta}}{dt} + \frac{\partial\ddot{\theta}}{\partial\theta}\frac{d\theta}{dt} + \frac{\partial\ddot{\theta}}{\partial\ddot{\theta}_r}\frac{d\ddot{\theta}_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\dot{\theta}_r}\frac{d\dot{\theta}_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\theta_r}\frac{d\theta}{dt} + \frac{\partial\ddot{\theta}}{\partial\tau_{ext}}\frac{d\tau_{ext}}{dt}\right) - K_1\frac{d\dot{\theta}}{dt} - K_0\frac{d\theta}{dt} + K_2\frac{d\ddot{\theta}_r}{dt} + K_1\frac{d\dot{\theta}_r}{dt} + K_0\frac{d\theta_r}{dt} + \frac{d\tau_{ext}}{dt}\right\}\right]_{w=M}. \quad (A.9)$$

We sort the above terms into two parts with respect to w and θ whose changing rate are quite different from each other.

$$\frac{dV}{dt} = E\left[(\tau_c + \tau_{ext})^T\left\{-K_2\frac{\partial\ddot{\theta}}{\partial w}\right\}\frac{dw}{dt}\right]_{w=M} + E\left[(\tau_c + \tau_{ext})^T\left\{-K_2\left(\frac{\partial\ddot{\theta}}{\partial\dot{\theta}}\frac{d\dot{\theta}}{dt} + \frac{\partial\ddot{\theta}}{\partial\theta}\frac{d\theta}{dt} + \frac{\partial\ddot{\theta}}{\partial\ddot{\theta}_r}\frac{d\ddot{\theta}_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\dot{\theta}_r}\frac{d\dot{\theta}_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\theta_r}\frac{d\theta_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\tau_{ext}}\frac{d\tau_{ext}}{dt}\right) - K_1\frac{d\dot{\theta}}{dt} - K_0\frac{d\theta}{dt} + K_2\frac{d\ddot{\theta}_r}{dt} + K_1\frac{d\dot{\theta}_r}{dt} + K_0\frac{d\theta_r}{dt} + \frac{d\tau_{ext}}{dt}\right\}\right]_{w=M} \quad (A.10)$$

The second term, $E[\cdot]$, on the right in eqn (A.10) vanishes if θ, $\dot{\theta}$, $\ddot{\theta}$, θ_r, $\dot{\theta}_r$, $\ddot{\theta}_r$, τ_{ext} are strongly stationary stochastic processes and the changing rate of w is sufficiently slow (i.e., the learning rate η is very small). The reason for this is explained below.

The second term of eqn (A.10) is the same with dV/dt when M is constant, as shown in the following equation:

$$\frac{dV_{M\,const}}{dt} = E\left[(\tau_c + \tau_{ext})^T\left\{-K_2\left(\frac{\partial\ddot{\theta}}{\partial\dot{\theta}}\frac{d\dot{\theta}}{dt} + \frac{\partial\ddot{\theta}}{\partial\theta}\frac{d\theta}{dt} + \frac{\partial\ddot{\theta}}{\partial\ddot{\theta}_r}\frac{d\ddot{\theta}_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\dot{\theta}_r}\frac{d\dot{\theta}_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\theta_r}\frac{d\theta_r}{dt} + \frac{\partial\ddot{\theta}}{\partial\tau_{ext}}\frac{d\tau_{ext}}{dt}\right) - K_1\frac{d\dot{\theta}}{dt} - K_0\frac{d\theta}{dt} + K_2\frac{d\ddot{\theta}_r}{dt} + K_1\frac{d\dot{\theta}_r}{dt} + K_0\frac{d\theta_r}{dt} + \frac{d\tau_{ext}}{dt}\right\}\right] \quad (A.11)$$

Here, we use the proposition that the expectation of any Baire function of a strongly stationary stochastic process does not depend on time [see for example, Ito (1953)]. Because of ASSUMPTION 3, $\ddot{\theta}$, $\dot{\theta}$, θ, and L are Baire functions of the strongly stationary processes, $\ddot{\theta}_r$, $\dot{\theta}_r$, θ_r, and τ_{ext}. Consequently, $V_{M\,const}$ does not depend on time, and $dV_{M\,const}/dt$ is equal to zero. Thus, the second term on the right in eqn (A.10) vanishes. Hence, we obtain:

$$\frac{dV}{dt} = E\left[(\tau_c + \tau_{ext})^T\left\{-K_2\frac{\partial\ddot{\theta}}{\partial w}\right\}\frac{dw}{dt}\right]_{w=M}. \quad (A.12)$$

We calculate the partial derivative of eqn (A.8) with respect to w while referring to eqn (A.1) in order to replace $\partial\ddot{\theta}/\partial w$ in eqn (A.12).

$$\frac{\partial f}{\partial\ddot{\theta}}\frac{\partial\ddot{\theta}}{\partial w} - \frac{\partial\Phi}{\partial\ddot{\theta}}\frac{\partial\ddot{\theta}}{\partial w} - \frac{\partial\Phi}{\partial w} = -K_2\frac{\partial\ddot{\theta}}{\partial w} \Leftrightarrow \frac{\partial\ddot{\theta}}{\partial w} = \left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1}\frac{\partial\Phi}{\partial w}. \quad (A.13)$$

Then, eqn (A.12) can be expressed as:

$$\frac{dV}{dt} = E\left[(\tau_c + \tau_{ext})^T\left\{-K_2\left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1}\frac{\partial\Phi}{\partial w}\right\}\frac{dw}{dt}\right]_{w=M} = E\left[-(\tau_c + \tau_{ext})^T K_2\left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1} \times \frac{\partial\Phi}{\partial w}\eta\left(\frac{\partial\Phi}{\partial w}\right)^T(\tau_c + \tau_{ext})\right]_{w=M}. \quad (A.14)$$

Therefore, if the following symmetric matrix B is positive semi-definite, that is,

$$B = \frac{A + A^T}{2} \geq 0$$

where

$$A = K_2\left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1}\frac{\partial\Phi}{\partial w}\eta\left(\frac{\partial\Phi}{\partial w}\right)^T, \quad (A.15)$$

we obtain:

$$\frac{dV}{dt} \leq 0. \quad (A.16)$$

Even though eqn (A.15) is a sufficient condition for convergence, not a necessary one, selective adaptation can be applied to guarantee the nondivergence during learning. We can calculate matrix A in eqn (A.15) if the object inertial property, $\partial f/\partial\ddot{\theta}$, is known. Thus, the neural network parameter, w, can be changed using the adaptation rule, eqn (A.2), while B in eqn (A.15) is a positive-definite matrix (i.e., all eigenvalues of B are positive). Consequently, the Lyapunov function V might be decreased. Near the ideal final state, $\partial\Phi/\partial\ddot{\theta}$ converges to $\partial f/\partial\ddot{\theta}$. Then,

$$A = K_2\left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1}\frac{\partial\Phi}{\partial w}\eta\left(\frac{\partial\Phi}{\partial w}\right)^T \to \frac{\partial\Phi}{\partial w}\eta\left(\frac{\partial\Phi}{\partial w}\right)^T. \quad (A.17)$$

Thus, the final state is locally stable because dV/dt is negative semi-definite.

For strict convergence of this learning scheme in the general case, we might use the following modifications. $K_2((\partial f/\partial\ddot{\theta}) - (\partial\Phi/\partial\ddot{\theta}) + K_2)^{-1}$ can be calculated if $\partial f/\partial\ddot{\theta}$ is known. Thus, the learning rule can be modified as:

$$\frac{dw}{dt} = \eta\left(\frac{\partial\Phi}{\partial w}\right)^T\left(K_2\left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1}\right)^T(\tau_c + \tau_{ext}). \quad (A.18)$$

Then,

$$\frac{dV}{dt} = E[-(\tau_c + \tau_{ext})^T\Lambda(\tau_c + \tau_{ext})]_{w=M} \leq 0 \quad (A.19)$$

$$\because \Lambda = K_2\left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1}\frac{\partial\Phi}{\partial w}\eta\left(\frac{\partial\Phi}{\partial w}\right)^T\left(K_2\left(\frac{\partial f}{\partial\ddot{\theta}} - \frac{\partial\Phi}{\partial\ddot{\theta}} + K_2\right)^{-1}\right)^T$$

is positive semi-definite.

Equations (A.5) and (A.16) hold only when $w = \hat{w}$ for sufficiently rich θ except at the local minimum, where $\partial\Phi/\partial w = 0$. Consequently, we conclude that w asymptotically converges to the optimal set of synaptic weights, $\hat{w}$, in the sense of quadratic mean convergence, if the local minima are avoided during learning. By using an annealing method for changing the weights, as in Geman (1979), the solution of eqn (A.2) will converge to the global minimum.

APPENDIX B

The convergence of the neural network in the NRL scheme is briefly explained below. The difference between the actual output and the desired output of the NNFC, Φ_d, is defined as P. P is given by the next equation using eqns (13), (16), and (17).

$$P = \Phi - \Phi_d = -(I + R(\theta)K_2^{-1})\tau_c \quad I\text{: unit matrix} \tag{B.1}$$

We consider the following function J as a possible Lyapunov function of this learning scheme:

$$J = E[\tfrac{1}{2}P^T P]_{w=M} \geq 0. \tag{B.2}$$

By the same procedure as used to prove the convergence of the IDML scheme, the following equation is obtained:

$$\frac{dJ}{dt} = E\left[-\tau_c^T(I + R(\theta)K_2^{-1})\frac{\partial \Phi}{\partial w}\eta\left(\frac{\partial \Phi}{\partial w}\right)^T \tau_c\right]_{w=M} + \frac{dJ}{d\hat{t}} \tag{B.3}$$

where

$$\frac{dJ}{d\hat{t}} = E\left[P^T\left\{\sum_{k=0}^{1}\frac{\partial P}{\partial\left(\frac{d^k\theta}{dt^k}\right)}\frac{d}{dt}\left(\frac{d^k\theta}{dt^k}\right) + \sum_{k=0}^{2}\frac{\partial P}{\partial\left(\frac{d^k\theta_r}{dt^k}\right)}\frac{d}{dt}\left(\frac{d^k\theta_r}{dt^k}\right)\right\}\right]_{w=M}.$$

The second term in the right of eqn (B.3) vanishes because θ, θ_r are assumed to be mixing process, i.e., when the rate of changing weights, η, is very small. Thus, we next examine the first term in the right of eqn (B.3). When the following symmetric matrix D can be designed to be positive-semi-definite during learning, for example,

$$D = \frac{C + C^T}{2} \geq 0, \tag{B.4}$$

where

$$C = (R(\theta)K_2^{-1})^T\frac{\partial \Phi}{\partial w}\eta\left(\frac{\partial \Phi}{\partial w}\right)^T,$$

we can obtain:

$$\frac{dJ}{dt} \leq 0. \tag{B.5}$$

Consequently, the function Φ of the neural network ideally acquires nonlinear compensation represented in eqn (17) after learning under the same assumptions used in Appendix A and the condition [eqn. (B.4)].