# Neuro-fuzzy control for DC motor friction compensation

Jun Oh JANG and Pyeong Gi LEE

Department of Computer Control Engineering
Uiduk University, Kyongju, 780-713, South Korea
Tel : +82-054-760-1624, Fax : +82-054-760-1605, Email : jojang@mail.uiduk.ac.kr

**Abstract :** This paper presents an application of a neuro-fuzzy controller for compensating the effects induced by the friction in a DC motor system. The neuro-fuzzy controller is a combination of a linear controller and a neuro-fuzzy network which compensates for nonlinear friction. The proposed scheme is implemented and tested on an IBM PC-based DC motor control system. The algorithm, simulations, and experimental results are described. The results are relevant for precision drive, such those found in industrial robots.

## 1. Introduction

Unknown and unmodeled nonlinearities play an important role in high-precision control. For instance, if the control strategy relies on linearizing a nonlinear system based on the fact that the nonlinearities are known [1], any cancellation by feedback that is not exact may produce undesirable closed-loop behavior. Friction is one of the most common nonlinearities present in mechanical system. For accurate position and low-velocity control, control strategies usually rely on accurate estimation of friction. The performance possible from a system, especially at low velocities, with reversals, is usually limited since accurate modeling of friction and parasitic effects is difficult.

Friction is a very complex phenomenon caused by one or more nonlinearities such as stiction, hysteresis, stribeck effect, stick-slip, velocity dependence, and input frequency dependence. All these nonlinearities are particularly conspicuous during motion at low velocities, especially with zero crossings. Various techniques have been investigated in the past to deal with the problem caused by friction [2]. Some compensation techniques without models, which rely primarily on linear parametrizations, have been proposed. The most common of such techniques is high gain compensation, and although such techniques do not involve explicit characterization of friction, some of the drawbacks are controller saturation and instability at low error values [3]. On the other hand, several friction models have been widely studied for providing a good understanding of the structure of friction. Models such as the Coulomb friction model, Dahl model [4], exponential model [5], bristle model [6], reset integral model [6], state variable model [7], and bristle based dynamic model [8] have been reported to estimate friction and some of them have been used in compensation strategies [2, 3, 9]. Adaptive observers have been reported for systems but under the assumption of some friction model [2, 10]. Canudas de Wit et al. [5] proposed an exponential adaptive friction compensation algorithm to estimate the unknown parameters under possible structural inaccuracies in the model. In their algorithm, the parameters of the exponential friction model and the inertia parameters of a robot manipulator were adaptively updated. Friedland and Park [10] present an adaptive compensation algorithm for a constant Coulomb friction model. In their approach, a nonlinear reduced-order observer was introduced, which forces the error between the estimated and actual parameter vector to converge asymptotically to zero.

The problem of friction becomes acute with high performance goals for various applications, and the choice of an appropriate model for a problem is an open question [11]. Also, there is a recognized need for precise position control at the submicron level of accuracy, e.g., advanced semiconductor manufacturing [11]. Few of the existing friction models are adaptive, and since most of the adaptive control techniques rely on linear parameterizations, they usually tend to be restrictive. A neural network on the other hand, represents a class of nonlinear parameterizations with attractive properties including learning [12-15]. Several friction compensation methods using neural networks have been reported [16, 17]. However few have reported any analytical developments or constructive design approaches.

In this paper, we present the idea of compensating the nonlinear friction with a proposed neuro-fuzzy controller. The neuro-fuzzy controller is composed of a PI controller and a neuro-fuzzy network(NFN) in parallel, which is trained through the neural network identifier(NNI) by an indirect learning scheme. By an example of a DC motor system with nonlinear friction, this paper gives an implementation of the idea of compensating the nonlinear friction using a PI controller and a neuro-fuzzy network.

## 2. DC motor friction modeling

A DC motor with a permanent magnet was used in our experiments. The motor is provided with an electronic amplifier with current feedback. Such a motor is commonly used in robots and precision servos. If all inertia are reflected to the motor axis, the motor can be described by the following model:

$$J\ddot{q}(t) = -B\dot{q}(t) + T(t) - T_f(t) - T_d(t) \qquad (1)$$

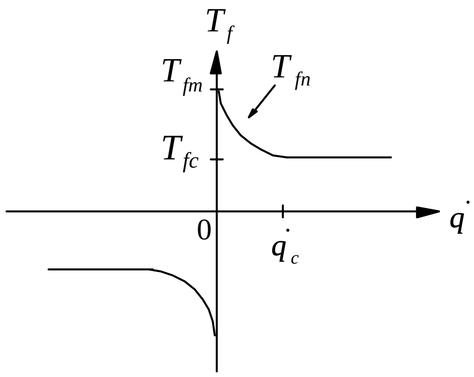



Fig. 1. Friction characteristic.

where $\dot{q}(t)$ is the velocity of the motor shaft, $J$ is the moment of inertia reflected to the motor axis, $B$ is the viscous friction, $T$ is the control input torque, $T_f$ is nonlinear friction torque, and $T_d$ is the load disturbance. For the purpose of the investigation of the friction compensation, phenomena like compliance and torque ripple are not included in the model (1).

A general friction characteristic that is valid for low velocities regime can be written with the friction $T_f$ as

$$T_f(\dot{q}(t), \ddot{q}(t)) = T_{fc}[\mathrm{sgn}(\dot{q}(t))] + T_{fn}[\dot{q}(t), sign(\ddot{q}(t))] \qquad (2)$$

where $T_{fc}$ is the constant Coulomb friction term dependent solely on the sign of velocity, $T_{fn}[\dot{q}(t), sign(\ddot{q}(t))]$ is the second nonlinear friction term. No assumptions are made on the shape of the second nonlinear component $T_{fn}[\dot{q}(t), sign(\ddot{q}(t))]$ except that it vanishes beyond some critical relative velocity, $\dot{q}_c$ known a priori.

## 3. Indirect neuro-fuzzy control for DC motor friction compensation

The structure of the neural network identifier(NNI) and design techniques of the proposed neuro-fuzzy controller are presented in this Section for experimentally controlling the velocity of the DC motor, $\dot{q}(t)$. Fig. 2 shows the NNI and the proposed neuro-fuzzy controller which has the

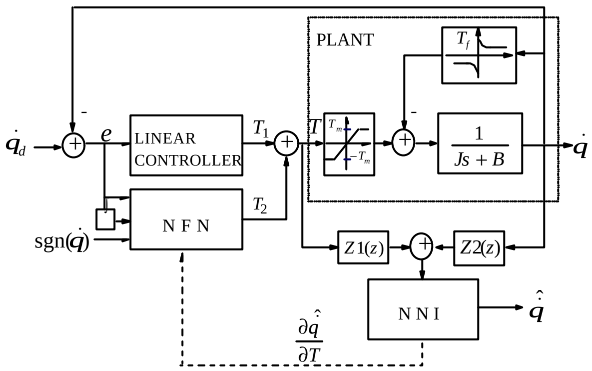



Fig. 2. The proposed neuro-fuzzy control system.

linear controller and the neuro-fuzzy network(NFN).

The NNI which emulates the dynamic behavior of the plant is necessary for the indirect neuro-fuzzy control scheme and plays an important role in the update of the neuro-fuzzy controller parameters. This update is achieved by obtaining a better estimate of the gradient in the gradient descent technique using the trained NNI. It is noted that the NNI is trained in a series-parallel mode [13] before being used for control. The parameters of the NNI are updated to follow the dynamics of the plant in the reported research since this is found to increase accuracy. The output of the NNI in Fig. 3(a), with the number of nodes in successive layers 4-5-1, is given by

$$\hat{\dot{q}} = \sum_{m=1}^{5}[w^i_{m1} \cdot \Psi^i(\sum_{l=1}^{4} v^i_{lm} \cdot x^i_l)] \qquad (3)$$

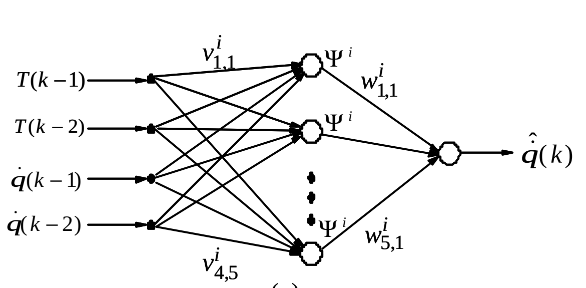
with $\Psi^i(\cdot)$ the hyperbolic tangent function, $v^i_{lm}$ the interconnection weights from first to second layer, $w^i_{m1}$ the interconnection weights from second to third layer, and the superscript $i$ implies NNI. The inputs to the NNI, $x^i = [x^i_1, x^i_2, x^i_3, x^i_4]^T = [T(k-1), T(k-2), \dot{q}(k-1), \dot{q}(k-2)]^T$, are composed of the control input and the plant output. For the NNI in Fig. 2, $Z1(z) = [z^{-1}, z^{-2}, 0, 0]^T$ and $Z2(z) = [0, 0, z^{-1}, z^{-2}]^T$.

The performance index for training the NNI is:

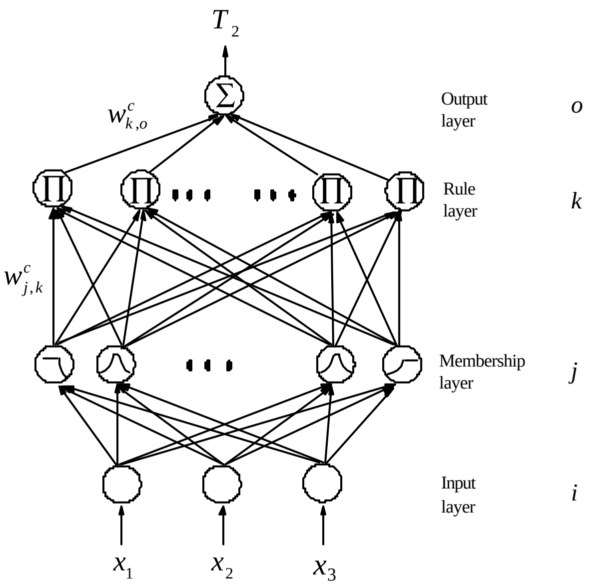
$$E^i(k) = \frac{1}{2}e^i(k)^2 = \frac{1}{2}[\dot{q}(k) - \hat{\dot{q}}(k)]^2 \qquad (4)$$

where $e^i(k)$ is defined as the error between the plant output, $\dot{q}(k)$ and the output of the NNI, $\hat{\dot{q}}(k)$. The second to third layer weights of the NNI, $w^i_{m1}$ are updated by back-propagating the error by the following equation:

$$\begin{aligned} w^i_{m1}(k+1) &= w^i_{m1}(k) - \boldsymbol{h} \cdot \frac{\partial E^i(k)}{\partial w^i_{m1}(k)} \\ &= w^i_{m1}(k) + \boldsymbol{h} \cdot \frac{\partial \hat{\dot{q}}(k)}{\partial w^i_{m1}(k)} \cdot e^i(k) \qquad (5) \\ &= w^i_{m1}(k) + \boldsymbol{h} \cdot \Psi^i(\sum_{l=1}^{4} v^i_{lm}(k) \cdot x^i_l(k)) \cdot e^i(k) \end{aligned}$$




(a)




(b)

Fig. 3. Structure of (a) the NNI and (b) the NFN.

where $\eta$ is the step size. Similarly, the first to second layer weights of the NNI, $v^i_{lm}$ are updated by following equation:

$$v^i_{lm}(k+1) = v^i_{lm}(k) - \eta \cdot \frac{\partial E^i(k)}{\partial v^i_{lm}(k)}$$
$$= w^i_{lm}(k) + \eta \cdot \frac{\partial \hat{\dot{\theta}}(k)}{\partial v^i_{lm}(k)} \cdot e^i(k) \quad (6)$$
$$= v^i_{lm}(k) + \eta \cdot x^i_l(k) \cdot \dot{\Psi}^i(\sum_{l=1}^{4} v^i_{lm}(k) \cdot x^i_l(k))$$
$$\cdot w^i_{m1}(k) \cdot e^i(k)$$

where $\dot{\Psi}^i(\cdot)$ is the derivative of $\Psi^i(\cdot)$.

Control input $T$ to the plant is defined as:

$$T = T_1 + T_2 \quad (7)$$

where $T_1$ is the output of the linear controller and $T_2$ is the output of the NFN. The output of the NFN in Fig. 3(b), with the number of nodes in successive layers 3-6-9-1, is given by

$$T_2 = \sum_{k=1}^{9} [w^c_{ko} \cdot \prod_j \{w^c_{jk} \cdot \exp(-\frac{(x_i - m_{ij})^2}{(\sigma_{ij})^2})\}],$$
$$i = 1,2,3, \quad j = 1,2,\cdots,6 \quad (8)$$

where $w^c_{ko}$ are the interconnection weights from rule layer to output layer, $w^c_{jk}$, which are assumed to be unity, are the interconnection weights from membership layer to rule layer, $m_{ij}$ and $\sigma_{ij}$ are, respectively, the mean and the standard deviation of the function in the $j$ th term of the $i$ th input linguistic variable $x_i$ to the node of membership layer. Each node $k$ in the rule layer is denoted by $\prod$, which multiplies the input signals and outputs the result of the product. $x^c = [x^c_1, x^c_2, x^c_3]^T = [e(k), \int e(k), \mathrm{sgn}(\dot{\theta}(k))]^T$ is the input of the NFN. The inputs to the neuro-fuzzy network, $e(k)$ and $\int e(k)$, are selected for compensation of nonlinearities in the plant since the PI controller is used for stabilization of plant dynamics. The signum function, $\mathrm{sgn}(\cdot)$, is needed in Coulomb friction term.

Since the NFN do not directly learn the nonlinearity of the plant, the interconnection weights of the NFN in Fig. 2 are updated by an indirect neuro-fuzzy control scheme with the performance index:

$$\hat{E}_c(k) = \frac{1}{2}\hat{e}_c(k)^2 = \frac{1}{2}[\dot{\theta}_d(k) - \hat{\dot{\theta}}(k)]^2 \quad (9)$$

where $\hat{e}_c(k)$ is defined as the identified tracking error between the reference signal and the output of the NNI. The indirect neuro-fuzzy control scheme of the neuro-fuzzy controller is as follows: the plant dynamics are identified by the NNI through a learning process, during which connection weights are updated in a direction to minimize the sum of squared errors between the plant output, $\dot{\theta}$ and the output of the NNI, $\hat{\dot{\theta}}$. Meanwhile, the partial derivative, $\partial \dot{\theta} / \partial T$, which is calculated by using the NNI and the error signal, $\hat{e}_c(k)$, is used to update the weights of the NFN by employing the dynamic back propagation scheme [13, 18], which provides a better estimate of gradients, $\partial \hat{\dot{\theta}} / \partial T$. Therefore, the rule layer to output layer weights of the NFN, $w^c_{ko}$, are updated by back propagation of the identified tracking error, $\hat{e}_c(k)$, through the NNI by the following equation:

$$w^c_{ko}(k+1) = w^c_{ko}(k) - \eta_c \cdot \frac{\partial \hat{E}_c(k)}{\partial w^c_{ko}(k)}$$
$$= w^c_{ko}(k) + \eta_c \cdot (\frac{\partial T_2(k)}{\partial w^c_{ko}(k)} + \frac{\partial T_1(k)}{\partial w^c_{ko}(k)}) \cdot \frac{\partial \hat{\dot{\theta}}(k)}{\partial T(k)} \cdot \hat{e}_c(k)$$
$$= w^c_{ko}(k) + \eta_c \cdot \prod_j \{w_{jk}(k) \cdot \exp(-\frac{(x_i(k) - m_{ij}(k))^2}{\sigma^2_{ij}(k)})\} \quad (10)$$
$$\cdot \sum_{m=1}^{5} [v^i_{1m}(k) \cdot \dot{\Psi}^i(\sum_{l=1}^{4} v^i_{lm}(k) \cdot x^i_l(k)) \cdot w^i_{m1}(k)] \cdot \hat{e}_c(k)$$

utilizing the fact that $\frac{\partial T_1(k)}{\partial w^c_{ko}(k)} = 0$ and

$$\frac{\partial T_2(k)}{\partial w_{ko}^c(k)} = \prod_j \{w_{jk}(k)\cdot \exp(-\frac{(x_i(k)-m_{ij}(k))^2}{s_{ij}^2(k)})\} \text{ from (8).}$$

The mean and the standard deviation of the NFN, $m_{ij}$ and $s_{ij}$, respectively, are similarly updated by the following equation:

$$\begin{aligned} m_{ij}(k+1) &= m_{ij}(k) - h_m \cdot \frac{\partial \hat{E}_c(k)}{\partial m_{ij}(k)} \\ &= m_{ij}(k) + h_m \cdot \frac{\partial T_2(k)}{\partial m_{ij}(k)} \cdot \frac{\partial \hat{\dot{q}}(k)}{\partial T(k)} \cdot \hat{e}_c(k) \\ &= m_{ij}(k) + h_m \cdot \frac{2(x_i(k)-m_{ij}(k))}{s_{ij}^2(k)} \cdot \exp(-\frac{(x_i(k)-m_{ij}(k))^2}{s_{ij}^2(k)}) \\ &\cdot w_{ko}^c(k) \cdot \sum_{m=1}^{5} [v_{1m}^i(k) \cdot \dot{\Psi}^i (\sum_{l=1}^{4} v_{lm}^i(k) \cdot x_l^i(k)) \cdot w_{m1}^i(k)] \cdot \hat{e}_c(k) \end{aligned} \quad (11)$$

where $h_m$ is the learning rate of the mean of the Gaussian function, and

$$\begin{aligned} s_{ij}(k+1) &= s_{ij}(k) - h_s \cdot \frac{\partial \hat{E}_c(k)}{\partial s_{ij}(k)} \\ &= s_{ij}(k) + h_s \cdot \frac{\partial T_2(k)}{\partial s_{ij}(k)} \cdot \frac{\partial \hat{\dot{q}}(k)}{\partial T(k)} \cdot \hat{e}_c(k) \\ &= s_{ij}(k) - h_s \cdot \frac{2(x_i(k)-m_{ij}(k))^2}{s_{ij}^3(k)} \cdot \exp(-\frac{(x_i(k)-m_{ij}(k))^2}{s_{ij}^2(k)}) \\ &\cdot w_{ko}^c(k) \cdot \sum_{m=1}^{5} [v_{1m}^i(k) \cdot \dot{\Psi}^i (\sum_{l=1}^{4} v_{lm}^i(k) \cdot x_l^i(k)) \cdot w_{m1}^i(k)] \cdot \hat{e}_c(k) \end{aligned} \quad (12)$$

where $h_s$ is the learning rate of standard deviation of the Gaussian functions.

## 4. Simulation and experimental results

In this Section, we illustrate the effectiveness of the proposed control scheme by computer simulations and experiments on a dc motor system. The experimental set up is shown in Fig. 4. It consists of a dc motor with a gear and load, an encoder and a counter for output signal, a Digital-to-Analog(D/A) converter and a servo amplifier for control signal, and an IBM PC equipped with an Intel 8255 based interface card. The voltage output from the computer is amplified using a pulse width modulated amplifier. An optical encoder with a quadrature decoder chip is used for angular position measurement. Angular velocity has been calculated from this position measurement by differentiation. In the experimental set up, the main control algorithm is implemented at a 100 Hz sampling rate via an IBM PC with an Intel 486DX-66 microprocessor. The proposed algorithm is written in C language.

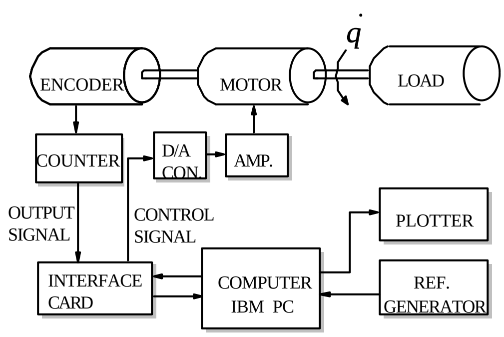
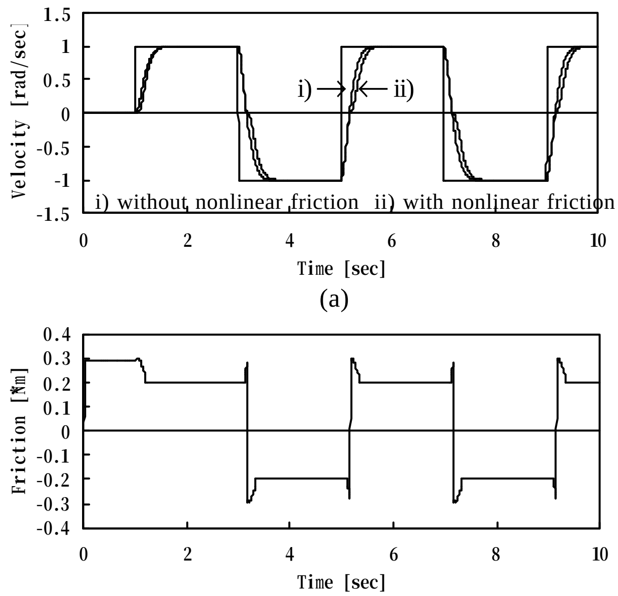



Fig. 4. Experimental set-up.




(b)

Fig. 5. (a) Simulation results of the DC motor with/ without the nonlinear friction by the PI controller and (b) the nonlinear friction.

From the frequency response and the curve fitting method, we obtained the linear model of the DC motor with the gear and load as follows:

$$G(s) = \frac{1}{0.0143\, s + 0.9385}. \quad (13)$$

The proportional integral (PI) gains of the closed loop system are adjusted after obtaining by the Ziegler-Nichols method [19]. For fairness of comparison, the gain of PI controller are chosen to the identical gains of the PI part in the neuro-fuzzy controller; the proportional gain $K_p = 4$ and the integral gain $K_I = 2$. The learning rate $h_c = 0.015$, $h_m = 0.01$, and $h_s = 0.012$ for the NFN and $h = 0.015$ for the NNI. Generally, there is nonlinear friction in the motor bearings and in the gear train. The simulation results of the DC motor with/without the nonlinear friction by the PI controller are shown in Fig. 5. When the nonlinear friction is considered, the

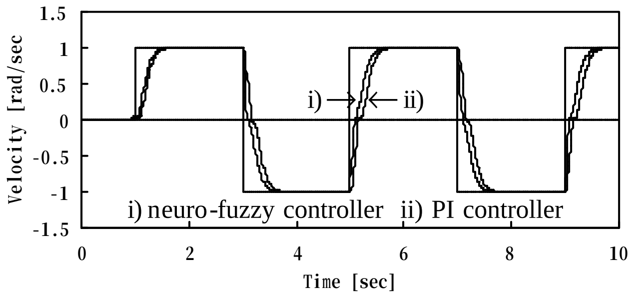



Fig. 6. Simulation results of the DC motor with the nonlinear friction by the PI controller and by the neuro-fuzzy controller.

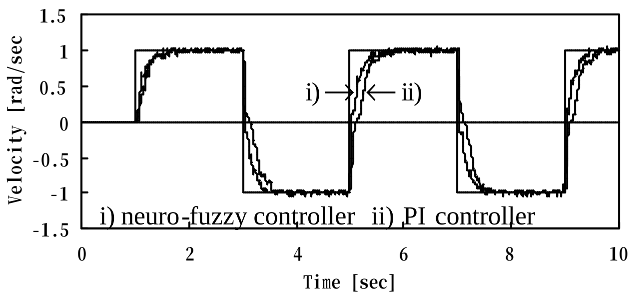



Fig. 7. Experimental results of the DC motor with the nonlinear friction by the PI controller and by the neuro-fuzzy controller.

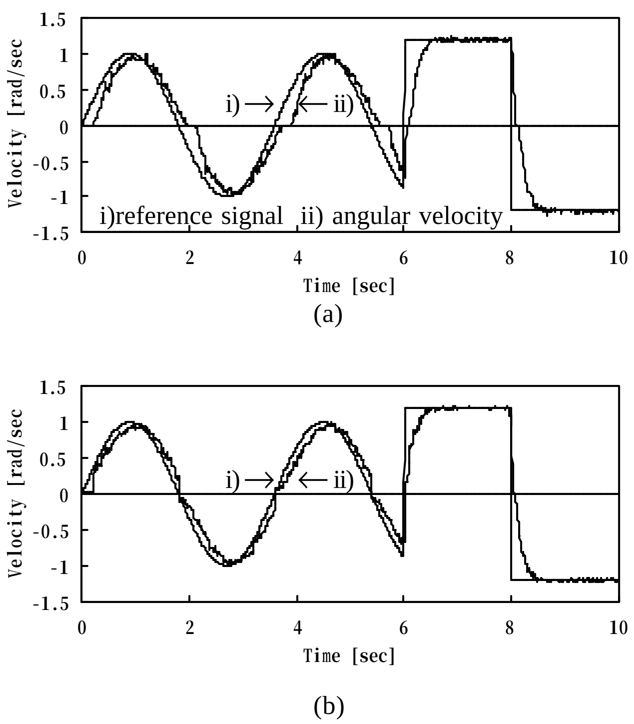



Fig. 8. Experimental results with the sinusoidal and step reference signal (a) by the PI controller and (b) by the neuro-fuzzy controller.

performance is degraded by the nonlinear friction in Fig. 5(b). Therefore, we trained the neuro-fuzzy controller as described in Section 3 in order to compensate for the effects of nonlinear friction. In simulations and experiments, the NNI and the NFN are on-lined trained. The simulation results of the DC motor with the nonlinear friction by the PI controller and by the neuro-fuzzy controller are shown in Fig. 6. The neuro-fuzzy controller exhibits an improvement in its response to the step input compared with the PI controller. The experiments on the DC motor were carried out with the PI controller and with the neuro-fuzzy controller for friction compensation. The experimental results are given in Fig. 7, which shows similar phenomena to found in the simulation.

We also investigate the motor velocity with the sinusoidal and step reference signal. From the response of the PI controller in Fig. 8, we see a deadzone in the vicinity of zero velocity of the sinusoidal reference signal. However, the neuro-fuzzy controller reduces the deadzone width significantly because of the nonlinear friction compensation.

## 5. Conclusions

In this paper, we have developed a neuro-fuzzy controller for friction compensation of a DC motor system. The neuro-fuzzy controller consists of a linear controller and an NFN. An indirect learning scheme for training the NFN is adopted. The friction compensation by the neuro-fuzzy controller has been found excellent since the friction depends on the operating conditions. The proposed neuro-fuzzy controller has been applied to a DC motor system and its benefit has been clearly demonstrated in the experiments where control law is implemented on an IBM PC.

## References

[1] J. E. Slotine and W. Li, *Applied nonlinear control.* Prentice-Hall, 1991.

[2] B. Armstrong-Helouvry, P. Dupont, and C. Canudus, "A survey of models, analysis tools, and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083-1138, 1994.

[3] S. C. Southward, C. J. Radcliffe, and C. R. MacCluer, "Robust nonlinear stick-slip friction compensation," *ASME J. Dynamic Syst., Measurement, Contr.*, vol. 113, pp. 639-645, 1991.

[4] P. R. Dahl, "Measurement of solid friction parameters of ball bearings," *Proc. 6th Annu. Symp. Incremental Motion Control Systems and Devices*, pp. 49-60, 1977.

[5] C. Canudus, P. Noel, A. Aubin, and B. Brogliato, "Adaptive friction compensation in robot manipulator : Low velocities," *Int. J. Robot. Res.*, vol. 10. no. 3, pp. 189-199, 1991.

[6] D. A. Haessig and B. Friedland, "On the modeling and simulation of friction," *ASME J. Dynamic Syst., Measurement, Contr.*, vol. 113, pp. 354-362, 1991.
[7] J. R. Rice and A. L. Ruina, "A stability of steady frictional slipping," *J. Appl. Mech.*, vol. 50, pp. 343-249, 1983.
[8] C. Canudas, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Trans. Automat. Contr.*, vol 40, pp. 419-425, 1995.
[9] P. E. Dupont, "Avoiding stick-slip through PD control," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1094-1097, 1994.
[10] B. Friedland and Y. Park, "On adaptive friction compensation," *IEEE Trans. Automat. Contr.*, vol 37. pp. 1609-1612, 1992.
[11] B. Friedland, *Advanced Control System Design*, Prentice-Hall, 1996.
[12] S. I. Mistry, S. Chang, and S. S. Nair, "Indirect control of a class of nonlinear dynamic systems," *IEEE Trans. Neural Networks*, vol. 7, pp. 1015-1025, 1996.
[13] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 252-262, 1991.
[14] F. J. Lin, R. J. Wai, and C. C.Lee, "Fuzzy neural network position controller for ultrasonic motor drive using push-pull DC-DC converter," *IEE Proc.-Control theory and Applications*, vol. 146, no. 1, pp. 99-107, 1999.
[15] L. B. Gutierrez, F. L. Lewis, and J. A. Lowe, "Implementation of a neural network tracking controller for a single flexible link: comparison with PD and PID controllers," *IEEE Trans. Ind. Electron.*, vol. 45, no. 2, pp. 307-318, 1998.
[16] A. Tzes, P. Peng, and C. Houng, "Neural network control for DC motor micro-maneuvering," *IEEE Trans. Ind. Electron.*, vol. 42, pp. 516-523, 1995.
[17] J. O. Jang and G. J. Jeon, "A parallel neuro-controller for DC motors containing nonlinear friction," *Neurocomputing*, vol. 30, pp. 233-248, 2000.
[18] J. O. Jang and G. J. Jeon, "Implementation of indirect neuro-control for nonlinear dynamic systems," *Mechatronics*, vol. 9, no. 6, pp. 675-686, 1999.
[19] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, pp. 759-768, 1942.