DESIGN OF FUZZY NEURAL NETWORK-BASED MULTIVARIABLE CONTROLLER FOR MANIPULATORS

Yoshishige Sato

Tsuruoka National College of Technology
Tsuruoka City, Yamagata Prefecture, 997-8511 JAPAN
y-sato@tsuruoka-nct.ac.jp

ABSTRACT

This paper proposes a robust multivariable control design by intelligent control that uses a fuzzy neural network by producing robustness capable of automatically controlling gain against a conventional, fixed PID control system. This structural feature of the proposed controller forms a nonlinear deviation compensator using fuzzy neural networks. Therefore, in multidimensionality the inverse dynamic model portion of the control law is referred to as a linearizing and decoupling control law. This method uses a control law where parameter response leads to critical damping and adaptive changes in gain according to time, making it possible to decouple mutual interference in each multivariable system.

Keywords: fuzzy neural network, multivariable control, decoupling, manipulator control

INTRODUCTION

In general, system design for manipulators regards the controlled object as a single-degree-of-freedom system even in a multi-degree-of-freedom system, and an approximate equation is used to design the control system. A method that constructs an independent control system for each joint axis is employed for numerous products. However, in most cases rigidity is weak and the nonlinear multi-input, multi-output system is subject to gravity and centrifugal force. Conventional methods achieve precision, vibration suppression, and robustness by utilizing PID control, phase compensation control, optimal servo systems, disturbance observer-based systems, or $H\infty$ control systems[(1)–(9)].Sufficient control, however, currently cannot be obtained. In this case, one possible approach is to linearize and decouple manipulator characteristics by compensating for the nonlinear dynamics of the manipulator, based on equations of motion.

On the other hand, fuzzy neural network-based control[(10)–(12)] is being suggested as one type of intelligent control. It possesses the salient feature of constructing a robust control system for factors such as nonlinearity, friction properties, variations in load and system parameters, and unknown disturbances in mechatronic servo systems. However, these proposals are limited to single-degree-of-freedom systems.

The method proposed in this paper constructs on the aforementioned suggested methods to create a multivariable system. A nonlinear manipulator equation of motion is directly utilized to bring about linearization using a neural network, while disturbance suppression and decoupling is controlled using a fuzzy neural network. A control law was devised that adaptively changes the gain according to time in order to continuously critically dampen the nonlinear system whose parameters var with time. The mutual interference of each joint in the multivariable system can be decoupled. In other words, perfect tracking control for the manipulator is achieved by designing a constant spring rate for the entire spring at all times.

This paper leverages these distinctive characteristics to propose a new design method for a multivariable controller for manipulators that utilizes a fuzzy neural network. To verify the effectiveness and feasibility of the proposed method, a DC motor was used as an actuator and to serve as a nonlinear

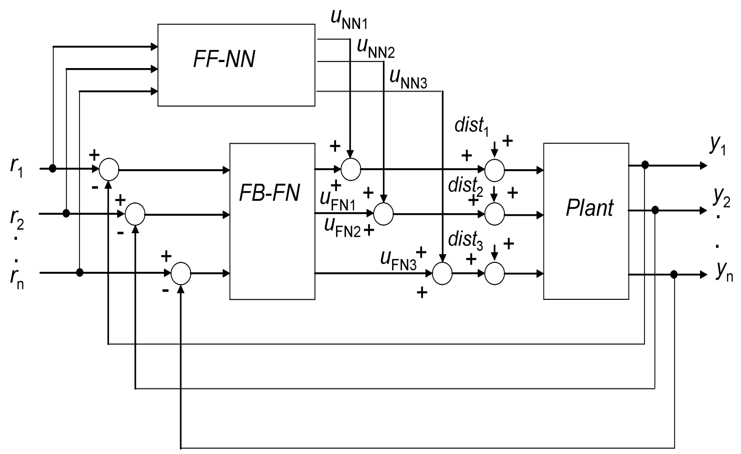


Fig.1 diagram multivariable manipulator control systems

controlled object. A decelerating mechanism determined the positioning that drives the manipulator, and simulations and an experiment were carried out on its application to servo systems.

DESIGN OF A MULTIVARIABLE CONTROLLER UTILIZING A FUZZY NEURAL NETWORK

Control System Summary

Generally, the dynamics of an n-degrees-of-freedom servo system would be described as follows.

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{f}_d(\dot{\boldsymbol{q}}) + \boldsymbol{T}_d = \tau \tag{1}$$

Here, $\boldsymbol{q} = [q_1,\ldots,q_n]^T$ is the n×1 vector expressing joint variables, $\boldsymbol{\tau} = [\tau_1,\cdots,\tau_n]^T$ is the n× 1 vector expressing the input torque, and $\boldsymbol{M}(\boldsymbol{q})$ is the n×n symmetric, positive inertia matrix. $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ is the n×1 vector expressing centrifugal force/ Coriolis force, $\boldsymbol{G}(\boldsymbol{q})$ is the n×1 vector expressing gravity, $\boldsymbol{F}_v$ is the n×n symmetric matrix expressing the coefficient in viscous friction, $\boldsymbol{f}_d(\dot{\boldsymbol{q}})$ is the n×1 vector that corresponds to the coulomb friction, and T_d is the n×1 disturbance vector. Fig. 1 shows a block diagram of that control system.

The multivariable controller is regarded as a combination of single-degree-of-freedom systems and is treated as an independent control system. The control system utilizing fuzzy neural networks is composed of 2 compensators, a feed-forward (FF-NN) compensator that learns the inverse dynamics of the servo system through a neural network, and gain scheduling (FN-GS) through a fuzzy neural network. Fig. 2 illustrates a block diagram of that control system. This control system is a two-degrees-of-freedom control system where target value response characteristics and closed-loop characteristics can be independently designed. To compensate for tracking to target input and dynamic characteristics of disturbance, FN-GS learns offline in order to minimize the square of the deviation signal *e(t)=r(t)-y(t)* of the target signal *r(t)* and the output signal *y(t)*. The proposed system is constructed after learning these values.

The following equation becomes true in relation to equation (1)for the n-degrees-of-freedom servo system when the manipulator is divided into the servo and model base and modeled.

$$f = \alpha f' + \beta \tag{2}$$

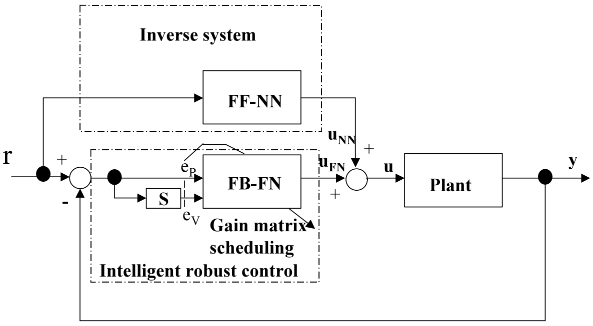


Fig. 2 Block diagram of FN robust fuzzy PID

However, $\alpha = \boldsymbol{M}(\boldsymbol{q}),\ \beta = \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{f}_d(\dot{\boldsymbol{q}}) + \boldsymbol{T}_d$
servo: $\alpha \boldsymbol{f}'$ α: decoupling matrix
model base: β $v \times 1$ vector
Here, $\alpha \boldsymbol{f}'$ represents the FB-FN nonlinear error compensator and β is the FF-NN inverse characteristic that makes up the two-degrees-of-freedom control system. Next, the following equation (3) is adopted as the servo control law for the multi variable system.

$$F' = \ddot{x}_d + k_v\dot{E} + k_pE + k_i\int Edt \tag{3}$$

Applying the manipulator equation (1) to the control input renders the following equation.

$$\ddot{q}_i = k_{pi}(q_{ri} - q_i) + k_{vi}(\dot{q}ri - \dot{q}_i) + k_{Ii}\int (q_{ri} - q_i)\mathrm{dt} \tag{4}$$

The control law that changes the gain according to time must be considered so that constant critical damping occurs in the system. Doing so will decouple each system. The pole will move the complex plane as a function of the object's position and thus the pole's position becomes fixed and a fixed gain cannot be selected.

The oscillation rate for the entire oscillation is made constant. The nonlinearity of the control system is negated with a nonlinear term so that the closed-loop system becomes linear. Therefore this control law is called the linear control law.

$$q_i(s) = \frac{sk_{vi} + k_{pi}}{s^2 + sk_{vi} + k_{pi}} q_{ri}(s) \tag{5}$$

Adjusting $k_{pi}, k_{vi} > 0(i = 1, 2, \cdots m)$ to the adaptive gain, results in $q_i \to q_{ri}, \dot{q}_i \to 0(i = 1, 2, \cdots m)$.

Trajectory Tracking Design

Neural network compensation is considered for equation (1) in Sec. 2.1, and when gravity G(q) and coulomb friction $f_d(q)$ are excluded, the following equation results.

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{T}_d = \boldsymbol{G}_1(\boldsymbol{x}_1)\boldsymbol{W}_1 + \varepsilon \tag{6}$$

The neural network is expressed as:

$$\begin{bmatrix} {}^1\tau_{1NN} \\ {}^2\tau_{1NN} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{11} \cdots \boldsymbol{G}_{1n} & 0 \cdots\cdots 0 \\ 0 \cdots\cdots 0 & \boldsymbol{G}_{21} \cdots \boldsymbol{G}_{2n} \end{bmatrix} \begin{bmatrix} \boldsymbol{W}_{11} \\ \vdots \\ \boldsymbol{W}_{1n} \\ \boldsymbol{W}_{21} \\ \vdots \\ \boldsymbol{W}_{2n} \end{bmatrix} \tag{7}$$

Equation (7) can be expressed as follows:

$$\tau_{1NN} = G_1(x_1)^T W_1 \tag{8}$$

However, the following holds true.

$\boldsymbol{G}_1(\boldsymbol{x}_1)$: nonlinear mapping matrix

$\boldsymbol{W}_1$: load adjust matrix

$\boldsymbol{x}_1$: control input matrix

$\boldsymbol{x}_1 = [q_{d1}\, q_{d2}\, \dot{q}_{d1}\, \dot{q}_{d2}\, \ddot{q}_{d1}\, \ddot{q}_{d2}]^T$

A neural network is introduced to improve the dynamic characteristic of the controlled object. The neural network learns the inverse dynamics of the servo system offline and is used as a feed-forward compensator. Plant dynamic characteristics that apply feedback law on position and velocity can be expressed as

$$J\frac{d^2y(t)}{dt^2} = -k_v\frac{dy(t)}{dt} - k_p y(t) + k_p u(t) \tag{9}$$

by approximating and linearizing nonlinear characteristics as a second order system.
Here, $u(t)$ is input, $y(t)$ is position, while k_p and k_v each represent position and velocity feedback gain. However, k_v is inclusive of k_e, the counter-electromotive force constant for the servo motor. The inverse dynamics of the servo system becomes

$$u(t) = r(t) + \frac{k_v}{k_p}\frac{dr(t)}{dt} + \frac{J}{k_p}\frac{d^2r(t)}{dt^2} \tag{10}$$

when the target trajectory is $r(t)$ and equation (9) is inversely solved as $y(t) = r(t)$. The neural network learns the relationship of $u(t)$ from $r(k)$, $dr/dt(k)$, and $d^2r/dt^2(k)$. Thus, input into the network is the column vector [3*1], I(k), expressed by the following equation.

$$I(k)=[r(k),\ dr/dt(k),\ d^2r/dt^2(k)]^T \tag{11}$$

The input-output relation of each unit utilizes the linear function $f(x)=x$. In addition, network output becomes the estimate value $\hat{\Phi}_k$ of the plant input.

When the output of the intermediate layer is $\psi_i(k)$, neural network output can be expressed in the following manner.

$$\hat{\Phi}_k = \sum_{i=1}^{n} \omega_i^2(k)\psi_i(k) \tag{12}$$

$$\psi_i(k) = \sum_{j=1}^{m} \omega_{ij}(k) Ij(k) \tag{13}$$

However, n is the unit number of the intermediate layer and m represents the unit number of the input layer. Neural network learning is defined as the evaluation function

$$E_k = \frac{1}{2}(\hat{\Phi}_k - \Phi_k)^2 \tag{14}$$

and carried out with Back Propagation (BP). Plant input is Φ_k, and $\hat{\Phi}_k$ is the estimate value of the plant input calculated from the neural network. The simulation and experiment carried out later utilize3 layers,3 inputs, 4 intermediate units, and the neural network of 1 output, as shown in Fig. 3.

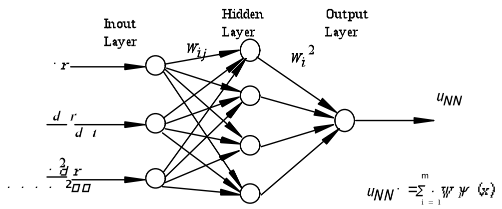


Fig. 3 Feed-forward neural network compensator

Nonlinear Error Control System Design

Fuzzy neural network is considered for Sec. 2.2, and adding coulomb friction $f_d(q)$ and gravity G(q)results in the following equation.

$$\begin{aligned} &M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_v\dot{q} + f_d(\dot{q}) + T_d \\ &= G_2(x_2)W_2 + \varepsilon \end{aligned} \tag{15}$$

Then, when the fuzzy neural network is accounted for the following equation can be obtained.

$$\begin{bmatrix} {}^1\tau_{2FN} \\ {}^2\tau_{2FN} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{11} \cdots \boldsymbol{G}_{1n} & 0 \cdots\cdots 0 \\ 0 \cdots\cdots 0 & \boldsymbol{G}_{21} \cdots \boldsymbol{G}_{2n} \end{bmatrix} \begin{bmatrix} \boldsymbol{W}_{11} \\ \vdots \\ \boldsymbol{W}_{1n} \\ \boldsymbol{W}_{21} \\ \vdots \\ \boldsymbol{W}_{2n} \end{bmatrix} \tag{16}$$

Equation (16) can be expressed as

$$\tau_{2FN} = G_2(x_2)^T W_2 \tag{17}$$

However, the following holds true.

$\boldsymbol{G}_2(\boldsymbol{x}_2)$: nonlinear mapping matrix

$\boldsymbol{W}_2$: load adjust matrix

$\boldsymbol{x}_2$: control input matrix

$\boldsymbol{x}_2 = [e_1\ e_2\ \dot{e}_1\ \dot{e}_2]^T$

Fig. 4 demonstrates the structure of the proposed FB-GS compensator. Fig. 4(1) is the fuzzy neural network architecture, Fig. 4(2) and (3) are the membership functions, and Fig. 4(4) indicates the overall FN structure. The fuzzy neural network has 2 inputs, 4 layers, and 1 output. The input layer of layer A is the input signal for displacement error signal e_p and the velocity error signal of the displacement differentiation. This input signal is distributed to the next layer's unit. Next, the intermediate layer of layer B holds the membership functions in the internal functions and employs a Gauss function. The output of each unit is a value of the membership function and is referred to as the antecedent. C layer is the consequent deduced with the fuzzy rule. Layer B is divided into 9 input spaces and the degree of compatibility of the divided spaces is calculated (equation (18) is the antecedent degree of compatibility for the fuzzy rule). The standardized value μ_i in equation (19) is produced as the aggregate of the antecedent degree of compatibility that can be gained from all units in layer C. Layer D is the linear unit (equation (21) and (22)) that produces aggregate input.

Given the above, it is possible to create a neural network that has a fuzzy-rule architecture, called fuzzy neural network. This fuzzy neural network can learn using BP, and by devising a hierarchical neural network connection it can be given a correspondence relationship with a fuzzy reasoning rule.

Output u_{FN} is the direct input of the plant. In the fuzzy portion the input space is divided into 9 parts, and the degree of compatibility of the divided space can be applied with the equation below.

$$\mu_i = A_{i_1}(e_p)A_{i_2}(e_v) \quad i=1, 2,\ldots, 9, \quad i_1, i_2=1, 2, 3 \tag{18}$$

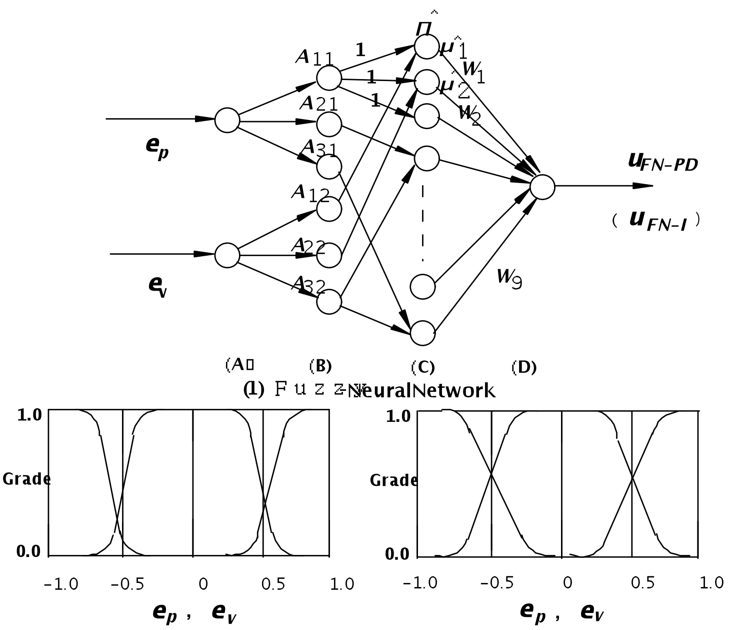
A_{ij} expresses a fuzzy variable with a membership function.

A_{i1} and A_{i2} each express the degree of compatibility for Positive big, Small, and Negative big.

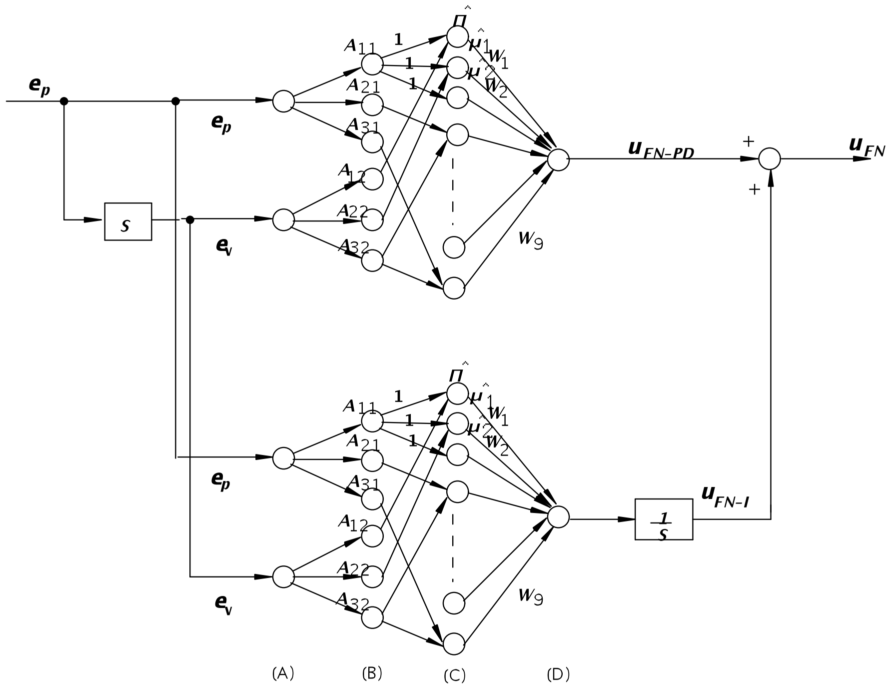
However, the degree of compatibility is normalized as follows.

$$\bar{\mu}_i = \frac{\mu_i}{\sum_k \mu_k} \tag{19}$$

To adaptively adjust the control gain, a fuzzy rule is established that adjusts output in response to volume of erroneous input.



(2) □□□□□□□□□□ □□□□□□□□ (FN-PD)(3) □□□□□□□□□□ □□□□□□□□ (FN-



(4) Architecture of the fuzzy-neural network

Fig. 4 Fuzzy neural network gain scheduling compensator

Table 1 Control rules

		e_v		
		Positive big	Small	Negative big
e_p	Positive big	P	PD	P
	Small	PD	PID	PD
	Negative big	P	PD	P

The fuzzy rule for i is:

R^i: IF e_p is A_{i_1} and e_v is A_{i_2} THEN $y = f_i(e_p,\ e_v)$ (20)

$u_{FN\text{-}PD}$ output is obtained as

$$u_{FN-PD} = \sum_{i=1}^{9} \bar{\mu}_i f_{i-PD}(e_p, e_v) \tag{21}$$

Here, $f_{i\text{-}PD}$ is

$$f_{i-PD} = k_{ip} e_p + k_{iv} e_v \tag{22}$$

$u_{FN\text{-}PI}$ output is obtained as

$$u_{FN-PI} = \sum_{i=1}^{9} \bar{\mu}_i f_{i-PI}(e_p, e_v) \tag{23}$$

When $f_{i\text{-}PI}$ is

$$f_{i-PI} = k_{iI} \int e_p dt + k_{ip} e_p \tag{24}$$

u_{FN} additional output $u_{FN\text{-}PID}$ output is

$$u_{FN-PID} = \sum_{i=1}^{9} \bar{\mu}_i f_{i-PD}(e_p, e_v) + \sum_{i=1}^{9} \bar{\mu}_i f_{i-PI}(e_p, e_v) \tag{25}$$

The deduction is obtained as

$$u_{FN} = \sum_{i=1}^{9} \bar{\mu}_i f_i(e_p, e_v) \tag{26}$$

When $u_{FN\text{-}PID}$ and u_{FN} are equal

f becomes

$$f_i = k_{iI} \int e_p dt + 2k_{ip} e_p + k_{iv} e_v \tag{27}$$

However, $k_{ij} \geq 0 \ (j = I, p, v)$ is the feedback gain.

The control method in equation (27) carries out an operation that switches fuzzy PD and fuzzy PI according to error response. To respond to impacts on the control system due to disturbance and parameter variables by immediately approximating the target value, as shown in Table 1 PID control is used when both e_p and e_v are small. When $|e_p|$ is big, not when e_p is small, damping is reduced and P control is used to gain a quick response. In other circumstances PD control is used. Each feedback gain is previously learned in the antecedent neural network. A control law is constructed with this type of structure. This composition becomes the control law adapted to errors.

In the neural network for the FN-GS compensator, feedback gain learning takes place to minimize the secondary position error.

$$E_p = 1/2 e_p^{\,2} \tag{28}$$

In addition, to continuously fulfill the condition $k_{ij} \geq 0 (j = I, p, v)$, the following is made true.

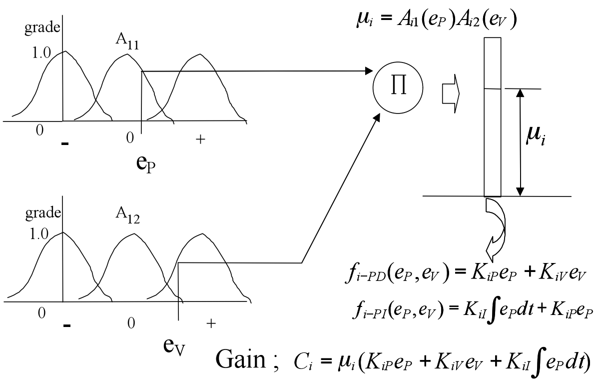


Fig. 5 Scheme diagram of FN-GS gain calculation

$$k_{ij} = w_{ij} \tag{29}$$

Fig. 4(1) and (4) specify in what layer of fuzzy neural network the processing expressed by equations (18)–(29) takes place.

SIMULATION

Using the designed multivariable controller the disturbance suppression and decoupling of the proposed control method were verified through simulation. The numerical simulation used an integral step size of 1[ms], a sampling duration of 2[ms], and a simulation period of 10[s] and was conducted with a sine-wave signal of 1[rad/s] and a x_d=1deg. target value.

The simulation parameters in Table 2 change for each of the circumstances in cases 1–3.

Using the above settings, the controller C(s) in the block diagram for the FN-GS gain controller shown in Fig. 2 can be applied as follows.

$$C(s) = \sum_{i=1}^{9} C_i(s) = \sum_{i=1}^{9} \mu_i (k_{ip} + k_{iv} s + k_{iI} \tfrac{1}{s}) \tag{30}$$

The FN-GS gain shown in Fig. 5 is calculated, and just as the scheme diagram for the FN-GS gain matrix shows, the k_{ij} gain is optimally adjusted to obtain inverse dynamics of the target trajectory error signal by learning the fuzzy PID gain.

Three simulations were conducted to demonstrate the effectiveness of the established system. Furthermore, the step response to the PID control and proposed robust gains scheduling are compared.

Case 1 Fig. 6 indicates response according to the signal from target signal 1rad/s, and reveals that the position error of the proposed control is reduced through variable gain.

Case 2 The simulation was carried out with a sampling time of 2[ms], simulation time of 10[s], a sine-wave signal of 1[rad/s], and a target value of 1[deg]. However, in contrast to link 1 a 90-degree phase was given to link 2. Fig. 7 shows simulation results. The trajectory of link 1 was not impacted by the interference of link 2, and the decoupling of the system is apparent.

Case 3 The responsiveness of the proposed method and the conventional method were compared against a 0.5[rad] step response. The results are shown in Fig. 8. The proposed method had a favorable response for start-up characteristics, demonstrating that an adaptive optimal gain was realized against conventional PID control and model-referenced gains scheduling.

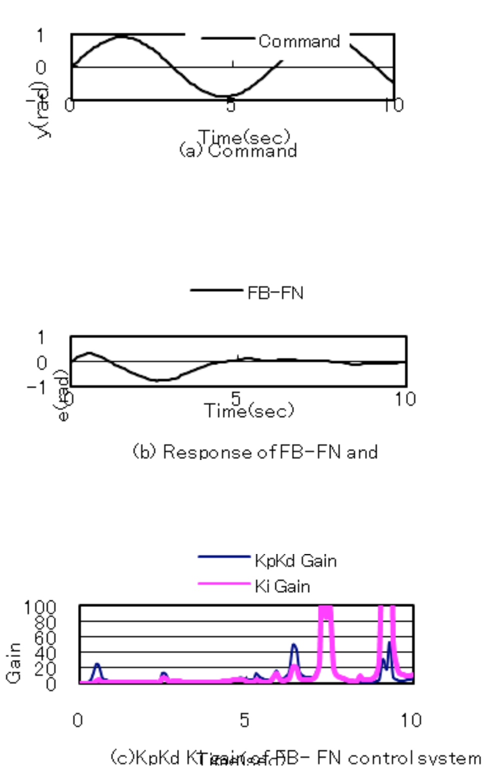
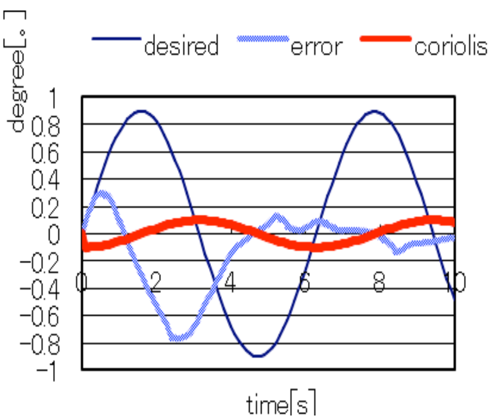


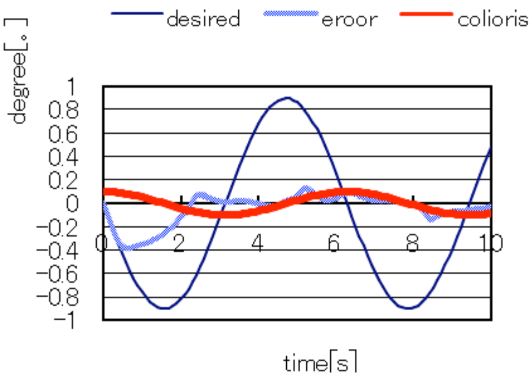
Fig. 6 Responses on condition of 1rad/s

Table 2 Simulation parameters

Notation	Value
Kp	10
Ka	1.2
Kv	2.5
Vc	± 12.0[V]
J	0.149[kgm^2]
D_d	2[deg]
F_c	1.96X10^{-2}[Nm]
C_q	1.5X10^{-3}[Nm/s^{2}rad]
D_z	10[deg]
Nois	0.05[rad]
Y_θ	1.0[rad]
F_S	3X10^{-2}[Nm]

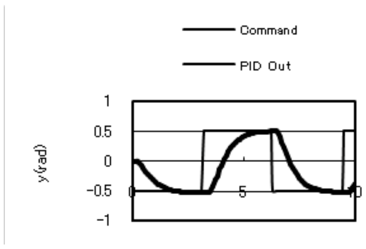


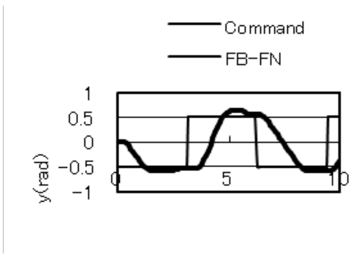
(a) Response of rink 1



(b) Response of rink 2

Fig. 7 Response of rink





(a) Step response of PID control (b) Step response of FN control

Fig. 8 Response for FN-GS control on condition of step disturbance

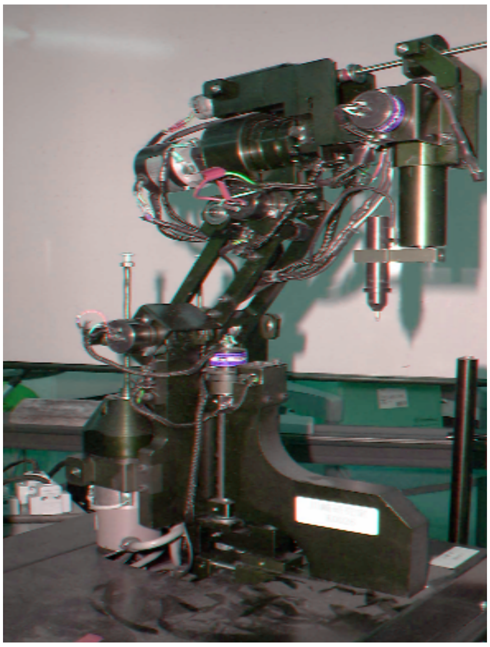


Fig. 9 Robot Manipulator

EXPERIMENT

To confirm the effectiveness of the proposed control method, an experiment was carried out using only the first joint of a small,6-axis robot as a single-degree-of-freedom system.

Each parameter utilized with a control algorithm was as follows.

A sampling time of Ts=2ms, a velocity gain of Kv=3, NN learning frequency N=5000, F-N learning frequency N=50, and target trajectory signal T(t) were set at a sine-wave of 1Hz and amplitude of 30deg. Fig. 10 reveals that the phase delay is improved, demonstrating that NN effectively follows the target trajectory. However, near the peak friction impact produces stationary error.

On the other hand, as seen by the response signal in Fig. 11, it was confirmed that the nonlinear stationary error compensation sufficiently performed using a fuzzy neural network. Deviation signals produced by elements such as model error and friction were adaptively reduced through the feed-forward NN compensator.

Furthermore, Fig. 12 illustrates the fuzzy neural network output when a disturbance was added in the form of a 1kg weight was placed on the tip of the robot. In other words, it marks the reply signal that displayed changes influenced by gravity. An F-NN output signal was created to follow the target signal based on disturbance. Results show verification of disturbance suppression and decoupling in response to disturbances.

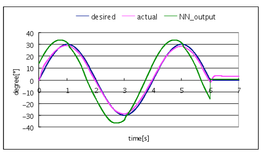


Fig. 10 Response of N-N

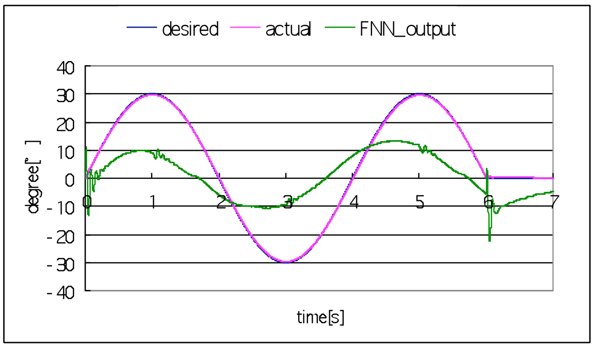


Fig. 11 Response of F-N

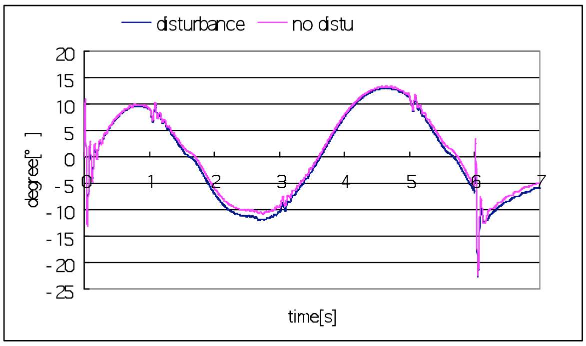


Fig. 12 Response of F-N control to disturbance

CONCLUSION

This paper proposes a multivariable control system design for manipulators using a fuzzy neural network. The method suggested here is a multivariable expansion of the author's proposed method[12]. Linearization was carried out with a neural network by directly using a nonlinear manipulator equation of motion. The control employs a fuzzy neural network to suppress disturbance and decouple. A control rule where the gain adaptively changes with time was used so that nonlinear systems that change parameters according to time continuously carry out critical damping, thereby accomplishing decoupling of mutual interference for each joint in the multivariable system. In other words, it was possible to design a constant oscillation rate for the entire oscillation at all times and achieve perfect tracking control of the manipulator.

To verify the effectiveness and feasibility of the proposed method, a DC motor was used as an actuator and to serve as a nonlinear controlled object. A decelerating mechanism determined the positioning that drives the manipulator, and simulations and an experiment were carried out on its application to servo systems.

The above results demonstrate the effectiveness and feasibility of the proposed control approach. There are plans to continue research on load deflection and cooperative control.

REFERENCES

[1] Astrom, K. J., and Hagglund, T. (1988). *Automatic Tuning of PID Controllers*, Instrument Society of America.
[2] Craig,J. J. (1988). *Adaptive Control of Mechanical Manipulators*., Addison-Wesley Publishing.
[3] Freund, E.(1977).A Nonlinear Control Concept for Computer-Controlled Manipulator, *Proc. IFAC Symp. Multivariable Technological System*, 395/403.
[4] Freund, E. (1985).Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators, *Int.J. Robotics Research*, 6-2, 16/27.
[5] Khosla, P. K., and Kanade, T.(1986). Real-Time Implementation and Evaluation of Model-Based Controls on CMU DD Arm Ⅱ, *Proc. IEEE Conf. Robotics and Automation*, 1546/1555.
[6] Miyazaki, T., andOishi, K.(1997). Design Method of a Robust, Stable 2-Degrees-of-Freedom System Using Disturbance Observer and Coprime Factorization, *IEEJ Journal Transactions on Industry Applications*, 117(9):572–578.
[7] Miyazaki, T., and Oishi, K. (1998).Robust Control Based on Disturbance Observer and Polynominal Coefficient Approach for Mechanical System, *IEEJ Journal Transactions on Industry Applications*,118(6): 707–714.
[8] OishiK., Fujita, M., and Miyazaki, M. (1992). A Motion Control Manipulator that Uses Only Positional Information, *Journal of the Robotics Society of Japan*, 10(6): 796–802.
[9] Oishi K., Onishi M., and Miyazaki, T.(1996). Multiple Joint Robot Motion Control Method Based on a Robust Speed Controller,*IEEJ Journal Transactions on Industry Applications*, 116(2): 144–151.
[10] Sato, Y. (2007). Design of a Robust Gains Scheduling Control System Using a Fuzzy Neural Network, *IEEJ*, 127(8): 1221–1227.

[11] Sato, Y.(2003). Robust Fuzzy Neural Network Based Control in a Mechatronic Servo System (Initial Report: Verification through simulation &experiment), *Journal of the Japan Society of Mechanical Engineers*, 69C-687: 2929–2936.
[12] Sato, Y.(2004). Robust Fuzzy Neural Network Based Control in a Mechatronic Servo System (2nd Report: Verification of system stability & robustness),*Journal of the Japan Society of Mechanical Engineers*, 70C-700: 78–85.