한국퍼지 및 지능 시스템학회 논문지 1998, Vol. 8, No. 2. (1998)50~59

Adaptive Control of a Flexible Single Link Robot Manipulator using Self-Recurrent Neural Networks

자기순환 신경망을 이용한 유연성 단일 링크 로봇 매니퓰레이터의 적응제어

Han-Ho Tack · Yeon-Gyu Choo

Department of Electronic Eng.,

Chinju National University

ABSTRACT

This paper presents stabilization and adaptive control of flexible single link robot manipulator system by self-recurrent neural networks that is one of the neural networks and is effective in nonlinear control. The architecture of neural networks is a modified model of self-recurrent structure which has a hidden layer. The self-recurrent neural networks can be used to approximate any continuous function to any desired degree of accuracy and the weights are updated by feedback-error learning algorithm. When a flexible manipulator is rotated by a motor through the fixed end, transverse vibration may occur. The motor torque should be controlled in such a way that the motor rotates by a specified angle, while simultaneously stabilizing vibration of the flexible manipulators so that it is arrested as soon as possible at the end of rotation. Accurate vibration control of lightweight manipulator during the large changes in configuration common to robotic tasks requires dynamic models that describe both the rigid body motions, as well as the flexural vibrations. Therefore, a dynamic models for a flexible single link robot manipulator is derived, and then a comparative analysis was made with linear controller through an simulation and experiment. The results are presented to illustrate the advantages and improved performance of the proposed adaptive control over the conventional linear controller.

1. Introduction

Today most of the robots often used in industrial fields for automatization and higher efficiency have rigid bodies and thick, heavy manipulators. The bad things about these robots are that they take a lot of space and that actuators require more motion energe, which in turn makes it difficult to promote motion speed. Now a lot of researches are under way to complement such weaknesses by reducing the weight and making the link more flexible[1-5].

On the other hand the flexibility of the link causes vibration in motion: therefore more precise position control is required for efficient vibration control, for which more exact dynamics equation and efficient control algorithm are indispensable. Y. Sakaw[1] and Z. H. Luo[2] applied optimal control to flexible link and demonstrated its utility. K. S. Yeung[3] applied variable structure control method and did the same. However linear optimal control or linear feedback control is robust within limited linear areas enough for systems to be sensitive to them when unexpected disturbances happen. Accordingly the regulation of system parameters is necessary in case system environment is unknown or uncertain.

Recently to cope with the control problems a lot of experts are conducting a study of control methods to which neural networks are applied, as opposed to existing mathematical analytical ones, on parallel process, learning, nonlinear mapping, pattern recognition, information process, industrial application, etc.[7-10]. Besides neural networks are characterized by nonlinearity, learning ability and optimalizing ability, and by making practical use of these characteristics and appling them to nonlinear control, adaptive control, etc. good results have been achieved[8, 9].

This study focuses on how to use self-recurrent neural networks based on adaptive controller[12] and bring under active control the position of flexible single link robot manipulators. The structure of neural networks has one hidden layer with self-recurrent architecture. And the initial weights are given by random number, the learning of weights was made use of feedback errors learning algorithm by simultaneous learning and control on behalf of on-line system. In addition to netural networks controller is carried on as on-line learning by means of desired trajectory in repetition, feedback errors were backpropagated through the neural networks. The purpose of learning in on-line is to lessen nonlinear errors and expand the operational region of adaptive control system with the reasonable control parameter against the nonlinear plant modification and external disturbance without teaching signal.

Also, generalized feedback-error learning algorithm is used for the learning of the self-recurrent neural networks based on adaptive controller. The kinetic equation of flexible link was derived using an assumed mode method and a Lagrange equation. To demonstrate the efficiency of the self-recurrent neural networks control algorithm presented in this study, a neural networks controller based adaptive controller was designed and then a comparative analysis was made with linear controller through an simulation and experiment

2.Self-recurrent neural networks

The arbitrary nonlinear function using neural networks is to approximation by accurately state. With the objective of a simple self-recurrent neuron at hidden layer and a shorter training time for neural networks model, neural networks, as shown in Fig. 1, is developed. Consider Fig. 1, where for each discrete time k, $I_i(k)$ is the i th input, $S_j(k)$ is the sum of inputs to the j th self-recurrent neuron, $X_j(k)$ is the output of the j th self-recurrent neuron, $O(k)$ is the output of the output layer, z^{-1} is delay time, and ● is self-recurrent neuron. Depending on the network, W^I W^O, or W^D represents hidden layer, output, or weight vectors of self-recurrent layer, respectively.The mathematical model for the output value and the weights update of the neural networks in Fig.1 is shown below:

$$O(k)=\sum_j W_j^o X_j(k)+\theta_k \quad (1)$$

$$X_j(k)=f(s_j(k))+\theta_j \quad (2)$$

$$S_j(k)=W_j^D X_j(k-1)+\sum_{i=1}^{n} W_{ij}^I I_i(k) \quad (3)$$

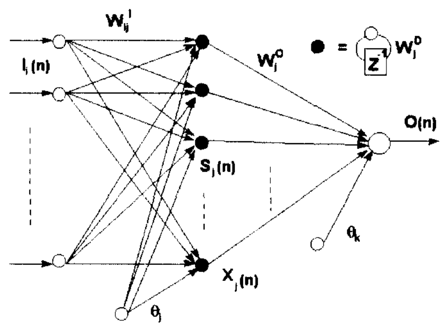


Fig. 1. Self-recurrent neural networks architecture.

where f(·) is the usual sigmoid function. Then error function for feedback-error learning algorithm for neural networks can be defined as

$$E(k)=\frac{1}{2}\sum_{k=1}^{n}(r(k)-y(k))^2 \quad (4)$$

where $r(k)$ is the desired response of plant, $y(k)$ is the actual response of plant. In general, the plant response is a nonlinear mapping G(·) of input $u(k)$, i.e., $y(k)$ = $G(u(i),\ i\le k\ )$.

$$\frac{\partial E(k)}{\partial W}=-e(k)\frac{\partial y(k)}{\partial W}=-e(k)\frac{\partial u(k)}{\partial W}$$

$$= -e(k)\frac{\partial O(k)}{\partial W} \quad (5)$$

where $e(k)$ is the error between the desired and output response of the plant. Given the self-recurrent neural networks shown in Fig. 1 and described by (1), (2), and (3), the output gradients with respect to output, recurrent, and input weights, respectively, are given by

$$\frac{\partial O(k)}{\partial W_j^o} = X_j(k) \quad (6)$$

$$\frac{\partial O(k)}{\partial W_j^D} = W_j^o P_j(k) \quad (7)$$

$$\frac{\partial O(k)}{\partial W_{ij}^I} = W_j^o P_{ij}(k) \quad (8)$$

and satisfy

$$P_j(k) = f'(S_j)(X_j(k-1) + W_j^D P_j(k-1)), \quad (9)$$
$$P_j(0) = 0$$

$$Q_{ij}(k) = f'(S_j)(I_j(k) + W_j^D Q_{ij}(k-1)), \quad (10)$$
$$Q_{ij}(0) = 0$$

The weights can now be adjusted following a gradient method, i.e., the update rule of the weights becomes

$$W(n+1) = W(n) + \eta(-\frac{\partial E(k)}{\partial W}) \quad (11)$$

where η is a learning rate, and $E(k)$ represents error function.

3. Modeling of flexible manipulator

Fig. 1 shows the manipulator system that has a flexible single link. I_h means the mass moment of inertia of the hub referring to the actuator operating the manipulator plus the part fixing the link to the actuator: M_e, the mass of end effect plus payload at the end-point of the link and J_e, its mass moment of inertia: g, the gravity heading below z axis. Flexible link were modeled on the thin long uniform

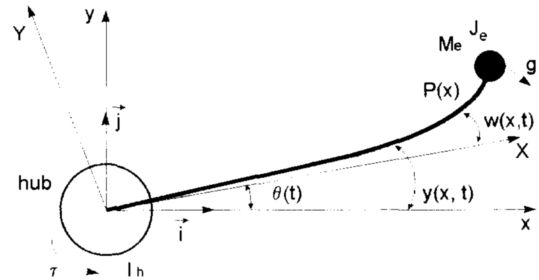


Fig. 2 Flexible single link robot manipulator.

Bernoulli-Euller beam which has such great length compared with a cross-section area that shear deformation and rotary inertia effects can be neglected. In the operation they don't show any distortion and it is assumed that they have only bending deformation. The friction of joints and the structural damping of flexible link are neglected in system modelling. The differential equation of motion should be satisfactory to the flexible link with free transverse vibration and the equation is as follows[13]:

$$\rho A(x)\frac{\partial^2 w(x,t)}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left[EI(x)\frac{\partial^2 w(x,t)}{\partial x^2}\right] = 0 \quad (12)$$

In this equation, ρ is linear mass density, A(x) the cross-section area of the link and EI(x) the bending stiffness of the link. w(x,t) the elastic displacement to x arbitrary point of the link is the linear combination of assumed mode shapes and generalized coordinates, which is approached as follows[12]:

$$w(x,t) = \sum_{i=1}^{n}\phi_i(x)q_i(t) = \phi q^T \quad (13)$$

where $\phi_i(x)$, the i th assumed mode shape of the link, is clamped-free eigenfunction, $q_i(t)$, the i th generalized coordinate which is the time function corresponding to $\phi_i(x)$. The link has a moment of inertia I_h and a length l. The angular displacement of the link is denoted as $\theta(t)$, so the total deflection $y(x,t)$ can be represented as

$$y(x,t) = x\theta(t) + w(x,t) \quad (14)$$

The elastic displacement on oprate plane of the system is being approximated by the assumed modes. The assumed modes is the assumed modes with function of space coodinates and the linear combination of generalized coordinates with time function, which is approached as (13). In addition, comparative function which can be satisfied with both the geomatrical boundary conditions that it has mass in end-point of the clamped-free link and the natural boundary condition simultaneously is used for the assumed modes shape. At the clamped end (x=0), when the zero conditon of deflection and gradient from hub the geomatrical boundary conditions given by bellow.

1) $w(0,t) = w_0 = 0$: The deflection must be zero.

2) $\frac{\partial w(0,t)}{\partial x} = 0$: The slope of the deflection curve must be zero.

At the free end ($x=l$), the shearing force and the bending moment by balance condition the natural boundary condition given by bellow.

$$1)EI\frac{\partial^2 w(l,t)}{\partial x^2} + J_e\left[\ddot{\theta} + \frac{\partial \ddot{\omega}(l,t)}{\partial x}\right] = 0$$

: The bending moment must be zero.

$$2)EI\frac{\partial^3 w(l,t)}{\partial x^3} + M_e\left[-\ddot{\omega}(l,t) + \dot{\theta}^2\omega(l,t) - \ddot{\theta}l + g\sin\theta\right] = 0$$

: The shearing force is zero.

The above boundary condition yield the transcendental equation for the natural frequency βl .

$$\cos(\beta l)\cos(\beta l) + 1 = 0 \qquad (15)$$

From (13), the eigenfunction $\phi(x)$ can be

$$\phi(x) = c_1\sin\beta x + c_2\cos\beta x + c_3\sinh\beta x + c_4\cosh\beta x \qquad (16)$$

represented as

where c_i is an arbitrary constant, Substituting boundary conditions, and reform for c_1 , c_2, c_3, and c_4

Kinetic energy come from rotary motion of hub, alignment or rotary motion of flexible link and that of the mass of end point. It is sums to total kinetic energy as following.

And potential energy in system is the very elastic

$$K = \frac{1}{2}I_h\dot{\theta}^2 + \frac{1}{2}\rho A\int_0^l\left[\dot{\theta}^2x^2 + \dot{w}^2 + 2\dot{\theta}\dot{w}x + \dot{\theta}^2w^2\right]dx + \frac{1}{2}J_e\left[\dot{\theta}^2 + 2\dot{\theta}\dot{w}_l' + \dot{w}_l'^2\right] + \frac{1}{2}M_e\left[\dot{w}_l^2 + \dot{\theta}^2 w_l^2 + 2\dot{\theta}w_l l + \dot{\theta}^2l^2\right] \qquad (17)$$

displacement of flexible link, if it was irrespective of the mass of end-point and gravity of flexible link, the above energy can be demonstrated as follows.

The equation of motion can apply the Euler-Lagrange equation.

$$V = \frac{1}{2}EI\int_0^l(\frac{\partial^2 w}{\partial x^2})^2dx + \int_0^l\rho Ag[w\sin\theta - x\cos\theta]dx + M_e g[w\sin\theta - x\cos\theta] \qquad (18)$$

where is input torque for drive of flexible link

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0, \quad i = 1,2,\ldots,n \qquad (19)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = \tau \qquad (20\ )$$

from actuator, and L = K - V. Substituting (17) and (18) into (19) and (20) gives, without regard to the mass of end-point of flexible link, the mass of inertia and gravity, applied partial differential to generalized coordinates. The kinetic equation of discreted differential equation type is given by

$$\ddot{\theta} = \frac{\tau + \sum_{j=1}^{n} q_i\omega_j^2\int_0^x\phi_i xdm}{I_h} \qquad (21)$$

$$\ddot{q}_i = -\frac{T}{I_h}\int_0^x\phi_i xdm - q_i w_i^2[1$$

$$+\frac{(\int_0^x \phi_i xdm)^2}{I_h} \;]$$

$$-\sum_{j\neq i}^{n} \frac{q_j \omega_j^2 \int_0^x \phi_i xdm \int_0^x \phi_j xdm}{I_h} \qquad (22)$$

4. Design of control system

In this section, a feedback-error learning scheme that consists of a linear feedback controller and the neural networks as a feedforward inverse controller, as shown in Fig. 3, is discussed. The transfer function of system at open loop analysis of flexible link is shown below.

$$G(s)=\frac{53.3s^3+63.96s^2+4774s+5692}{s^4+1432s^2} \qquad (23)$$

At this time, because the pole exists in all imaginary axis, this system will vibrate under the unstability. To ensure unstable system to stability, if zero point was added to transfer function equation, pole point in imaginary axis moves to left half surface, can attain safe system. In this Fig. 3, the neural networks is configured in parallel to a conventional feedback controller and the network is trained on-line by repeating the desired trajectory cycles where the feedback-error is backpropagated through the network. Convergence is achieved when the neural networks has learned the inverse of the plant and it then takes control of the plant and eliminates the significance of the feedback controller. This scheme consists of a fixed gain linear feedback controller that makes the overall system stable, and a feedforward controller which updates its internal weights to generate the control signal $u_{nn}(k)$ in the process of becoming an inverse model of the plant. networks. During the initial training period, the control signal $u_{nn}(k)$ was very insignificant. The control signal from the feedback controller, $u_c(k)$, was significant because of the large initial error. Hence, in the early stage of learning, the component $u_c(k)$ was dominant over the $u_{nn}(k)$. However, as the learning

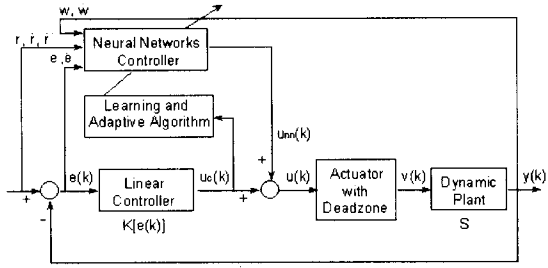


Fig. 3 Total control system using neural

trials increased $u_{nn}(k)$ become dominant over $u_c(k)$. In general, the feedback-error learning has the following advantages: (i) a teaching signal is not required to train the neural networks, instead, the error signal is used as the training signal, (ii) the learning and control are performed simultaneously in sharp contrast to the conventional 'learn-then-control' approach. In Fig. 3, S represents a dynamic plant, $K[e(k)]$ is a linear function of the error and the derivative of error representing a linear control law. The dynamics of the overall system shown in Fig. 3 are described by the following equations.

$$e(k)=r(k)-y(k) \qquad (23)$$

where $r(k)$ is the desired trajectory,

$$K[e(k)]=K_p e(k)+K_d \dot{e}(k) \qquad (24)$$

$$u(k)=u_{nn}(k)+u_c(k) \qquad (25)$$

$$y(k)=Su(k) \qquad (26)$$

In this Fig. 3, the characteristics of the actuator with deadzone are described by the function causes method

$$D[u]=\begin{bmatrix} u(k)-d & if & u(k)>d, \\ 0 & if & -d\le u(k)\le d, \\ u(k)+d & if & u(k)<d \end{bmatrix} \qquad (27)$$

where the parameter d represents the width of the deadzone, and even if it cannot be used in simulation, at real experiment if large input disposed in plant, it can bad influence on total system, plant input can be controlled by the addition of actuator with deadzone. At this experiment, controller was utilized by limit switch, and system operation on program will be stopped. The control signal to the plant is then given by

$$v(k) = D[u(k)] \qquad (28)$$

The neural networks controller, once trained, will represent the inverse dynamics model of the dynamic plant. The fixed gain linear controller ensures adequate performance prior to the convergence of the neural networks controller parameters, and reduces the steady-state output errors due to the disturbance inputs. In essence, the output of the linear controller is an indication of the mis-match between the dynamics of the plant and the inverse dynamics model obtained by the neural networks controller This is because if the true inverse dynamics model has been learned, the neural networks controller alone will provide the necessary control signal to achieve the desired trajectory. With zero trajectory error, the linear controller produces no output and, hence, indicates that learning has completed. The neural networks controller is a three layer network that has 7 input neurons, 10 neurons in the hidden layer and 1 output neuron. The activation function used for the input and hidden layers is the tan-sigmoid function and for the output layer is a linear function. The inputs to the neural networks are $r(k)$, $\dot{r}(k)$, $\ddot{r}(k)$, $e(k)$, $\dot{e}(k)$ $w(k)$, and $\dot{w}(k)$ and the output of the neural networks is the control action $u_{nn}(k)$

5. Experimental results

To demonstrate the efficiency of the self-recurrent neural networks control algorithm presented in this study, a neural networks controller based adaptive controller was designed and then a comparative analysis was made with linear controller(PD controller) through an simulation and experiment, and the formation of the experimental system of the controller is shown in Fig. 4. The simulation were performed using MATLAB software package.

In Fig. 4, the camera is equipped with linearlly arranged 2048 analog photo detectors, each of which is 13[μm] wide. The whole length of the arranged detectors is 26.624mm. The light of LED is projected on the arranged detectors through the lense. It in turn is changed into voltage distribution and computed through the encoder. For the motor PMI DD motor with 3000rpm and servo amplifier were used. The system parameters used in this study are the length of the link l=1.2[m], the width of the link w=0.0254[m], the thickness of the link D=0.0032[m], mass density rho ρ =0.2332[kg/m^2], coefficiency of elasticity EI=6.715[$N \cdot m^2$], hub inertia I_h =0.017[kg/m^2], the matrial of the hub is aluminum, and the end-point payload of the link not considered.

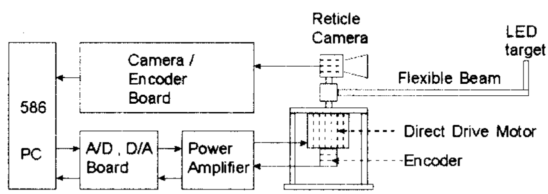
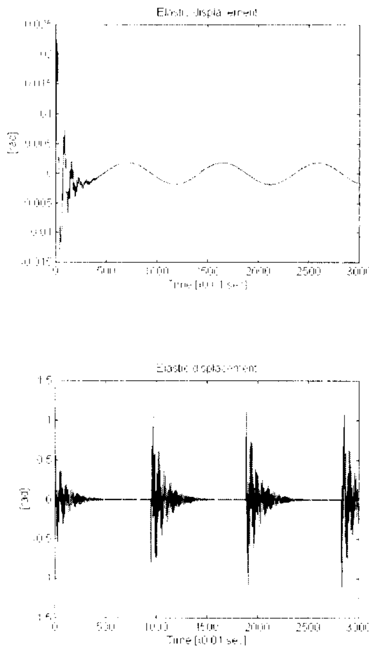


Fig. 4 Configuration of experimental.

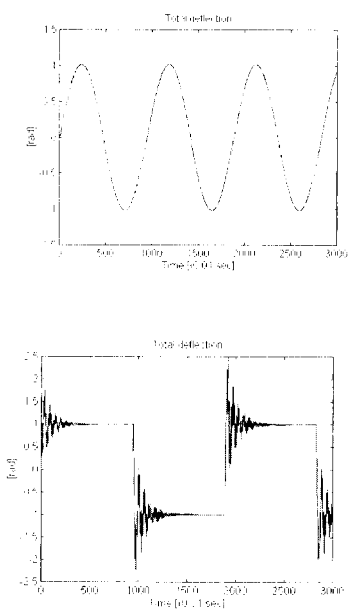
The resonance ferquency of the link are omega w_1 = 1.38 [rad/s] and w_2 = 8.71 [rad/s] The initial value of the weight to the neural networks controller was made random numbers between -1 and 1: η the initial learning rates of the neural networks controller, 0.094. K_1 and K_2, which are gain constants, were made 0.1 and 5.9 respectively: bias weights, between -1 and 1. In the structure presented in this paper, after making lots of information on the flexible link modeling, and then which is controlled as a inverse dynamics and neural networks made a compensation for as much as the amount of error in the practical environment in the presence of errors of modeling, the mapping region is not large. As in spite of neural networks with the structure of multi input-output, limiting early weights to very small area, that we have made learning more faster. If we control it with neural networks controller only without linear controller, the mapping area increases, so the learning of neural networks becomes difficult and makes divergence.

The reference input used in this simulation is two types, which used sin(t) as a position control as to trajectory tracking, and $\pm 1[rad]$ step as a position, we observed the response as to total deflection and elastic displacement in output. Fig. 5 shows the response for PD controller and Fig. 6 is the response for proposed neural networks controller. As you know from the Fig., though both of the two controller is well carried on, the proposed neural networks controller with the function of learning has learned the optimal trajectory on the given trajectory compared with linear controller as well as showed the superior ability to cope with elastic displacement and robust qualites. Fig. 7 shows the results of practical experimental though PD controller and neural networks controller. The experiment in this paper was made in the way of changing the initial position of the flexible single link robot manipulator from 0° to $\pm 10^{\circ}$ the angle of the link position of the system output which is wanted and making a analysis of the total deflection and elastic displacement of the link for PD controller and neural network controller in moving. Fig. 7(a), (b)



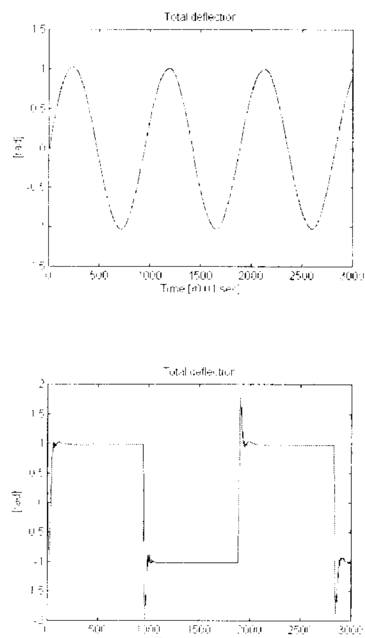
(b) Elastic displacement.

Fig. 5 Simulation response of PD controller.

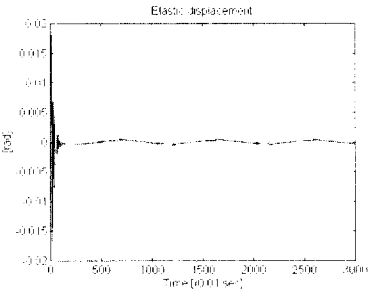


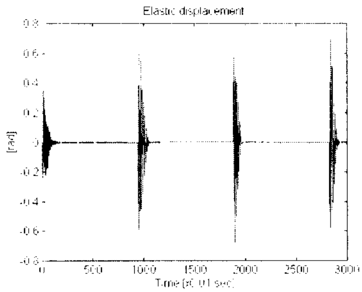
(a) Total deflection.

Fig. 5 Simulation response of PD controller(continue)



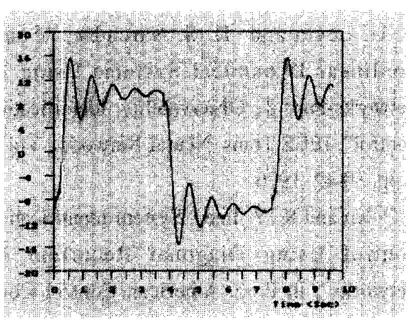
(a) Total deflection.



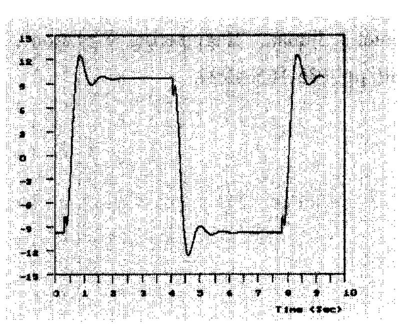


(b) Elastic displacement.

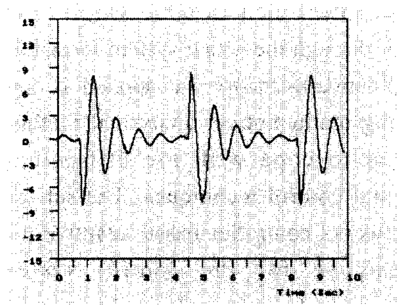
Fig. 6 Simulation response of neural networks controller.



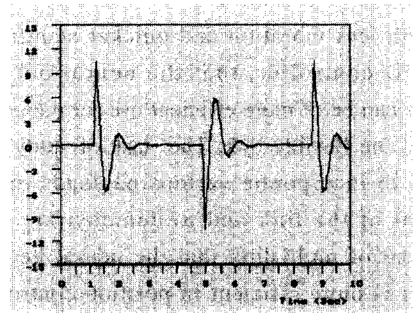
(a) Total deflection(PD controller)



(b) Total deflection(NN controller)



(c) Elastic displacement(PD controller)



(d) Elastic displacement(NN controller)

Fig. 7 Experimental response

represents whole total deflection(deg) which is shown in vertical axis, and Fig. 7(c), (d) is elastic displacement which is shown in vertical axis(deg). Knowing from the Fig., we can see the satisfactory efficiency of reference input compared with results of simulation, the learning ability of neural networks controller can see exact position control according to various sudden change of paremeter of plant. In this results, the dynamic structure of neural networks controller is made to be an approximate inverse model of the plant under cotrol thereby achieving almost unity mapping between the input and output signal space. The learning and adaptive algorithm decides the number of controller modules to be activated to match the order of the dynamic system under control.

6. Conclution

In this study, an experiment was made of the adaptive control of a flexible single link robot manipultor through the actual system formation where the neural networks theory was applied as opposed to the existing mathematical analytical methods. The purpose is to cope with the difficulty of the conventional control techniques. The self-recurrent neural networks controller based adaptive controller were designed for the control system, the feedback-error learning algorithm was used to improve the weight of the other. The result of the experiment was that neural network controller, a nonlinear adaptive controller can make any desired angle with less vibration and quicker stabilization than a PD controller, that the neural networks controller can be of more efficient use for the position control of the flexible link. The task of study in the future is to incorporate various payloads into the end-point of the link and to demonstrate in the experiment of multi-link that the neural networks controller is more efficient in position control than other conventional ones.

References

[1] Y. Sakawa, F. Matsuno, and S. Fukushima, "Modeling and Feedback Control of A Flexible Arm", Journal of Robotic System, Vol. 2, No. 4, pp. 453-472, 1985.

[2] Z. H. Luo, "Direct Strain Feedback Control of Flexible Robot Arms: New Theoretical and Experimental results", IEEE Tran. on Automati -c Control. Vol. 38, No. 11, pp. 1610-1622, 1993.

[3] S. Nicosia, P. Tomei, and A. Tornambe, "Non-Linear Control and Observation Algorithm for A Single-Link Flexible Robot Arm", Int. J. Control, Vol. 49, No. 3, pp. 827-840, 1989.

[4] O. Morgul, "Orientation and Stablization of A Flexible Beam Attached to a Rigid Body: Planar Motion", IEEE Tran. on Automatic Control. Vol. 38, No. 11, pp. 953-962, 1991.

[5] K. S. Yeung and Y. P. Chen, "Sliding modecontroller design of a single-link flexible manipulator under gravity", Int. J. Control, Vol. 52, No. 1, pp. 101-117, 1990.

[6] K. G. Shin and X. Cui, "Design of a Knowledge-Based Controller for Intelligent Control System," IEEE Trans. Syst., Man, Cybe., Vol. 21, No. 2, pp. 368-375, 1991.

[7] X. Cui and K. G. Shin, "Intelligent Coordination of Multiple Systems with Neural Networks", IEEE Trans. Syst., Man, Cybern., Vol. 21, No 6, pp. 1488-1497, 1991.

[8] M. Sekiguchi, T. Sugasaka and S. Nagata,"Control of Multivariable System by a Neural Network", IEEE International Conf. on Robotics and Automation, pp. 2644-2649, 1991.

[9] K. S. Narendra and K. Parthasarathy, "Identific-ation and Control of Dynamical SystemsUsing Neural Networks", IEEE Trans. Neural Networks, Vol. 1, No. 1, pp. 4-27, 1990.

[10] A. U. Levin and K. S. Narendra, "Control of Nonlinear Dynamical Systems Using Neural Networks-Part Ⅱ: Observability, Identification, and Control", IEEE Trans. Neural Networks, Vol. 7, No. 1, pp. 30-42, 1996.

[11] C. C. Ku and K. Y. Lee, "System Identificati-on and Control Using Diagonal Recurrent Neural Networks", in Proc. American Control Conf., pp. 545-549, 1992.

[12] Leonard Meirovitch, "Analystical Methods in Vibrations", Macmillan Publishing Co, pp. 126-166, 1996.

[13] Daniel J. Inman, "Engineering Vibration", Prentice Hall, pp. 126-166, 1994.

탁 한 호 (Han-Ho Tack) 종신회원
1987년 2월 : 부경대학교 전자공학과 졸업
1992년 2월 : 동아대학교 전자공학과 졸업(석사)
1998년 2월 : 한국해양대학교 전자통신공학과 졸업(박사)
1987년 1월 ~ 1989년 9월 : (주)홍창 부설 연구소 연구원
1991년 ~ 현재 : 진주산업대학교 전자공학과 조교수

추 연 규 (Yeong-Gyu Choo) 정회원
1988년 2월 : 부경대학교 전자공학과 졸업
1991년 2월 : 동아대학교 전자공학과 졸업(석사)
1997년 2월 : 동아대학교 전자공학과 졸업(박사)
1997년 3월 ~ 현재 : 진주산업대학교 전자공학과 전임강사