Vol. 3, No. 4 / December 1999
pp. 331 ~ 338

Structural Vibration Control Using Artificial Neural Networks

By Ju-Tae Kim*, Ju-Won Oh and In-Won Lee*****

Abstract

Structural control using neural networks is presented. A cost function comprised of control energy and controlled response is defined to train controller neural network. In the model based optimal control, external disturbances and non-linearities cannot be considered. However these limitations can be overcome in the response based neuro-control algorithm. Not only linear structure but nonlinear structure can be easily controlled via proposed technique. Numerical example shows that the controller can suppress the vibration induced by not only trained earthquakes but also untrained ones.

Keywords : Vibration control, Neural networks, Training, Cost function.

1. Introduction

Artificial neural networks (ANNs) have been widely used in the field of structural engineering in recent years. Especially, vibration control using neural networks has been a new research topic for structural control engineers during last decade. Some characteristics of neural networks appealing to control engineers are non-linearity, parallelism, and learning capability. Although modern control theories have been well established in electrical engineering, they cannot be directly applied to civil structures due to some problems such as non-linearity, uncertainty and time-varying properties in them. These problems necessarily make neural networks a promising tool for the control of civil structures.

Pioneering studies by H. M. Chen *et al.* (1995) and J. Ghaboussi *et al.* (1995) show that neural networks can be successfully applied to the control of large civil structures. The vibration of nonlinear structures showing hysteretic behavior has also been controlled via nonlinearly trained neural networks (K. Bani-Hani *et al.* 1998).

In their studies, controller neural networks, so called neuro-controller, are trained via certain criteria. H. M. Chen *et al.* defined instantaneous error function as the summation of error between actual and desired responses. Then, training rule of neuro-controller is derived by minimizing the error function. They set the desired response in cost function to be zero, which means that structural responses should disappear in one time step. J. Ghaboussi et *al.*, however, used algorithm which aims the average of expected responses for a few future time steps to be zero. This scheme is for the smooth reduction of structural responses. Although the desired response can be set by some strategy, the selection of desired response is not straightforward, and it may not be

*Graduate Student, Department of Civil Engineering, Korea Advanced Institute of Science and Technology, Science Town, Taejon, 305-701, Korea
**Professor, Department of Civil and Environmental Engineering, Hannam, University, 300-791, Taejon, Korea.
***Professor, Department of Civil Engineering, Korea Advanced Institute of Science and Technology, Science Town, Taejon, 305-701, Korea.

The manuscript for this paper was submitted for review on June 24, 1999.

optimal. To avoid this problem, a new algorithm which does not require desired response is needed.

To this end, a cost function is defined in this study. Then, training rule for neuro-controller is derived by minimizing the cost function. There is no need to set the desired response at each training step. The controller trained by this scheme can be said to be optimal neuro-controller because the cost function is minimized after training. The advantages of optimal neuro-controller can be summarized as: 1) it can be simply applied to non-linear structures; 2) it can consider external excitations such as earthquake ground motions. In conventional optimal control, linearization procedure is needed for the control of non-linear structure (J. N. Yang 1994). Moreover external disturbance cannot be considered in the design of optimal controller.

In numerical examples, the earthquake-induced vibrations of both a linear and a nonlinear structure are controlled by neural networks trained via predefined cost function. Results show that structural vibration can be reduced successfully.

2. Learning Algorithm for Controller Neural Network

Block diagram for the control of structural vibration is shown in Fig. 1. Emulator neural network (ANN 1) is first trained to simulate the structural response for the same input signal as applied to the structure. Then it is used to obtain sensitivity information of the structural responses. Controller neural network (ANN 2) is then trained to suppress undesired vibration induced by external disturbances such as wind and earthquakes. The training rule updates the weights of the controller with the help of the information on the sensitivity of the structural responses to control force and control force itself. Weight updating criterion is to reduce the cost function defined by Eq. (1).

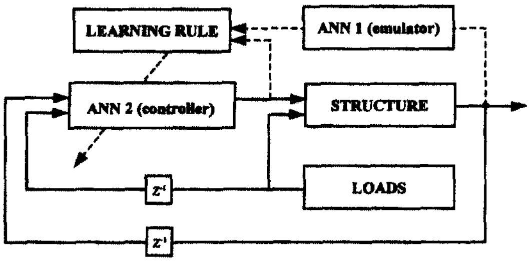


Fig. 1. Block diagram for structural control using neural network

$$J_{\infty} = \frac{1}{2}\int_0^{\infty}(\bar{x}^T Q \bar{x} + \bar{u}^T R \bar{u})dt \tag{1}$$

where $\bar{x}, \bar{u}$ are state vector, control force vector respectively, and Q, R are weighting matrices. The discrete form of Eq. (1) can be written as Eq. (2) where Δt is time increment for analysis and control and T_f is the total time step considered.

$$\hat{J}_{T_f} = \sum_{n=0}^{T_f} \hat{J}_n = \sum_{n=0}^{T_f} \frac{1}{2}\{\bar{x}(n)^T Q \bar{x}(n) + \bar{u}(n)^T R \bar{u}(n)\}\Delta t \tag{2}$$

Because the state, $\bar{x}$ and control force, $\bar{u}$ are the implicit and the explicit function of the weight of the controller neural network, $\hat{J}_{T_f}$ can be minimized through appropriate weight modification. There are two types of learning mode. One is pattern learning and the other is batch learning. Weights are updated at each time step in pattern learning mode. Therefore instantaneous cost function, $\hat{J}_n$, is minimized. Weights are updated once for all time steps in batch learning. Global cost function, $\hat{J}_{T_f}$, is minimized in this mode. The former mode is related to instantaneous optimal control and the latter scheme to conventional optimal control. In this paper, pattern learning mode is used for weight updates. Although $\hat{J}_n$ is minimized in pattern learning mode, $\hat{J}_{T_f}$ can be minimized. This is shown in numerical examples.

Neural network model with one hidden layer is

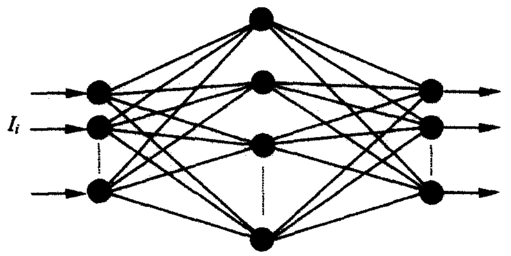


Fig. 2. Neural network with one hidden layer

shown in Fig. 2. Inputs to the neural network are feedback information including delayed signals of structural response and ground acceleration. Outputs are the control forces. Weight between the *k-th* output node and the *j-th* hidden node is denoted by W_{kj} and between the *j-th* hidden node and the *i-th* input node is denoted by V_{ji}. Only one hidden layer is used here for the derivation of weight update equation. But the number of hidden layers is not restricted to one. One can easily extend the following equations to the case of more than 2 hidden layers. Weight update equation between output and hidden layer can be expressed as Eq. (3) by steepest descent rule.

$$\Delta W_{kj} = -\eta\frac{\partial \hat{J}_n}{\partial W_{kj}}$$
$$= -\eta\left(\left\langle\frac{\partial \hat{J}_n}{\partial \bar{x}}\right\rangle\left[\frac{\partial \bar{x}}{\partial \bar{u}}\right]\left\{\frac{\partial \bar{u}}{\partial W_{kj}}\right\} + \left\langle\frac{\partial \hat{J}_n}{\partial \bar{u}}\right\rangle\left\{\frac{\partial \bar{u}}{\partial W_{kj}}\right\}\right)$$
$$= -\eta\Delta t\left(\bar{x}^T Q\left[\frac{\partial \bar{x}}{\partial \bar{u}}\right] + \bar{u}^T R\right)\left\{\frac{\partial \bar{u}}{\partial W_{kj}}\right\} \quad (3)$$

where η is learning rate. The elements of response sensitivity matrix can be obtained from the dynamics of the structure, or in the case of a structure with unknown dynamics, from Eq. (4) in which the response, x_s, is obtained by emulator neural network.

$$\frac{\partial x_s}{\partial u_k} \cong \frac{x_s(u_k + \Delta u_k) - x_s(u_k)}{\Delta u_k} \quad (4)$$

Since the weight, W_{kj} affects only the output u_k, Eq. (3) is further simplied to Eq. (5).

$$\Delta W_{kj} = -\eta\Delta t\left(\bar{x}^T Q\left\{\frac{\partial \bar{x}}{\partial u_k}\right\} + r_k u_k\right)\frac{\partial u_k}{\partial W_{kj}}$$

$$= \eta\Delta t\left(\bar{x}^T Q\left\{\frac{\partial \bar{x}}{\partial u_k}\right\} + r_k u_k\right)g'(net_k)\cdot f(net_j) \quad (5)$$

where

$$net_k = \sum_j f(net_j)W_{kj}, \quad (6)$$

$$net_j = \sum_i I_i V_{ji} \quad (7)$$

$f(\cdot)$: activation function of hidden layer

$g(\cdot)$: activation function of output layer

r_k : the *k-th* diagonal element of R.

By introducing generalized error of output layer, δ_k, Eq. (5) can be written as

$$\Delta W_{kj}=\eta\delta_k f(net_j) \quad (8)$$

where

$$\delta_k = -\Delta t\left(\bar{x}^T Q\left\{\frac{\partial \bar{x}}{\partial u_k}\right\} + r_k u_k\right)g'(net_k), \quad (9)$$

The same rule is applied to the weights between hidden and input layer to produce update equation as

$$\Delta V_{ji} = -\eta\frac{\partial \hat{J}_n}{\partial V_{ji}}$$
$$= -\eta\left(\left\langle\frac{\partial \hat{J}_n}{\partial \bar{x}}\right\rangle\left[\frac{\partial \bar{x}}{\partial \bar{u}}\right]\left\{\frac{\partial \bar{u}}{\partial V_{ji}}\right\} + \left\langle\frac{\partial \hat{J}_n}{\partial \bar{u}}\right\rangle\left\{\frac{\partial \bar{u}}{\partial V_{ji}}\right\}\right) \quad (10)$$

where

$$= -\eta\Delta t\left(\bar{x}^T Q\left[\frac{\partial \bar{x}}{\partial \bar{u}}\right] + \bar{u}^T R\right)\left\{\frac{\partial \bar{u}}{\partial V_{ji}}\right\}$$

$$\left\{\frac{\partial \bar{u}}{\partial V_{ji}}\right\} = \left\{\begin{matrix} \frac{\partial u_1}{\partial net_1}\ \frac{\partial net_1}{\partial net_j}\ \frac{\partial net_j}{\partial V_{ji}} \\ \vdots \\ \frac{\partial u_k}{\partial net_k}\ \frac{\partial net_k}{\partial net_j}\ \frac{\partial net_j}{\partial V_{ji}} \\ \vdots \\ \frac{\partial u_N}{\partial net_N}\ \frac{\partial net_N}{\partial net_j}\ \frac{\partial net_j}{\partial V_{ji}} \end{matrix}\right\}$$

$$= \left\{\begin{matrix} g'(net_1)\ W_{1j}\ f(net_j)I_i \\ \vdots \\ g'(net_k)\ W_{kj}\ f(net_j)I_i \\ \vdots \\ g'(net_N)\ W_{Nj}\ f(net_j)I_i \end{matrix}\right\}. \quad (11)$$

Eq. (10) can be further simplified to Eq. (12) by introducing generalized error of hidden layer, δ_j.

$$\Delta V_{ji} = \eta\delta_j I_i \quad (12)$$

where

$$\delta_j = -\Delta t\bar{x}^T Q\left[\frac{\partial \bar{x}}{\partial \bar{u}}\right] + \bar{u}^T R\left\{\begin{matrix} g'(net_1)\ W_{1j}\ f(net_j) \\ \vdots \\ g'(net_k)\ W_{kj}\ f(net_j) \\ \vdots \\ g'(net_N)\ W_{Nj}\ f(net_j) \end{matrix}\right\}. \quad (13)$$

Finally the training procedure of neuro-controller is summarized as follows.

Step 1 : Initialize weights, set target cost(TC).
Step 2 : Set cost function, $\hat{J}_{T_f}$ to zero and let n=1.
Step 3 : Feed delayed signals of state and ground acceleration as input signals to neural network.
Step 4 : Calculate network output, $f(net_j)$, j=1, 2, ..., M and $g(net_k)$, k=1,2,..., N.
Step 5 : Apply control force u_k, k=1,2,..., N to structure and obtain responses.
Step 6 : Calculate cost function, $\hat{J}_n$ and $\hat{J}_{T_f} = \hat{J}_{T_f} + \hat{J}_n$.
Step 7 : Calculate response sensitivity, $\left[\frac{\partial x}{\partial u}\right]$.
Step 8 : Calculate δ_k, k = 1,2,...N and ΔW_{kj}, k = 1, 2,...,N, j=1,2,...,M.
Step 9 : Calculate δ_j, j = 1,2,...M and ΔW_{ji}, j = 1, 2,...,M, i = 1,2,...,L.
Step 10 : Update weights, $W_{kj} \leftarrow W_{kj}+\Delta W_{kj}$, $W_{ji} \leftarrow W_{ji}+\Delta W_{ji}$ and n=n+1.
Step 11 : If n < T_f then go to Step 3, else go to next step.
Step 12 : If $\hat{J}_{T_f}$ > TC then go to Step 2, else stop.

3. Control Of Linear SDOF Structure

Let's consider single degree of freedom structure excited by earthquake ground motion. The equation of motion of the system can be written as

$$m\ddot{y} + c\dot{y} + ky = -m\ddot{y}_g + u \tag{14}$$

where m, c, and k are mass, damping and stiffness of the structure. Ground acceleration is denoted by $\ddot{y}_g$, and u is control force produced by neuro-controller. Eq. (14) can be reformulated to state space form as Eq. (15).

$$\begin{Bmatrix} \dot{y} \\ \ddot{y} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{Bmatrix} y \\ \dot{y} \end{Bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} u + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \ddot{y}_g \tag{15}$$

If we introduce state variable $\bar{x} = \{y\dot{y}\}^T$, Eq. (15) is simplified to Eq. (16).

$$\dot{\bar{x}} = A\bar{x} + Bu + F\ddot{y}_g \tag{16}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1/m \end{bmatrix},\ F = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{17-19}$$

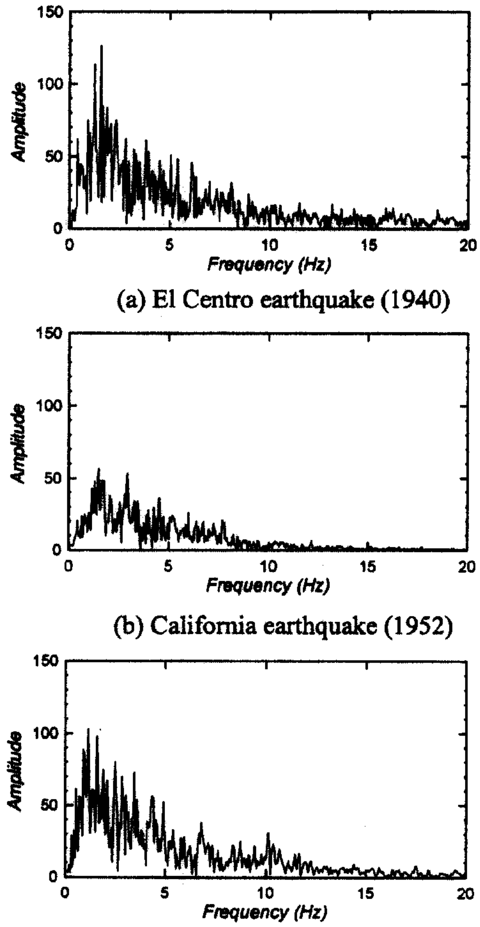
To derive weight update equation of neuro-controller, let's define weighting matrices of cost function as

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & q \end{bmatrix},\ R = [r] \tag{20, 21}$$

Then, the increment of the weights between output and hidden layer can be written as

$$\Delta W_{1j} = \eta\delta_1 f(net_j) \tag{22}$$

where



(a) El Centro earthquake (1940)

(b) California earthquake (1952)

(c) Northridge earthquake (1994)

Fig. 3. Frequency components of ground motions.

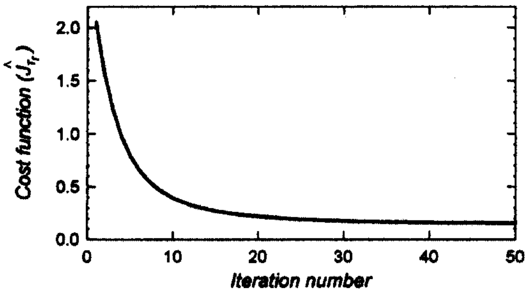


Fig. 4. Learning history (linear case)

$$\delta_1 = -\Delta t\left(\bar{x}^T Q\left\{\frac{\partial \bar{x}}{\partial \bar{u}}\right\} + ru\right)g'(net_1)$$

$$= -\Delta t\left(y\frac{\partial y}{\partial u} + q\dot{y}\frac{\partial \dot{y}}{\partial u} + ru\right)g'(net_1) \qquad (23)$$

The subscript 1's of Eq. (22) and (23) mean that there is only one node in the output layer of controller neural network.

The increment of the weights between hidden and input layer can be written as

$$\Delta V_{ji} = \eta \delta_j I_i \qquad (24)$$

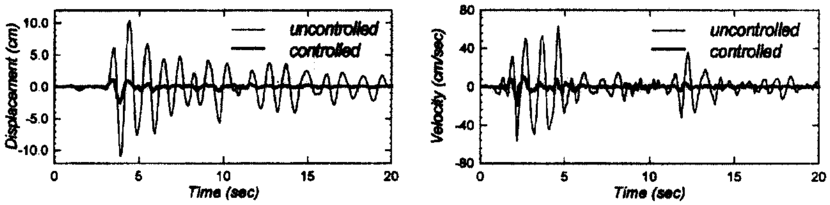
where

$$\delta_j = -\Delta t(\bar{x}^T Q\left[\frac{\partial \bar{x}}{\partial \bar{u}}\right] + ru)g'(net_1)W_{1j}f'(net_j)$$

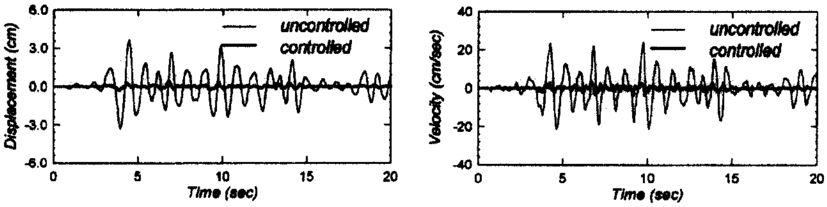
$$= -\Delta t(y\frac{\partial y}{\partial u} + q\dot{y}\frac{\partial \dot{y}}{\partial u} + ru)g'(net_1)W_{1j}f'(net_j) \qquad (25)$$

For numerical simulation mass(1 kg), damping (1.25 N/m/sec) and stiffness(39 N/m) values are set. El Centro earthquake(1940) is used for the training of controller. Then, two more ground motions, California earthquake(1952) and Northridge earthquake (1994), are used for test. Fig. 3 shows the frequency components of three ground motions. The inputs to the controller neural network are delayed signals of structural displacement, velocity and ground acceleration, namely $y(n-1)$, $\dot{y}(n-1)$ and $\ddot{y}_g(n-1)$. The number of nodes in hidden layer is four. The output is control force. Fig. 4 shows learning history of the global cost function, $\hat{J}_{T_f}$. Although the controller is trained in pattern learning mode which minimizes instantaneous cost function $\hat{J}_n$, the global cost function, $\hat{J}_{T_f}$, is minimized.

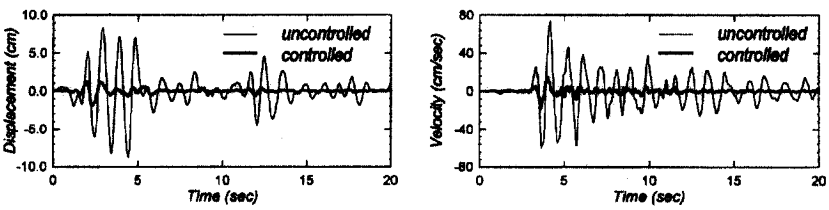
Controlled and uncontrolled responses are shown



(a) El Centro earthquake



(b) California earthquake



(c) Northridge earthquake

Fig. 5. Uncontrolled and controlled responses of linear structure.

in Fig. 5. Neuro-controller trained by El Centro earthquake can also reduce vibrations induced by other earthquakes which are not trained. This is one of the important characteristics of neural networks.

4. Control of Nonlinear Structure

To simulate control effect on nonlinear structure, the following nonlinear equation of motion is considered.

$$m\ddot{y} + c\dot{y} + s(y, \dot{y}) = -m\ddot{y}_g + u \quad (26)$$

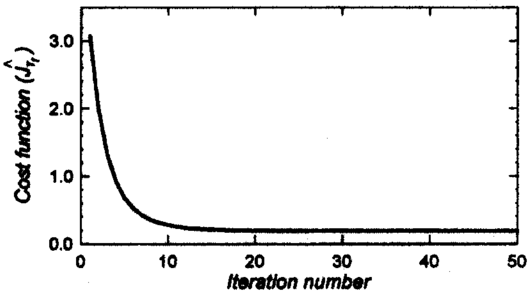


Fig. 6. Learning history (non-linear case)

where m and c are mass and damping respectively. The nonlinear stiffness function, $s(y, \dot{y})$, is defined by

$$s(y, \dot{y}) = aky + (1 - \alpha)kdw \quad (27)$$

where and are positive numbers. The variablein the above equation is governed by the following differential equation.

$$\dot{w} = \frac{1}{d}(a\dot{y} - \beta|\dot{y}||w|^{p-1}w - \gamma\dot{y}|w|^{p}) \quad (28)$$

where a, β, γ are positive numbers and p is an odd number. This model can simulate nonlinear hysteretic behavior. Detailed description of this model can be seen in the reference [H. Irschik *et al.* 1998].

Each parameter is set for numerical simulation. Namely, a=0.6, β =0.5, γ =0.5, a=1.0, d=0.04 and p=5. The system parameters, m, c, and k are the same as in the linear model. Fig. 6 shows learning history of cost function for nonlinear structural control. Convergence is almost the same as in the case of linear structure. Fig. 7 shows that neuro-control-

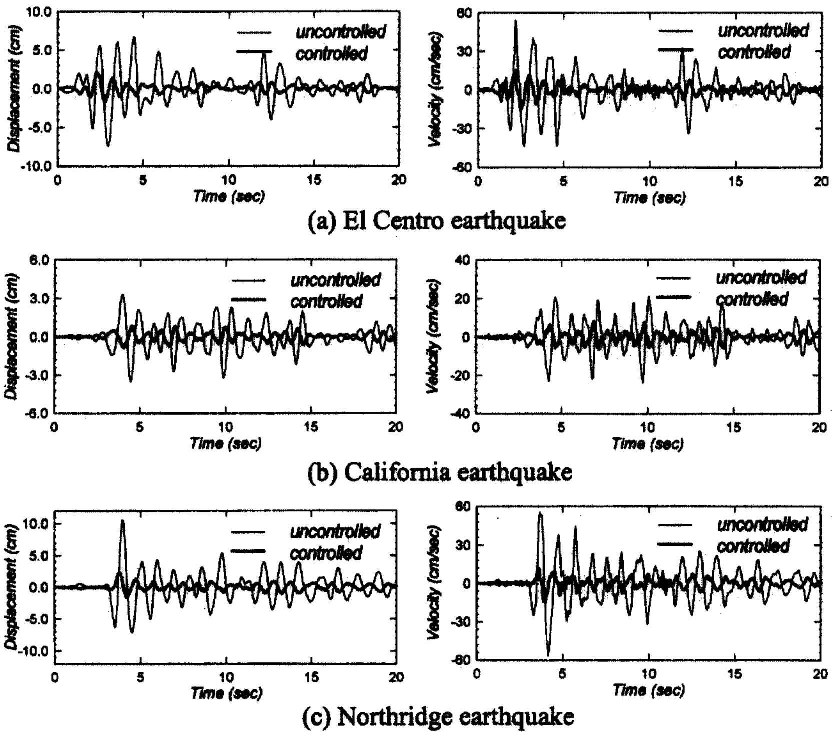
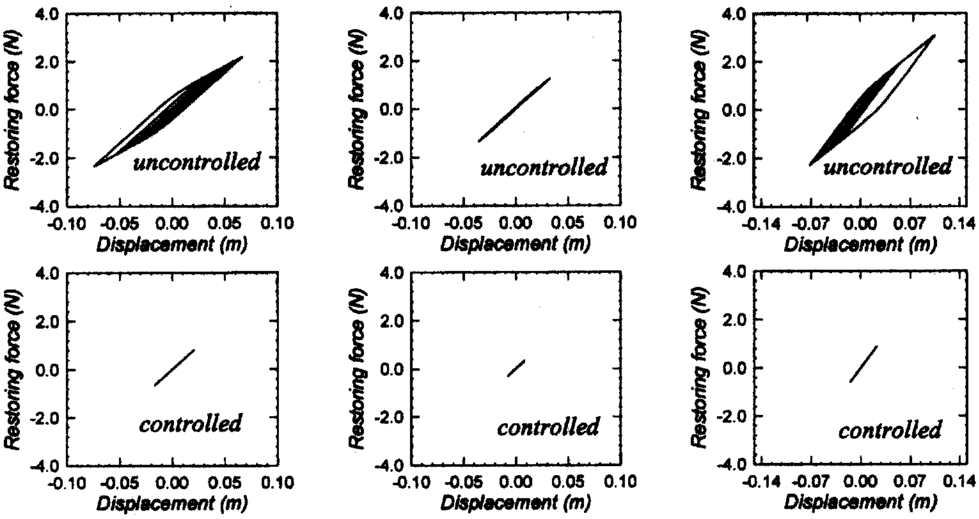


Fig. 7. Uncontrolled and controlled responses of nonlinear structure.



(a) El Centro earthquake (b) California earthquake (c) Northridge earthquake

Fig. 8. Restoring force vs. displacement

ler can also control the vibrations of nonlinear structure. Fig. 8 shows the relationship between restoring force and displacement. Uncontrolled responses show bilinear hysteretic behavior while controlled responses show linear behavior with amplitudes being reduced.

In the model based conventional optimal control, linearization procedure is required to derive control law of nonlinear structure. But there is no need to linearize the structural parameter in the response based optimal neuro-control scheme because learning algorithm is independent of whether the structural is linear or nonlinear. In addition, external disturbance such as earthquake is neglected in the model based optimal control. Therefore it may not be optimal when external disturbance exists. But neuro-controller learns to be optimal when disturbance exists.

5. Conclusions

A new learning algorithm for neuro-controller is derived. There is no need to set desired response to derive learning rule for neuro-controller. Learning rule is extracted through minimization of instantaneous cost function, but it can reduce global cost function. Three main advantages of structural control using neural network can be summarized as follows. First, we can control the structure of unknown dynamics through the learning of responses itself. Second, it can be easily applied to the control of nonlinear structure without linearization of structural parameters. Third, external disturbance can be considered in the optimal control.

In simulation study, vibration of linear structure can be reduced successfully. Nonlinear structure having hysteretic characteristics behaves linearly under control action. Only displacement and velocity is considered in this study. However, acceleration can also be controlled if acceleration term is included in the cost function contain acceleration term. Future works should include the control of multi degree of freedom structure, time delay effect and the saturation of control force.

References

1. Bani-Hani, K. and Ghaboussi, J. (1998). "Nonlinear Structural Control Using Neural Networks," *Journal of Engineering Mechanics, ASCE*, Vol. 124, No. 3, pp. 319-327.
2. Chen, H. M., Tsai, K. H., Qi, G. Z., Yang, J. C. and Amini, F. (1995). "Neural Networks for Structure Control," *Journal of Computing in Civil Engineering, ASCE*, Vol. 9, No. 2, pp. 168-176.

3. Ghaboussi, J., and Joghataie, A. (1995). "Active Control of Structures Using Neural Networks," *Journal of Engineering Mechanics, ASCE*, Vol. 121, No. 4, pp. 555-567.
4. He, Y., Wu, J. (1998). "Control of Structural Seismic Response by Self-recurrent Neural Networks (SRNN)," *Earthquake Engineering and Structural Dynamics*, Vol. 27, pp. 641-648.
5. Irschik, H., Schlacher, K. and Kugi, A. (1998). "Control of Earthquake Excited Nonlinear Structures Using Liapunov's Theory," *Computers & Structures*, Vol. 67, pp. 83-90.
6. Nguyen, D. H. and Widrow. B. (1990). "Neural Networks for Self-learning Control Systems," *IEEE Control Systems Magazine*, pp. 18-23.
7. Tang,Y. (1996). "Active Control of SDF Systems Using Artificial Neural Networks," *Computers & Structures*, Vol. 60, No. 5, pp. 695-703.
8. Venini, P. (1998). "Robust Control of Uncertain Structures," *Computers & Structures*, Vol. 67, pp. 165-174.
9. Yang, J. N., Li, Z., and Vongchavalitkul, S. (1994). "Generalization of Optimal Control Theory : linear and nonlinear control," *Journal of Engineering Mechanics, ASCE*, Vol. 120, No. 2, pp. 266-283.