Active Control of Wind-Induced Vibrations in Tall Buildings using Neural Networks

Allan C. Nerves and R. Krishnan

The Bradley Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0111, U.S.A.
Phone:(703)231-4311 Fax:(703)231-3362
e-mail: kramu@vtvm1.cc.vt.edu

***Abstract* - A neural network is proposed in the paper to provide active control to a building-tuned mass damper system using direct control by a modified back-propagation scheme. The specialized learning architecture which allows the neural controller to learn to control the system interactively and autonomously while it is in operation is applied in this paper to overcome the effects of wind induced oscillations in tall civil structures. Simulation results are presented to study the properties of the proposed scheme and it is found that these results are encouraging for possible control implementations.**

I. INTRODUCTION

Dynamic loads that act on large civil structures can be classified into two main types : environmental, such as wind, wave, and earthquake loads ; and man-made, such as vehicular and pedestrian traffic and those caused by reciprocating and rotating machineries. The response of these structures to dynamic loads will depend on the intensity and duration of the excitation, the structural system, and the ability of the structural system to dissipate the excitation's energy. The shape of the structure also has a significant effect on the loading and resulting response from wind excitation.

The advent of high-strength, lighter, and more flexible construction materials has created a new generation of tall buildings. Due to the smaller amount of damping provided by these modern structures, large deflection and acceleration responses result when they are subjected to environmental loads. Such large responses, in turn, can cause human discomfort or illness and, sometimes, unsafe conditions. Passive, semi-active, and active vibration control schemes are becoming an integral part of the structural system of the next generation of tall buildings [1],[2],[3].

It has been shown in field studies that tall buildings that are subjected to wind-induced oscillations usually oscillate at the fundamental frequency of the building. In some cases this is coupled with torsional motion when the torsional and lateral oscillation frequencies are close. One of the most common control schemes used to correct these oscillations is a tuned mass damper(TMD) system. Basically, a TMD consists of a mass attached to a building, such that it oscillates at the same frequency as the structure but with a phase shift. The mass is attached to the building via a spring-dashpot system and the energy is dissipated by the dashpot as relative motion develops between the mass and the structure.

The objective of this study is to design an active controller for a tuned mass damper system which is being used to control the first fundamental mode of a building's vibration due to wind excitation. In designing an active control system to control building vibration due to wind excitation, several schemes can be used, depending on the way the measured information is utilized. One option is to use closed-loop control where the control force is determined by the measured response of the building, such as displacement or velocity. Another option is to use open-loop control where the control force is determined by the measured wind excitation. A third option is to use closed-open-loop control where the control force is determined by both the measured response of the building and the measured wind excitation. For this study, only the closed-loop control scheme is considered.

One of the important problems in control systems is the inverse problem of determining the input to produce a desired output. The design of a controller based on a linear approximation of the plant characteristics has certain inescapable disadvantages : (1) The model becomes computationally prohibitive for complex and highly dynamic plants, (2) the control quality is very sensitive to the quality of the linear approximation, and (3) the model has difficulty in adapting to time-varying plant parameters. Therefore, an adaptive controller that learns to control the system interactively and autonomously while in operation is required. Because of their ability to learn actual relevant control and plant data, artificial neural networks have become a promising approach to controlling complicated, multidimensional and dynamic systems [4].

Therefore, the objective of this study is to design a robust and easy-to-use neural controller for the active control of wind-induced vibrations in tall buildings. A specialized learning architecture is chosen[5]. The back-propagation algorithm [4],[5],[6] is used to train the network because it requires minimum knowledge of the plant characteristics. A proportional controller designed using classical methods is also presented for comparison.

The paper is organized as follows. Section II contains the methodology for training a multi-layer neural network using a modified back-propagation algorithm which includes the specialized learning feature. A comparison of various learning architectures is also given. Modeling of the building-TMD system is given in section III. The simulation results of the proposed neural network controller is given in section IV. Section V contains the conclusions of the study. Nomenclature is placed in section VI for easy reference.

0-7803-1328-3/94$03.00© 1994 IEEE

II. METHODOLOGY

A. Learning Architecture

Learning control is a control method which determines the inputs required to control or improve the transient behavior of a dynamic system, given that prior knowledge of the controlled system may not be available. The problem is to develop a learning control algorithm which will always produce a smaller error on successive trials. One of the possible solutions to the learning control problem is to use artificial neural networks to determine the control inputs [7],[8].

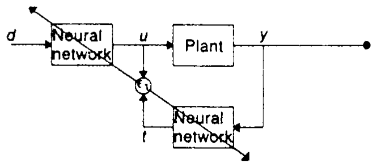

Fig. 2.1 Indirect Learning Architecture

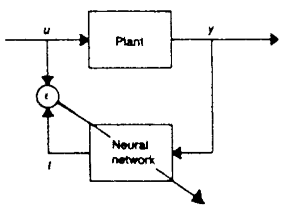

Fig. 2.2 Generalized Learning Architecture

Three learning control architectures for neural networks have been proposed by Psaltis, Sideris, and Yamamura [5] and by Nguyen and Widrow [9]. In the indirect learning (IL) architecture (see Fig. 2.1)[5], the desired plant output d is propagated through a neural network to produce the plant input u. The actual plant output y is then used as input to a copy of the controlling network. The difference between the outputs of the controlling network and its copy is the indirect or training error which is used to adapt both networks to drive this error to zero. A disadvantage of this architecture is that the networks tend to settle to a solution that maps all desired responses to a single plant input. This input is in turn mapped by the plant to an output that gives zero training error but obviously a non-zero total error. To overcome this problem, the generalized learning (GL) architecture (see Fig. 2.2)[5],[9] trains the neural network by using different u's as inputs to the plant and teaches the neural controller to map the corresponding output y's back to the u's. Disadvantages of this learning strategy are : (1) The controller is inoperative during the learning phase, (2) it cannot delimit its operating range to the relevant one, and (3) the plant is assumed to be a static physical process.

The above disadvantages can be avoided by using the specialized learning (SL) architecture (see Fig. 2.3)[5]. Training involves using the desired responses d's as input to the network. The network then learns to find the plant inputs u's that drive the system outputs y's to the desired d's by using the error between the desired and actual responses of the plant. Thus the network (1) specifically learns in the region of interest, and (2) may be trained on-line, fine-tuning itself while performing its function. As a result, the controller learns continuously and is able to control plants with time-varying characteristics. Therefore, a specialized learning architecture is used in this present study.

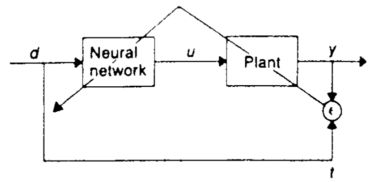

Fig. 2.3 Specialized Learning Architecture

B. Training Algorithm

The back-propagation algorithm cannot be directly applied to the specialized learning architecture because of the location of the plant. Psaltis et. al. [5] proposed that the plant can be thought of as an additional, although unmodifiable layer (see Fig. 2.4)[4]. Therefore, the error signal at the system output could be propagated back through the plant using the partial derivatives of the plant at its operating point. The error back-propagation rule can then be used to adjust the weights in all layers except the virtual output layer (the plant). The control system architecture shown in Fig. 2.4 is used in this study.

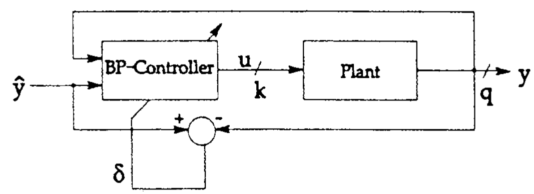

Fig. 2.4 Controller Architecture

During a forward-pass of the back-propagation algorithm, an activation of the input is propagated through the network and plant to yield an output. The output layer (plant) can be thought of as consisting of units that use a linearized transfer function, while the layers of the neural controller consists of units having a chosen activation function. Therefore, the connection weight between the controller output u_k and the plant output y_q can be expressed as

$$w_{qk} = \frac{\partial y_q}{\partial u_k} \tag{2.1}$$

During the backward-pass of the algorithm, the differences between the desired and the actual outputs are propagated back through the plant and the network. By starting at the controller's output layer, each layer of the neural controller is successively adjusted by using these back-propagated differences. The generalized delta-rule [10] can be used to evaluate the error and the weighted-sum-of-inputs at the plant's output units. Hence,

$$d_q = (\hat{y}_q - y_q)\ \frac{\partial\ a_q}{\partial\ n_q} \tag{2.2}$$

$$n_q = \sum_t \frac{\partial\ y_q}{\partial\ u_t}\ u_t \tag{2.3}$$

However, the transfer function of each plant unit is simply equal to unity. Therefore, the error at a plant output can be written as

$$d_q = (\hat{y}_q - y_q) \tag{2.4}$$

These errors are, in turn, used to calculate the errors in the output units of the controller. Therefore, the error in unit k is given by

$$d_k = f'(n_k)\sum_s (d_s w_{ks}) \tag{2.5}$$

where s denotes all units in the succeeding layer connected to unit k. f' is the derivative of unit k's activation function. Substituting (2.1) and (2.4) in (2.5), the error in the output units of the controller becomes

$$d_k = f'(n_k)\left\{\sum_q (\hat{y}_q - y_q)\frac{\partial\ y_q}{\partial\ u_k}\right\} \tag{2.6}$$

Using the generalized delta-rule, the weight adjustments in the controller output layer are obtained as

$$\Delta w_{ij} = \eta\ d_j\ a_i \tag{2.7}$$

where η is the learning rate. This update equation is also used in all preceding network layers. In general, the partial derivatives $\partial\ y_q/\partial\ u_k$ depend on the plant's operating point and, therefore, these are essentially unkown. In most cases, however, the signs of these sensitivity derivatives are known and are basically constant over the operating range of the plant. Generally, one would have a qualitative knowledge of the orientation in which the control parameters influence the outputs of the plant. Therefore, we can approximate the partial derivatives by their sign [6]. The neural controller creates a distributed representation of the numerical values of these partial derivatives by incorporating these values into the weights of the neural controller [4]. The errors in the controller's output layer can finally be expressed as

$$d_k = f'(n_k)\left\{\sum_q (\hat{y}_q - y_q)\cdot \mathrm{sign}\left(\frac{\partial\ y_q}{\partial\ u_k}\right)\right\} \tag{2.8}$$

III. SYSTEM MODEL

To simulate the system dynamics, a linear model of the building-TMD system is used [1],[2],[3]. This model is used with a Runge-Kutta algorithm to approximate the system response. It should be noted that this model is never used to design the controller. Fig. 3.1 shows a schematic diagram of the building-TMD system[2]. Since the TMD is used primarily to suppress the first fundamental mode in wind-induced motion, a one-degree-of-freedom system gives a good approximation to the first-mode structural motion. Therefore, the building is modeled as a single-degree-of-freedom system with a first-mode modal mass m_B, a damping constant b_B, and a spring (stiffness) constant k_B. The corresponding quantities associated with the TMD are indicated as m_D, b_D, and k_D. The absolute displacements of the building and the TMD are labeled as y_B and y_D, respectively, while the wind force acting on the building and the control force are indicated as F_w and u, respectively.

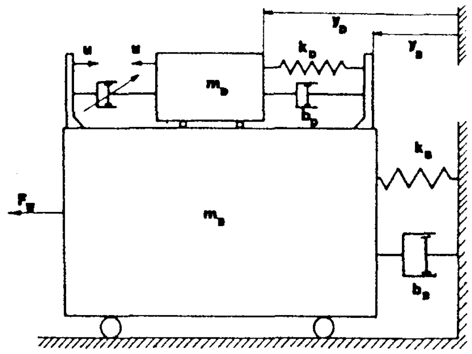

Fig. 3.1 Building-TMD System

The equations of motion for the system of Fig. 3.1 can be written as:

$$m_B\ddot{y}_B + b_B\dot{y}_B + k_B y_B = b_D\dot{z} + k_D z + F_w - u \tag{3.1}$$

$$m_D(\ddot{y}_B + \ddot{z}) + b_D\dot{z} + k_D z = u \tag{3.2}$$

where $z = (y_D - y_B)$ = relative displacement
In state-space form :

$$\dot{x} = Ax + B_1 u + B_2 F_w \tag{3.3}$$

$$y = Cx \tag{3.4}$$

where $x = [\,y_B \quad z \quad \dot{y}_B \quad \dot{z}\,]$ = state variables

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -w_B^2 & u_m w_D^2 & -2s_B w_B & u_m 2s_D w_D \\ w_B^2 & -(1+u_m)w_D^2 & 2s_B w_B & -(1+u_m)2s_D w_D \end{bmatrix} \tag{3.5}$$

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ -1/m_B \\ (1+u_m)/(u_m m_B) \end{bmatrix} \tag{3.6}$$

$$B_2 = \begin{bmatrix} 0 \\ 0 \\ 1/m_B \\ -1/m_B \end{bmatrix} \tag{3.7}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

The other parameters in the above equations are defined as:

$$\left.\begin{array}{ll} w_B = \sqrt{k_B/m_B} & w_D = \sqrt{k_D/m_D} \\ s_B = b_B/2m_B w_B & s_D = b_D/2m_D w_D \\ u_m = m_D/m_B & \end{array}\right\} \tag{3.9}$$

In simulating the system, it is assumed that the wind force is a deterministic input of the form :

$$F_w = 3p\sin(wt) + 7p\sin(2wt) + 5p\sin(3wt) + 4p\sin(4wt) \tag{3.10}$$

IV. SIMULATION RESULTS

To determine the quality of control, the proposed scheme is applied to the building-TMD system used by Hrovat et. al. [2]. A two-input, two-layer neural network is chosen as the controller. The two inputs are the building displacement and the desired displacement (zero). The fully-interconnected network has seven neurons in the first layer and a single neuron in the second (controller output) layer. A linear activation function is used for all neuron units. All weights are set initially to 1.0.

Several network structures were considered by varying the number of layers and the number of neurons per layer. It is found that a two-layer network is sufficient for the test system. A linear activation function with a slope of 0.05 is found to give good results. In general, a learning rate of 1.0 results in faster learning times.

The results of some significant test cases are given in the following table :

Case	n_l	n_n	h	η	t_s	y_{max}	u_{max}	
1	2	7,1	0.5	1.0	250	0.69	5.7	
2	2	2,1	0.5	1.0	340	0.73	5.5	
3	2	7,1	1.0	1.0	500	1.04	5.3	
4	2	7,1	0.1	1.0	85			
5	2	7,1	0.5	0.5	550	1.15	4.1	
6	3	7,7,1	0.5	1.0	330	0.73	11.2	
7	2	7,1	0.5	1.0	200	0.75	8.0	$w'=10$
Hro-vat					0	0.88	9.0	
P-control			0.1		0	0.50	7.5	

where
n_l = no. of layers
n_n = no. of neurons
h = update time increment (sec.)
η = learning rate
t_s = time it takes to fall below 1" displacement
y_{max} = maximum displacement at 500 sec.
u_{max} = maximum control force at 500 sec.

If no control force is applied, the building displacement would be as shown in Fig. 4.1.

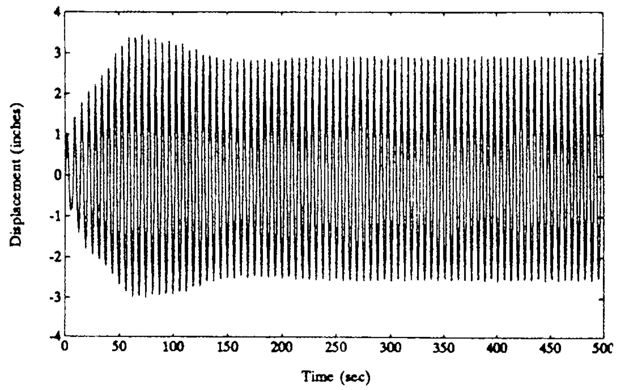

Fig. 4.1 No control force applied

It can be observed that the results for Case 1 (see Fig. 4.2) are slightly better than those obtained in Hrovat et al. [2] where linear-quadratic (LQ) control is used.

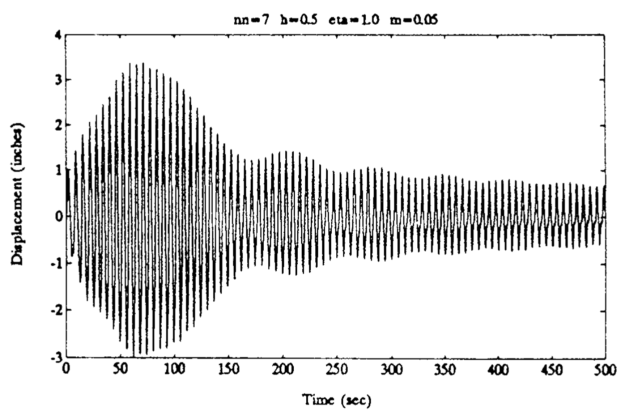

Fig. 4.2 Case 1 Results (a) Dispalcement

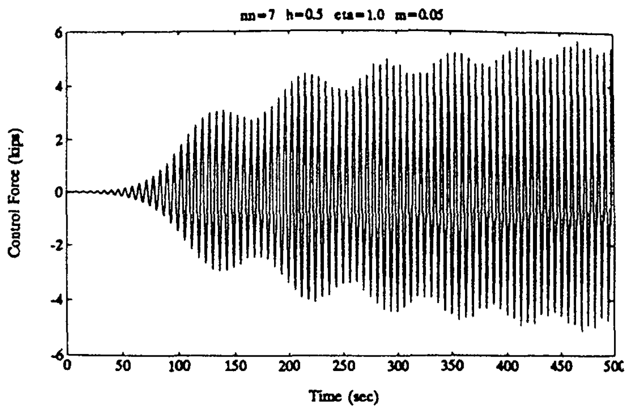

Fig 4.2 Case 1 Results (b) Control Force

Case 1 results are also comparable to those obtained using a proportional controller (see Fig. 4.3). For this case, the building displacement is fed back through a proportional gain (K) to determine the control force.

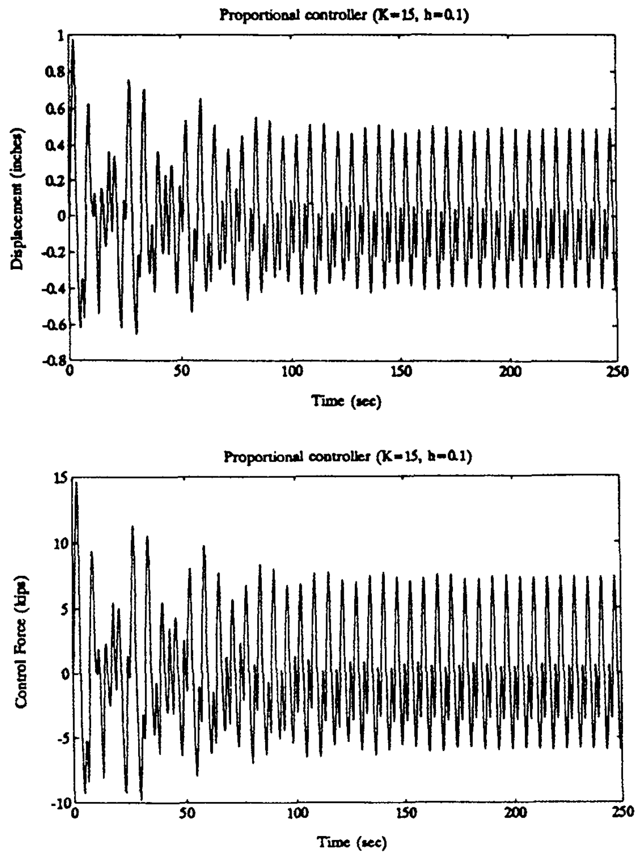

Fig. 4.3 Proportional Control

From the test cases considered, it is observed that the controller's tendency to oscillate and the time it takes to attain the desired displacement limits, is very sensitive to the initial weights, the update time increment, and the learning rate. Lower values for the update time increment result in faster reduction of the displacement. However, larger control forces are required to attain this. A compromise between the speed and maximum force will have to be decided by the designer. Decreasing the learning rate tends to slow down the learning process of the network.

V. CONCLUSIONS

A neural network controller has been designed to provide active control of wind-induced vibrations in tall buildings. The controller learns to control the plant autonomously and without need of a specific learning stage. By treating the plant as an additional and unmodifiable layer of the neural network, the back-propagation algorithm can be used while requiring only simple qualitative knowledge of the plant.

Some of the possible directions that can be pursued in future works on active TMD control using neural networks include : (1) on-line optimization of the controller's structure, (2) use of adaptive learning rates for individual weights, and (3) indirect control by dynamic back-propagation.

VI. NOMENCLATURE

d_q, d_k, d_j	error of unit q, k, j
y_q	plant output
u_k	controller output
a_q, a_k	output value for unit q, k
n_q, n_k	weighted sum of inputs to unit q, k
f'	derivative of the activation function
y_q	desired outputs
η	learning rate
w_{ij}	connection weight between units i and j
n_n	number of neurons per layer
h	update time increment
m	slope of the linear activation function
m_B	building modal mass
b_B	building damping constant
k_B	building spring constant
y_B	building displacement
m_D	TMD modal mass
b_D	TMD damping constant
k_D	TMD spring constant
y_D	TMD displacement
F_w	wind force
u	control force
A, B_1, B_2, C	state-space matrices
w	excitation fundamental frequency
K	feedback gain

VII. REFERENCES

[1] Chang, J.C.H. and Soong, T.T.,"Structural Control Using Active Tuned Mass Dampers, " ASCE Journal of Engineering Mechanics, Vol. 106, No. EM6, December 1980, pp. 1091 - 1098.

[2] Hrovat, D., Barak, P., and Rabins, M.," Semi-Active Versus Passive or Active Tuned Mass Dampers for Structural Control," ASCE Journal of Engineering Mechanics, Vol. 109, No. 3, June 1983, pp. 691 - 705.

[3] Kwok, K.C.S., R. Narayanan and T.M. Roberts," Damping and Control of Structures Subjected to Dynamic Loading," Elsevier Science Publishers Ltd., London and New York, 1991, pp. 303 - 334.

[4] Schifman, W.H. and Geffers, H.W.," Adaptive Control of Dynamic Systems by Back Propagation Networks," Neural Networks, Vol. 6, 1993, pp. 517 - 524.

[5] Psaltis, D., Sideris, A., and Yamamura, A.A.," A Multilayered Neural Network Controller," IEEE Control Systems Magazine, Vol. 8, No. 2, April 1988, pp. 17 - 21.

[6] Saerens, M. and Soquet, A.," A Neural Controller Based on Back Propagation Algorithm," Proc. of the First IEEE International Conference on Artificial Neural Networks, IEEE Press, 1989, pp. 211 - 215.

[7] Chen, V.C. and Pao Y-H.," Learning Control with Neural Networks, " Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 3, 1989, pp. 1448 -1453.

[8] Waddoups, M.A. and Moore, K.L.," Neural Networks for Iterative Learning Control," Proceedings of the American Control Conference, Chicago, Vol. 4, June 1992, pp.3049 - 3051.

[9] Nguyen, D.H. and Widrow, B.," Neural Networks for Self-Learning Control Systems," IEEE Control Systems Magazine, April 1990, pp.18 - 23.

[10] Rumelhart, D.E., Hinton, G.E., and Williams, R.J.," Learning Internal Representations by Error Propagation," in Parallel Distributed Processing, Vol. 1 : Foundations. Cambridge, MA: MIT Press, 1986.