Seismic Response Modeling of Multi-Story Buildings Using Neural Networks

JOEL P. CONTE* AND AHMAD J. DURRANI**
Civil Engineering Department
Rice University
Houston, TX 77251-1892

ROBERT O. SHELTON†
Software Technology Branch
NASA
Johnson Space Center
Houston, TX 77058

ABSTRACT: A neural network based approach to model the seismic response of multi-story frame buildings is presented. The seismic response of frames is emulated using multi-layer feedforward neural networks with a backpropagation learning algorithm. Actual earthquake accelerograms and corresponding structural response obtained from analytical models of buildings are used in training the neural networks. The application of the neural network model is demonstrated by studying one to six story high building frames subjected to seismic base excitation. Furthermore, the learning ability of the network is examined for the case of multiple inputs where lateral forces at floor levels are included simultaneously with the base excitation. The effects of the network parameters on learning and accuracy of predictions are discussed. Based on this study, it is found that appropriately configured neural network models can successfully learn and simulate the linear elastic dynamic behavior of multi-story buildings.

INTRODUCTION

DURING their lifetime building structures can experience severe external hazards such as hurricane winds and earthquakes; therefore, they need to be designed to safely resist such dynamic loads. The design procedure for dynamic loads typically requires calculation of the dynamic response of the structure, which has been traditionally performed using two- and three-dimensional finite element analysis (Clough and Penzien, 1993). The finite element models are based on a number of simplifying assumptions which result in significant discrepancies between analytically predicted and actual measured responses. This may not be a serious matter as far as design and safety are concerned, but it may have far reaching consequences when dealing with inverse dynamics problems such as in active control of structures (Soong, 1990), monitoring of structural integrity, and signature analysis as a non-destructive evaluation technique for assessing the damage of buildings (Agbabian et al., 1991). In the inverse problem, often referred to as system identification, the parameters of the assumed class of structural models are estimated from excitation and actual structural response measurements. Most of the existing system identification techniques as applied to civil engineering structures are based on linear time-invariant structural models whose parameters are estimated using various deterministic or stochastic methods such as least squares, maximum likelihood, and Kalman filtering (Kozin and Natke, 1986). These techniques typically utilize the modal decomposition concept (Beck and Jennings, 1980). Research in non-linear system identification of civil engineering structures has been rather limited due to the difficulties in analytically modeling the types of non-linearity present in these structures (Hoshiya and Saito, 1984).

Artificial neural networks (ANNs) provide a fundamentally different approach to system identification. They have been successfully applied for identification and control of dynamic systems in various fields of engineering (Bialasiewicz and Ho, 1991; Bozich and Mackay, 1991; Nikolaou and Hanagandi, 1991; Singh et al., 1991). Applications of ANNs to civil engineering problems in structural dynamics are only a recent phenomenon (Rehak et al., 1989; Wen et al., 1992; Elkordy et al., 1993). Despite the fact that neural networks do not provide direct physical insight into structural response, their ability to learn efficiently the system dynamics from input-output data renders them particularly attractive for adaptive real-time identification and control applications.

*Assistant Professor of Civil Engineering and author to whom correspondence should be addressed.
**Associate Professor of Civil Engineering.
†Research Director.

1045-389X/94/03 0392–11 $6.00/0
© 1994 Technomic Publishing Co., Inc.

THE NEURAL NETWORK CONCEPT

In work spanning at least fifty years, researchers from many disciplines have attempted to understand the information processing capabilities of biological nervous systems. This work has resulted in a rudimentary understanding of a few principles that appear common to most organic information processing. The key concept that has been distilled from the biological systems is the artificial neuron. Real neurons are cells that are sensitive to electrical stimulation. When sufficient stimulation is present, the neuron fires, i.e., produces electrical pulses which can in turn stimulate other neurons. The exact internal dynamics of nerve cells is complex and not considered at all in most artificial neural systems. Artificial neurons are modeled as nodes in a directed graph, each arc in the graph representing, by its direction, the flow of electrical impulses in a biological system. The electrical pulses of the organic system are represented as scalar activations which are passed along each arc in the graph, and which are modified in transit by multiplication by a weighting value associated with the arc which represents the strength of the connection. The output of the artificial neuron is calculated by accumulating the weighted sum of all its inputs, and evaluating the transfer function that is associated with each artificial neuron. The transfer function is used to supply a non-linearity; the remainder of the artificial neural system is comprised of linear multiplications and sums well-known mathematically as dot products.

In spite of the biological motivation, the concepts that will be applied in this context are easily recognizable as generalizations of familiar mathematical techniques for function approximation. These neural models are said to learn associations from exposure to data, but in reality the process is no more or less learning than is the use of linear regression to obtain parameter estimates in classical system identification. To be specific, the task at hand is to construct a model of a dynamic system. The target system has a vector input $\mathbf{X}(t)$ and a vector output $\mathbf{Y}(t)$. As is customary for the numerical treatment of such systems, the time variable is made discrete. The resulting system maps its input and current state $(\mathbf{X}_{n+1}; \mathbf{Y}_n)$ into its next output $\mathbf{Y}_{n+1}$. As a matter of implementation, the components of $(\mathbf{X}_{n+1}; \mathbf{Y}_n)$ may span one or more time steps. This so-called windowing technique, as applied to identification of multi-degree-of-freedom (MDOF) dynamic systems, provides to the estimation algorithm a truncated time history of system inputs and outputs from which the subsequent system output is to be predicted. Although the windowing concept is a well-known method for treatment of dynamic systems, the problem-dependent details of implementation are described here.

Suppose that the system in question has a c-dimensional vector input $\mathbf{X}(t)$ or forcing function and a d-dimensional vector output $\mathbf{Y}(t)$ or response vector. Let Δt denote the time sampling interval. If K_O output vector values and $K_I + 1$ input vector values are to be used in the prediction process, then

$$\mathbf{X}_{n+1} = \begin{bmatrix} x_1((n+1)\Delta t), x_1(n\Delta t), \ldots, \\ x_1((n - K_I + 1)\Delta t), \\ x_2((n+1)\Delta t), x_2(n\Delta t), \ldots, \\ x_2((n - K_I + 1)\Delta t), \ldots, \\ x_c((n+1)\Delta t), x_c(n\Delta t), \ldots, \\ x_c((n - K_I + 1)\Delta t) \end{bmatrix}_{1 \times c(K_I+1)} \tag{1a}$$

$$\mathbf{Y}_n = \begin{bmatrix} y_1(n\Delta t), y_1((n' - 1)\Delta t), \ldots, \\ y_1((n - K_O + 1)\Delta t), \\ y_2(n\Delta t), y_2((n-1)\Delta t), \ldots, \\ y_2((n - K_O + 1)\Delta t), \ldots, \\ y_d(n\Delta t), y_d((n-1)\Delta t), \ldots, \\ y_d((n - K_O + 1)\Delta t) \end{bmatrix}_{1 \times d(K_O)} \tag{1b}$$

Note that the neural network model of the system only needs to supply an estimate, denoted by the "hat" symbol, of the d-dimensional system output at time $t_{n+1} = (n+1)\Delta t$, $\hat{\mathbf{T}}_{n+1}$. During the learning phase of neural network modeling, an ensemble of input- (exact) output pairs $[(\mathbf{X}_{n+1}, \mathbf{Y}_n): \mathbf{T}^{\mathrm{T}}_{n+1}]$, called the training set, is presented to the network. On the other hand, when the neural network model of the system is recalled (network generalization) for different system input data, the input vector to the network at time $t_{n+1} = (n+1)\Delta t$, $(\mathbf{X}_{n+1}, \mathbf{Y}_n)$, is constructed from the known real system inputs $\mathbf{X}_{n+1}$ and the neural network predicted system output $\hat{\mathbf{Y}}_n$ in place of the unknown actual system output $\mathbf{Y}_n$.

If the system in question is linear, then there are matrices $\mathbf{A}$ and $\mathbf{B}$ such that

$$\hat{\mathbf{T}}_{n+1} = \mathbf{A}\mathbf{Y}^{\mathrm{T}}_n + \mathbf{B}\mathbf{X}^{\mathrm{T}}_{n+1} \tag{2}$$

where the superscript $(\ )^{\mathrm{T}}$ indicates the transpose operation and the $(d \times 1)$ column vector $\hat{\mathbf{T}}_{n+1}$ is the estimate for the system state at time $t_{n+1} = (n+1)\Delta t$. If input-output data from the system is available, then the matrices $\mathbf{A}$ and $\mathbf{B}$ may be found by means of linear regression methods. These techniques, as normally applied, require a table of pairs of the type $[(\mathbf{X}_{n+1}, \mathbf{Y}_n): \mathbf{T}^{\mathrm{T}}_{n+1}]$. Given such a table, it is easy to find the matrices $\mathbf{A}$ and $\mathbf{B}$ that most closely duplicate the target system in the least squares sense.

If the system at hand is believed to be a non-linear perturbation of a linear system, then the model [Equation (2)] can

be augmented in a very simple way. Specifically, a non-linearity $\mathbf{F}$ is added so that Equation (2) becomes

$$\hat{\mathbf{T}}_{n+1} = \mathbf{A}\mathbf{Y}_n^{\mathrm{T}} + \mathbf{B}\mathbf{X}_{n+1}^{\mathrm{T}} + \mathbf{F}(\mathbf{X}_{n+1}, \mathbf{Y}_n) \quad (3)$$

where $\hat{\mathbf{T}}_{n+1}$ is the estimated value for $\mathbf{T}_{n+1}$, the actual system state at time t_{n+1}, and

$$\mathbf{F}(\mathbf{X}_{n+1}, \mathbf{Y}_n) = \sum_{i=1}^{N_H} \boldsymbol{\beta}_i f(\mathbf{V}_i \mathbf{X}_{n+1}^{\mathrm{T}} + \mathbf{W}_i \mathbf{Y}_n^{\mathrm{T}} - \theta_i) \quad (4)$$

Each term of the sum in Equation (4) would, to a connectionist, represent the output of a neuron connected to the inputs $\mathbf{W}_{n+1}$ and $\mathbf{Y}_n$ with synaptic weights $\mathbf{V}_i$ and $\mathbf{W}_i$, respectively, as represented in Figure 4 (page 397) in anticipation of the structural application presented hereafter. The values θ_i, $i = 1, \ldots, N_H$, are called offsets or biases, and the vector coefficients $\boldsymbol{\beta}_i (N_H \times 1)$ contain the strengths of the output connections linking the hidden and the output layers. In Equation (4), N_H stands for the number of hidden nodes in the three layer network representation of Equation (3). From Figure 4, it is noted that only the hidden nodes are non-linear, i.e., contain the non-linear activation function $f(\ )$; both input and output nodes are linear. It is also noted that the linear input-output relationship described in Equation (2) is represented by the "linear connections" linking input and output nodes directly (see Figure 4). The hyperbolic tangent function, or its close relative the sigmoid logistic map, is a frequent choice for the function $f(\ )$ because the flat asymptotes of both functions form an approximation of the phenomena of neuronal saturation. The present application uses the hyperbolic tangent function, i.e.,

$$f(u) = \tanh(u) \quad (5)$$

No claim of optimality is made for either the hyperbolic tangent or the logistic map, but these functions do possess several desirable properties. In particular, such functions are bounded, monotonic, have derivative = 1, 0.25, respectively, at the origin, and decay rapidly to zero for large values of the argument. The effect of these conditions on the function implemented in Equation (3) is for the mapping to act like a collection of several linear functions smoothly patched together. In this way, something resembling a linear spline fit is produced. However, the corners normally associated with the knots are rounded and the locations of these transition regions are evolved adaptively from a simple update equation for the model parameters.

In the linear case [Equation (2)], there are well-known efficient methods for finding optimal matrices $\mathbf{A}$ and $\mathbf{B}$ (Ljung, 1987). These methods have non-linear extensions that can be applied to the problem of optimal parameter estimation in Equation (3), but the present application will be confined to the simple gradient update scheme. Although such steepest descent methods are mathematically trivial, there are several reasons why they may be the technique of choice for many situations of interest. In particular, second order methods will normally require second order storage, i.e., if the number of variables in the problem is n, then the space (and time per step) needed will be proportional to n^2. Another difficulty with second order optimization for estimating parameters describing a time-varying system is that most implementations require a fixed set of input-output pairs of the form $[(\mathbf{X}_{n+1}, \mathbf{Y}_n) : \mathbf{T}_{n+1}^{\mathrm{T}}]$. As noted before, such information is available from system observations, but it may not make sense to collect the large amount of data needed to drive a sophisticated second order optimization method, and run the optimization, only to discover that the plant has drifted to such an extent that the parameter estimates so obtained are no longer useful.

In contrast, the gradient scheme is cheap, local, i.e., it can easily process data from the system as it is measured, and thus may be ideal for adaptively following a time-varying system. The equations for all estimation methods are derived by first constructing an objective function. The general structure of such a function is a "sum of squares" of residuals (prediction errors) of the form $\mathbf{T}_{n+1} - \hat{\mathbf{T}}_{n+1}$. For the gradient method, we take the sum at a single point in time. The objective function so constructed is given by

$$e_{n+1} = |\mathbf{T}_{n+1} - (\mathbf{A}\mathbf{Y}_n^{\mathrm{T}} + \mathbf{B}\mathbf{X}_{n+1}^{\mathrm{T}} + \mathbf{F}(\mathbf{X}_{n+1}, \mathbf{Y}_n))|^2 \quad (6)$$

where $|\ \ldots\ |$ symbolizes the Euclidean or L^2 norm. The parameters of the model [Equations (3) and (4)] are grouped notationally into six arrays, five of which ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{V}$, $\mathbf{W}$ and $\boldsymbol{\beta}$) are matrices. The offsets $\boldsymbol{\theta}$ form a vector of dimension $(N_H \times 1)$. If the row vectors containing the past and current (at time t_{n+1}) input excitation, $\mathbf{X}_{n+1}$, and past responses, $\mathbf{Y}_n$, are of dimensions $(1 \times p)$ and $(1 \times q)$, respectively, and the system output vector is of dimension $(d \times 1)$, then $\mathbf{A}$ and $\mathbf{B}$ are of sizes $(d \times q)$ and $(d \times p)$, respectively. Define the matrices $\mathbf{W}$ and $\mathbf{V}$ of dimension $(N_H \times p)$ and $(N_H \times q)$, respectively, so that the weight row vectors $\mathbf{W}_i$ and $\mathbf{V}_i$ in Equation (4) comprise respectively the ith row of each matrix. Finally, the matrix $\boldsymbol{\beta}$ is constructed so that the vector coefficients $\boldsymbol{\beta}_i$ in Equation (4) form its columns, and thus $\boldsymbol{\beta}$ is of dimension $(d \times N_H)$. The coefficients comprising $\mathbf{A}$, $\mathbf{B}$, $\mathbf{W}$, $\mathbf{V}$, $\boldsymbol{\beta}$, and $\boldsymbol{\theta}$ are then evolving over time, so their dependence on the time t_n $(= n\Delta t)$ is indicated (where necessary) by $\mathbf{A}(n)$, $\mathbf{B}(n)$, $\mathbf{W}(n)$, $\mathbf{V}(n)$, $\boldsymbol{\beta}(n)$ and $\boldsymbol{\theta}(n)$. A popular form of the gradient update rule for the model parameters is described by the equations (Rumelhart and McClelland, 1988)

$$\mathbf{A}(n+1) = \mathbf{A}(n) - \delta\mathbf{A}(n) \quad (7)$$

$$\mathbf{B}(n+1) = \mathbf{B}(n) - \delta\mathbf{B}(n) \quad (8)$$

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \delta\mathbf{W}(n) \quad (9)$$

$$\mathbf{V}(n+1) = \mathbf{V}(n) - \delta\mathbf{V}(n) \quad (10)$$

$$\boldsymbol{\beta}(n+1) = \boldsymbol{\beta}(n) - \delta\boldsymbol{\beta}(n) \tag{11}$$

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) - \delta\boldsymbol{\theta}(n) \tag{12}$$

where

$$\delta\mathbf{A}(n) = \alpha_{\mathbf{A}}(n)[(1 - \lambda_{\mathbf{A}}(n))\nabla_{\mathbf{A}}e_{n+1} + \lambda_{\mathbf{A}}\delta\mathbf{A}(n-1)] \tag{13}$$

$$\delta\mathbf{B}(n) = \alpha_{\mathbf{B}}(n)[(1 - \lambda_{\mathbf{B}}(n))\nabla_{\mathbf{B}}e_{n+1} + \lambda_{\mathbf{B}}\delta\mathbf{B}(n-1)] \tag{14}$$

$$\delta\mathbf{W}(n) = \alpha_{\mathbf{W}}(n)[(1 - \lambda_{\mathbf{W}}(n))\nabla_{\mathbf{W}}e_{n+1} + \lambda_{\mathbf{W}}\delta\mathbf{W}(n-1)] \tag{15}$$

$$\delta\mathbf{V}(n) = \alpha_{\mathbf{V}}(n)[(1 - \lambda_{\mathbf{V}}(n))\nabla_{\mathbf{V}}e_{n+1} + \lambda_{\mathbf{V}}\delta\mathbf{V}(n-1)] \tag{16}$$

$$\delta\boldsymbol{\beta}(n) = \alpha_{\boldsymbol{\beta}}(n)[(1 - \lambda_{\boldsymbol{\beta}}(n))\nabla_{\boldsymbol{\beta}}e_{n+1} + \lambda_{\boldsymbol{\beta}}\delta\boldsymbol{\beta}(n-1)] \tag{17}$$

$$\delta\boldsymbol{\theta}(n) = \alpha_{\boldsymbol{\theta}}(n)[(1 - \lambda_{\boldsymbol{\theta}}(n))\nabla_{\boldsymbol{\theta}}e_{n+1} + \lambda_{\boldsymbol{\theta}}\delta\boldsymbol{\theta}(n-1)] \tag{18}$$

In the above formulas, the gradient of the scalar function e_{n+1} with respect to a vector $\mathbf{v}$ is denoted by $\nabla_{\mathbf{v}}(e_{n+1})$ and is defined as $[(\partial e_{n+1}/\partial v_1),\ldots,(\partial e_{n+1}/\partial v_n)]^{\mathrm{T}}$, whereas the gradient of the scalar function e_{n+1} with respect to a matrix $\mathbf{M}$ is denoted by $\nabla_{\mathbf{M}}(e_{n+1})$ and is defined such that $[\nabla_{\mathbf{M}}(\mathbf{e}_{n+1})]_{ij} = \partial e_{n+1}/\partial m_{ij}$. The parameters $\alpha_{\mathbf{A}}(n)$, $\lambda_{\mathbf{A}}(n)$, $\alpha_{\mathbf{B}}(n)$, $\lambda_{\mathbf{B}}(n)$, $\alpha_{\mathbf{W}}(n)$, $\lambda_{\mathbf{W}}(n)$, $\alpha_{\mathbf{V}}(n)$, $\lambda_{\mathbf{V}}(n)$, $\alpha_{\boldsymbol{\beta}}(n)$, $\lambda_{\boldsymbol{\beta}}(n)$, $\alpha_{\boldsymbol{\theta}}(n)$, $\lambda_{\boldsymbol{\theta}}(n)$, may be fixed values or may be selected adaptively, e.g., back-tracking line search, in order to remove as much error as possible with each update. In the neural network literature (Wasserman, 1989), the parameters $\alpha_{(\ldots)}$ are called the training or learning rate coefficients, whereas the parameters $\lambda_{(\ldots)}$ are called the momentum coefficients in that they produce an exponentially decaying average of previous local updates. The partial derivatives in Equations (13) to (18) are given by:

$$\nabla_{\mathbf{A}(n)}(e_{n+1}) = 2(\hat{\mathbf{T}}_{n+1} - \mathbf{T}_{n+1})\mathbf{Y}_n \tag{19}$$

and

$$\nabla_{\mathbf{B}(n)}(e_{n+1}) = 2(\hat{\mathbf{T}}_{n+1} - \mathbf{T}_{n+1})\mathbf{X}_{n+1} \tag{20}$$

where $\hat{\mathbf{T}}_{n+1}$ is the state estimate from Equation (3), $\mathbf{X}_{n+1}$ is the current (at time t_{n+1}) and truncated past system input and $\mathbf{Y}_n$ is the truncated past measured (actual) system output (up to time t_n).

$$\nabla_{\boldsymbol{\beta}_i(n)}(e_{n+1}) = 2f(\mathbf{V}_i\mathbf{X}_{n+1}^{\mathrm{T}} + \mathbf{W}_i\mathbf{Y}_n^{\mathrm{T}} - \theta_i)[\hat{\mathbf{T}}_{n+1} - \mathbf{T}_{n+1}] \tag{21}$$

where $f(\ )$ is the non-linear function appearing in Equation (4). The gradient vector $\nabla_{\boldsymbol{\beta}_i(n)}(e_{n+1})$ of dimension $(d \times 1)$ in Equation (21) represents the ith column of the gradient matrix $\nabla_{\boldsymbol{\beta}(n)}(e_{n+1})$ of dimension $(d \times N_H)$.

$$\frac{\partial e_{n+1}}{\partial \theta_i(n)} = -2f'(\mathbf{V}_i\mathbf{X}_{n+1}^{\mathrm{T}} + \mathbf{W}_i\mathbf{Y}_n^{\mathrm{T}} - \theta_i)\boldsymbol{\beta}_i^{\mathrm{T}}[\hat{\mathbf{T}}_{n+1} - \mathbf{T}_{n+1}] \tag{22}$$

The single partial derivative in Equation (22) represents the ith component of the gradient vector $\nabla_{\boldsymbol{\theta}(n)}(e_{n+1})$.

$$\nabla_{\mathbf{V}_i(n)}(e_{n+1}) = -\frac{\partial e_{n+1}}{\partial \theta_i(n)}\mathbf{X}_{n+1}^{\mathrm{T}} \tag{23}$$

The transpose of the gradient vector $\nabla_{\mathbf{V}_i(n)}(e_{n+1})$ in Equation (23) is the ith row of the gradient matrix $\nabla_{\mathbf{V}(n)}(e_{n+1})$.

$$\nabla_{\mathbf{W}_i(n)}(e_{n+1}) = -\frac{\partial e_{n+1}}{\partial \theta_i(n)}\mathbf{Y}_n^{\mathrm{T}} \tag{24}$$

Similarly, the transpose of the gradient vector $\nabla_{\mathbf{W}_i(n)}(e_{n+1})$ in Equation (24) is the ith row of the gradient matrix $\nabla_{\mathbf{W}(n)}(e_{n+1})$. The preceding algorithm for updating parameters by gradient descent, together with the equations for the partial derivatives, comprise the so-called back-propagation method for neural networks. Since the hyperbolic tangent function is used for the non-linearity f, its derivative f' is calculated from the relation

$$f'(u) = 1 - f^2(u) \tag{25}$$

STRUCTURAL MODEL

Multi-story buildings can be represented conveniently by shear-beam type models as shown in Figure 1. This simplified structural model assumes that (1) the floor diaphragms are infinitely rigid in their plane, (2) the beams are infinitely stiff (rigid) in flexure compared to the columns, and (3) all the masses are lumped at the floor levels.

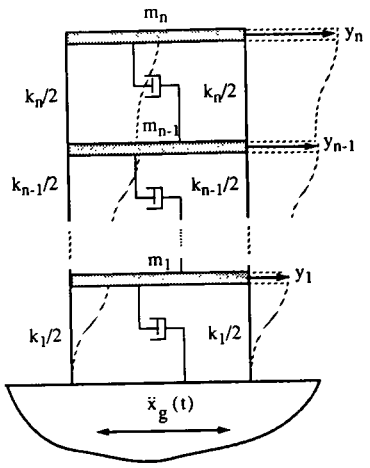

Figure 1. *Shear beam structural model.*

Table 1. Structural properties of building models.

	One-Story	Two-Story	Six-Story
Story stiffnesses (k/in)	$k = 109.7$	$k_1 = 30.69$ $k_2 = 44.17$	$k_1 = k_2 = k_3 = 43.94$ $k_4 = k_5 = k_6 = 35.16$
Story masses (k-sec²/in)	$m = 0.25$	$m_1 = 0.15$ $m_2 = 0.10$	$m_1 = m_2 = m_3 = 0.15$ $m_4 = m_5 = m_6 = 0.10$

The earthquake response of a N-story building is governed by the following equation of motion:

$$\mathbf{M}\ddot{\mathbf{Y}}(t) + \mathbf{C}\dot{\mathbf{Y}}(t) + \mathbf{K}\mathbf{Y}(t) = -\mathbf{M}\mathbf{1}\ddot{x}_g(t) \qquad (26)$$

In the above equation, $\mathbf{M}_{N\times N}$, $\mathbf{C}_{N\times N}$, and $\mathbf{K}_{N\times N}$ represent the mass, damping, and stiffness matrices of the system, respectively; the vector $\mathbf{Y}_{N\times 1}(t) = [y_1(t)y_2(t) \ . \ . \ . \ y_N(t)]^T$ contains the displacement response of the various floors relative to the ground; the "dot" symbol indicates a time derivative; $\mathbf{1}_{N\times 1}$ represents a column vector of ones; and $\ddot{x}_g(t)$ is the input ground acceleration time history.

Several multi-story building models were studied to investigate the ability of neural networks to learn the dynamics of MDOF structures. These included one-story, two-story, and six-story buildings. The structural properties and dynamic characteristics of these building models are given in Table 1 and Table 2, respectively.

The dynamic response of the building models was studied for the El Centro record from the 1940 Imperial Valley earthquake in California, shown in Figure 2, and the Orion Blvd. record from the 1971 San Fernando earthquake in California, shown in Figure 3. These records were chosen because of their historical importance and their distinctly different frequency characteristics. In each case, the actual (exact) earthquake response of the building model is determined by integrating the equation of motion [Equation (26)] using modal superposition and an exact piecewise linear integration scheme which assumes a linear interpolation between the digitized values of the earthquake ground acceleration.

NEURAL NETWORK MODEL

For an N-story shear-building model subjected to a single component of earthquake ground acceleration, the input-output pairs $[(\mathbf{X}_{n+1}, \mathbf{Y}_n):\mathbf{T}^T_{n+1}]$ needed in training and recalling the neural network represented in Figure 4 take the following form:

$$\mathbf{X}_{n+1} = [\ddot{x}_g((n+1)\Delta t), \ddot{x}_g(n\Delta t), \ldots, \ddot{x}_g((n - K_I + 1)\Delta t)]_{1\times(K_I+1)} \qquad (27)$$

$$\mathbf{Y}_n = \begin{bmatrix} y_1(n\Delta t), y_1((n-1)\Delta t), \ldots, \\ y_1((n - K_O + 1)\Delta t), \\ y_2(n\Delta t), y_2((n-1)\Delta t), \ldots, \\ y_2((n - K_O + 1)\Delta t), \ldots, \\ y_N(n\Delta t), y_N((n-1)\Delta t), \ldots, \\ y_N((n - K_O + 1)\Delta t) \end{bmatrix}_{1\times N(K_O)} \qquad (28)$$

$$\mathbf{T}_{n+1} = \begin{bmatrix} y_1((n+1)\Delta t) \\ y_2((n+1)\Delta t) \\ \ldots \\ y_N((n+1)\Delta t) \end{bmatrix}_{N\times 1} \qquad (29)$$

Although this particular neural network implementation does not require any scaling, to maintain consistency with previous work, the system input ($\mathbf{X}$) and output ($\mathbf{Y}$ and $\mathbf{T}$) values were scaled between 0.1 and 0.9 before training and recalling (input only) the network and correspondingly unscaled after recalling the network. In the situation where the network is recalled for a forcing function different than the one used in training and when no actual initial response

Table 2. Dynamic properties of building models.

	One-Story	Two-Story	Six-Story
Undamped natural periods (sec)	$T = 0.30$	$T_1 = 0.603$ $T_2 = 0.218$	$T_1 = 1.341$ $T_2 = 0.517$ $T_3 = 0.307$ $T_4 = 0.240$ $T_5 = 0.201$ $T_6 = 0.181$
Modal damping ratios (-)	$\xi = 0.05$	$\xi_1 = 0.05$ $\xi_2 = 0.05$	$\xi_1 = \xi_2 = \xi_3 = 0.05$ $\xi_4 = \xi_5 = \xi_6 = 0.05$

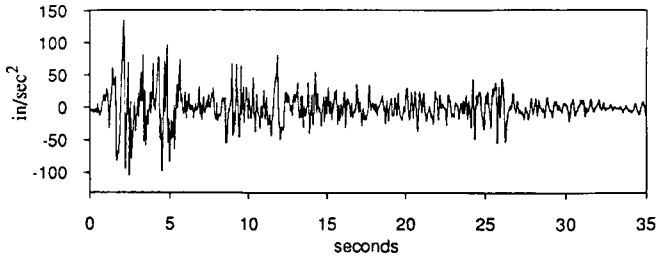

Figure 2. *El Centro ground acceleration record from the 1940 Imperial Valley earthquake.*

data is available, the initial input sample $(\mathbf{X}_{n+1}, \hat{\mathbf{Y}}_n)$ cannot be formed. To overcome this difficulty, all input and output records are padded with K_I and K_O zeros, respectively, at the beginning of the record prior to learning and recall. The learning rates are determined dynamically for each input-output pair presented to the network at each iteration by back-tracking line search which consists of minimizing the local error function e_{n+1}, which is interpolated quadratically. For this study, the gradient update rule was implemented with zero momentum, i.e., the corrections applied to the model parameters are independent of the previous updates.

In the case of multiple excitations, the input vector $\mathbf{X}$ needs to be augmented to include all the forcing functions. For example, if a two-story building is subjected simultaneously to an earthquake ground motion $\ddot{x}_g(t)$, and two lateral control forces at floor levels, $F_1^c(t)$ and $F_2^c(t)$, the input vector $\mathbf{X}_{n+1}$ takes the following form:

$$\mathbf{X}_{n+1} = \begin{bmatrix} \ddot{x}_g((n+1)\Delta t), \ddot{x}_g(n\Delta t), \ldots, \\ \ddot{x}_g((n-K_I+1)\Delta t), \\ F_1^c(n\Delta t), F_1^c((n-1)\Delta t), \ldots, \\ F_1^c((n-K_I+1)\Delta t), \\ F_2^c(n\Delta t), F_2^c((n-1)\Delta t), \ldots, \\ F_2^c((n-K_I+1)\Delta t) \end{bmatrix}_{1\times(3K_I+1)} \tag{30}$$

Here, there is a time delay of Δt on $F_1^c(t)$ and $F_2^c(t)$ in the input vector $\mathbf{X}_{n+1}$ in order to simulate the case of state-feedback control forces (unavailable at time t_{n+1}).

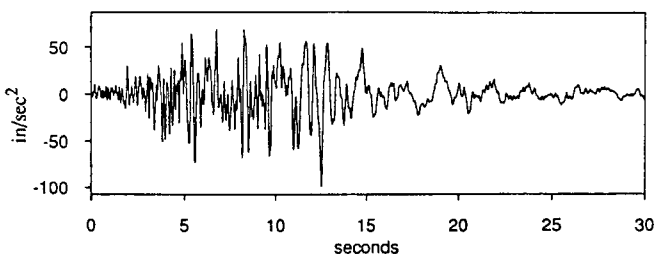

Figure 3. *Orion Blvd. ground acceleration record from the 1971 San Fernando earthquake.*

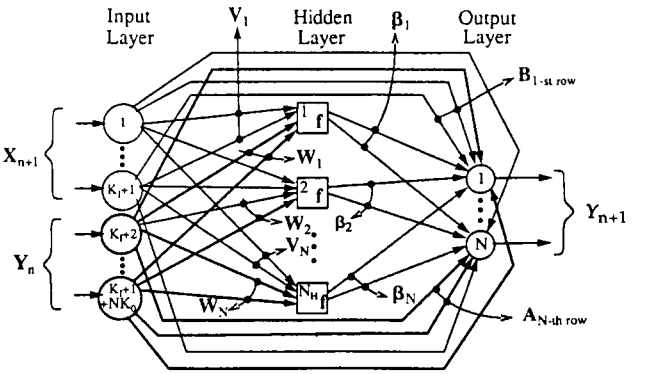

Figure 4. *ANN configuration for dynamic structural systems with single forcing function (N dégrees of freedom).*

RESULTS

The network training for the one-story structural model was performed using a 21-1-1 network configuration (21 input nodes, 1 hidden node, and 1 output node). The input presented to the network consisted of eleven current and past values of the earthquake input excitation and ten past values of the actual structural response, whereas the target output value corresponded to the actual current structural response. A training set of 150 input-output pairs is taken at time interval Δt from the beginning of the El Centro earthquake input and structural response records, which corresponds to the first 3.20 seconds of the records. Figure 5 shows the root-mean-square (RMS) error and the Max error as a function of the number of learning cycles, a cycle being

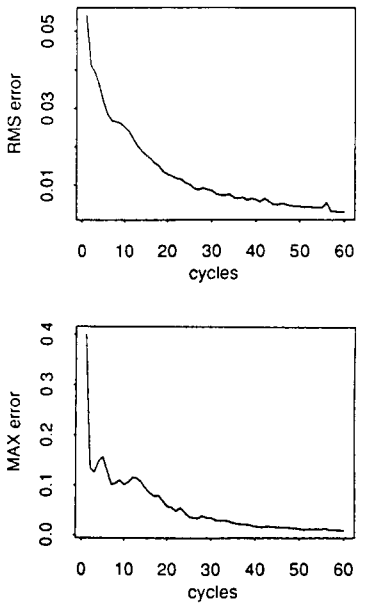

Figure 5. *Convergence of network learning (1 DOF case).*

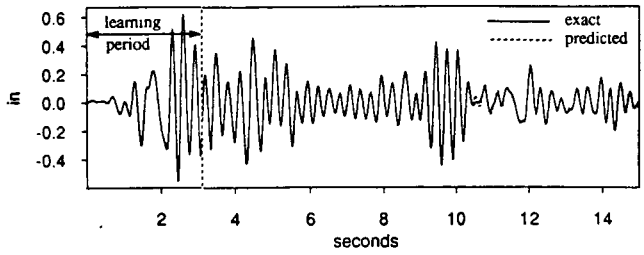

Figure 6. *Exact and predicted SDOF structural response to El Centro, 1940 (learning period: 3.20 sec).*

defined as a single learning loop over the entire training set. The RMS error and Max error are respectively defined to be $[1/Np\Sigma_{n=1}^{p}|\hat{\Upsilon}_n - \Upsilon_n|^2]^{1/2}$ and $\text{Max}_{n=1,\ldots,p}\|\hat{\Upsilon}_n - \Upsilon_n\|$ where p is the number of input-output pairs in the training set, $|\ldots|$ indicates the Euclidean norm (or L^2 norm), and $\|\ldots\|$ indicates the max norm (or L^∞ norm). Since the outputs are normalized in the interval [0.1–0.9], both error measurements are dimensionless quantities. The training of the neural network was completed in 59 cycles with a maximum error of less than 0.01. The structural response predicted by the trained network and the actual structural response are compared in Figure 6 for the first 15 seconds of the El Centro earthquake record. The network trained with the El Centro record is then recalled to predict the structural response to the Orion Blvd. record, the results of which are shown in Figure 7. Near perfect agreement between the predicted and actual earthquake responses indicate that the neural network is able to accurately learn the dynamics of the one-story building model.

The neural network model is next used to learn the dynamics of the two-story building. In this case, the network model consisted of 31 input nodes, 5 hidden nodes, and 2 output nodes (31-5-2 configuration). The 31 input nodes correspond to 11 current and past earthquake input values and 10 past output values for each floor level. The 2 output nodes represent the current response of the two floors. The network is trained over the first 10.20 seconds of the El Centro input and output records which comprise 500 input-output pairs taken at time interval Δt from the beginning of the records. The convergence of network learning is shown in Figure 8. The network is able to learn the dynamic behavior of the system within 30 training cycles with a Max

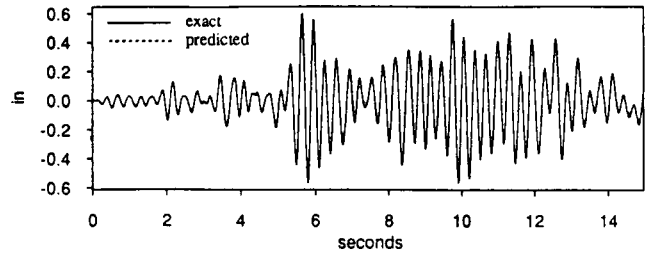

Figure 7. *Exact and predicted SDOF structural response to Orion Blvd., 1971 (network recall only).*

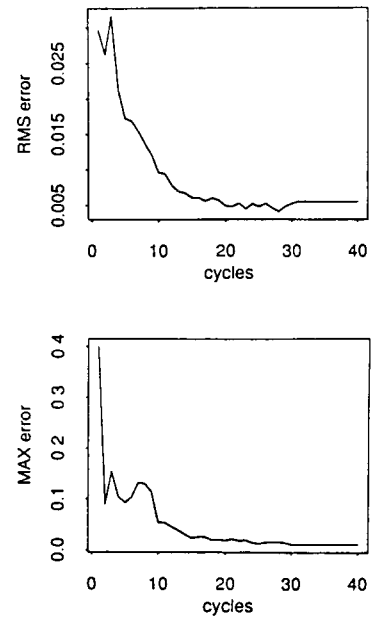

Figure 8. *Convergence of network learning (2 DOF case).*

error of 0.012. The floor displacements predicted by the neural network are compared with the actual displacements in Figure 9 for the entire duration of the El Centro record. As in the one-story case, the trained network is then used to predict the building response to the Orion Blvd. record, whose characteristics are very different from those of the El Centro record. The actual and predicted displacements for the two floors are compared in Figure 10. For both earthquake records, the agreement between predicted and actual response is excellent, indicating that the neural network has learned accurately the dynamics of the two-story building model as well.

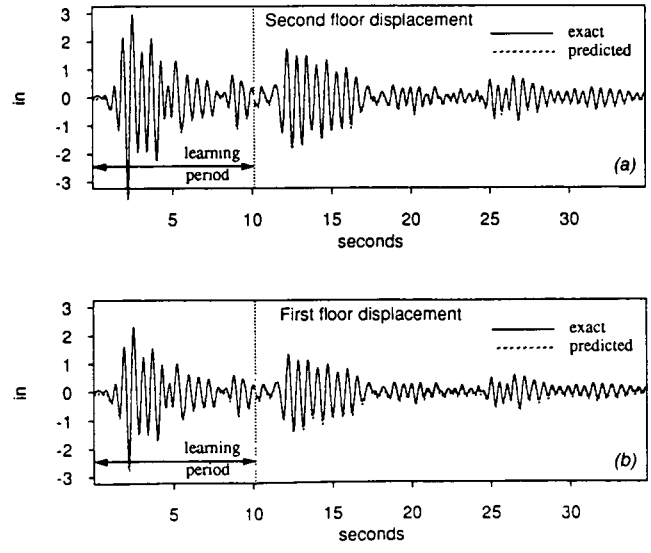

Figure 9. *Exact and predicted 2 DOF structural response to El Centro, 1940 (learning period: 10.20 sec).*

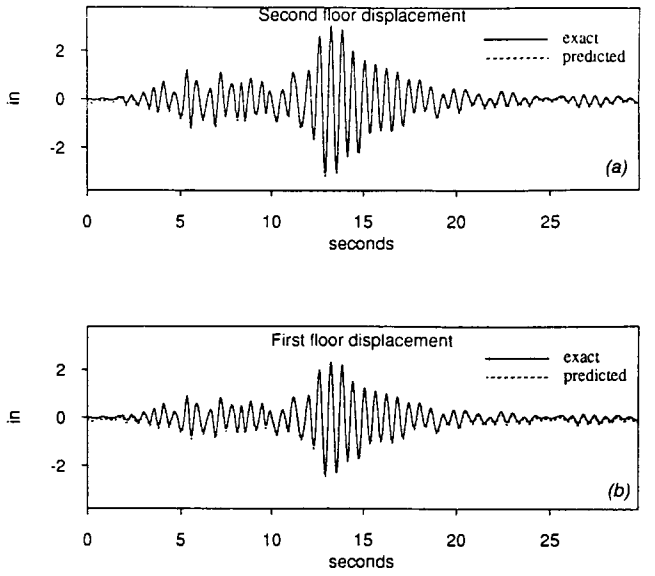

Figure 10. *Exact and predicted 2 DOF structural response to Orion Blvd., 1971 (network recall).*

The final network configuration was arrived at after an extensive experimentation with the number of input and output nodes. It was observed that as the number of input nodes (K_I and K_O) decreases (less history dependence of the system state is considered), more training cycles are required for a given error tolerance. However, there is a limit for the number of input nodes below which the network cannot be trained to satisfy a specified small error tolerance. Furthermore, there appeared to be an optimum number of input nodes below or above which the network prediction degraded. The number of hidden nodes had only a small effect on the accuracy of the network prediction and a larger number of hidden nodes did not necessarily improve the accuracy. No clear trend was found between the number of hidden nodes and the network learning, although for the two-story model there appeared to be an optimal number of hidden nodes.

With no hidden node, the neural network used in this study degenerates into a simple multi-variate linear regressor. In the case of the one-story model, it was found that the trained linear regressor is able to predict accurately the waveforms of the structural response, but with a low frequency departure from the actual response. This shift was completely eliminated with the addition of a single hidden neuron. In general, increasing the number of input-output pairs in the training set implied a reduction of the number of training cycles. The presence of the direct "linear connections" between the input and output layers improved the match between the network-predicted and the actual responses at the peaks and crests as compared to the prediction produced by a network wtihout these linear connections and with activation functions in the output nodes. In the latter case, the amplitude of the peaks and crests is systematically underestimated. However, the network with "linear connections" and no activation function in the output nodes can exhibit instability during recall.

The influence of the number of stories on the learning ability of the neural network model was investigated by considering a six-story building. The six-story structural model was selected to determine if the efficiency and accuracy of the network model obtained for the one- and two-story buildings could be maintained for a significantly larger number of stories. After several trials, the most satisfactory results were obtained with a network configuration of 71 input nodes, 5 hidden nodes, and 6 output nodes. The data presented to the 71 input nodes included 11 current and past values of the earthquake input and 10 past values of displacement response for each floor. The network was trained for the first 4.20 seconds of the El Centro record, which corresponds to 200 input-output training pairs. The network learning required approximately 200 cycles to converge to a Max error of 0.0165, as shown in Figure 11. The network-predicted and actual response time histories of the first and sixth floor for the El Centro record are compared in Figure 12. A similar comparison for the Orion Blvd. record for which the trained network is recalled only is shown in Figure 13. The agreement between predicted and actual responses is acceptable, but not as superior as for the one- and two-story buildings. The reduced accuracy of the network prediction can be explained from the dynamic behavior of multi-story systems. By examining the equation of motion of a multi-degree-of-freedom system, such as the one under consideration, it can be readily seen that the dynamic response of a given floor depends only on the state of the two adjacent floors and the earthquake ground acceleration. The inclusion of redundant information, such as past response of non-adjacent floors, in training the network only degrades

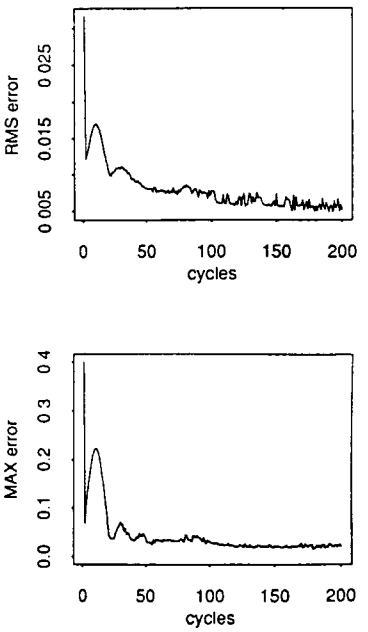

Figure 11. *Convergence of network learning (6 DOF case).*

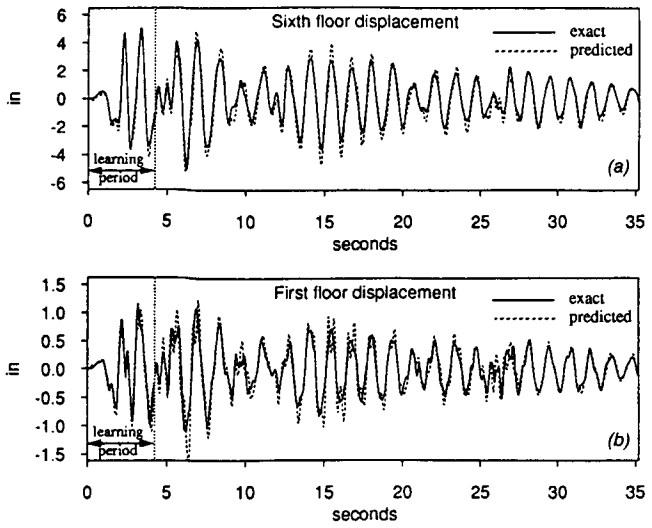

Figure 12. Exact and predicted 6 DOF structural response to El Centro, 1940 (learning period: 4.20 sec).

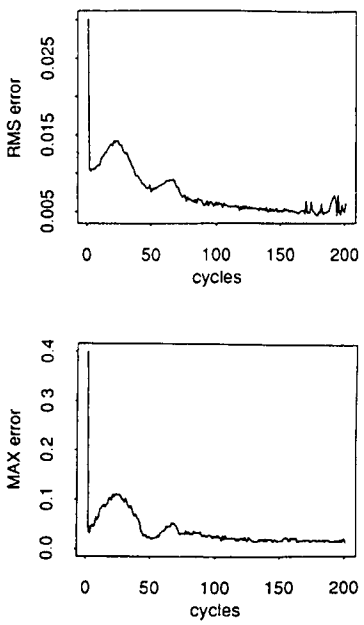

Figure 14. Convergence of network learning (second floor of 6 DOF case).

its learning ability. Based on this observation, a network modeling approach that involves learning the dynamics of a substructure at a time is devised. Following this approach, the number of input nodes were reduced to 41, which include the current and 10 past values of the earthquake ground excitation, 10 past response values for the floor under consideration, and 10 past values for each of the two adjacent floors. This approach gives the response of one floor at a time (one output node) and requires the availability of the past response of the adjacent floors. A generalization of this approach can provide the response at all floor levels by cascading neural network models of all substructures, thereby eliminating the need for past actual floor responses. As an illustration, the dynamics of the substructure consisting of the first, second, and third floors of the six-story building were learned using a 41-5-1 network configuration. The network learning convergence is shown in Figure 14, and the predicted and actual second floor responses are shown in Figures 15 and 16 for El Centro and Orion Blvd., respectively. These results show a marked improvement in the agreement between predicted and actual responses.

Previous discussion has focused on structural response to a single-input excitation. Next, the above neural network model is extended to include multiple-input excitation. The two-story structural frame discussed previously is subjected to excitations at each floor level in addition to the ground acceleration. The lateral floor excitations, shown in Figure 17, which are used here only as an illustration of a multiple-input case, correspond to the active control forces computed separately according to the optimal control theory for flexible structures (Meirovitch, 1990; Soong, 1990). The input to the neural network consisted of 11 current and past values of the ground acceleration, 10 past values of each floor excitation and response, whereas the network output comprised the 2 current floor responses. A network of configuration 51-5-2 was trained with 200 input-output pairs correspond-

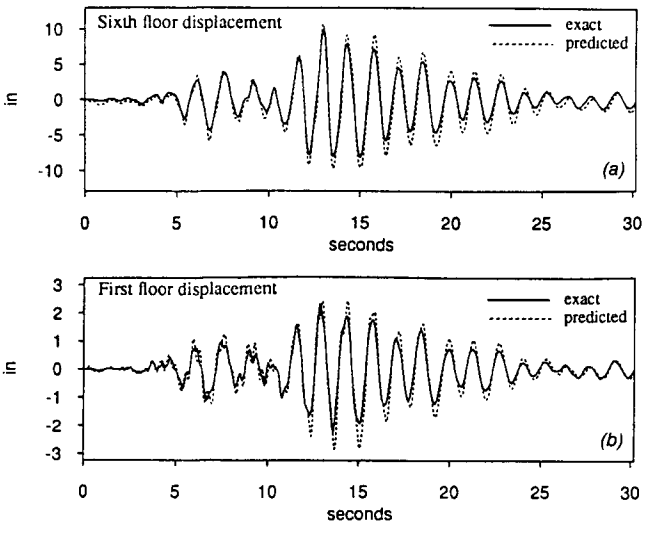

Figure 13. Exact and predicted 6 DOF structural response to Orion Blvd., 1971 (network recall).

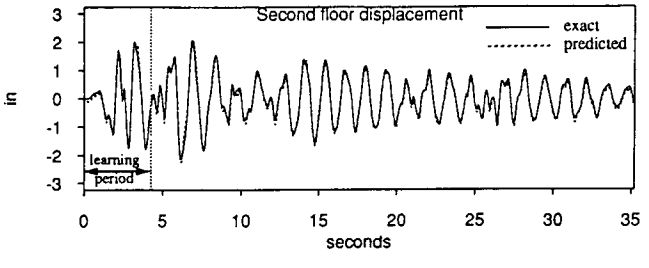

Figure 15. Exact and predicted second floor response of 6 DOF structural model to El Centro, 1940 (learning period: 4.20 sec).

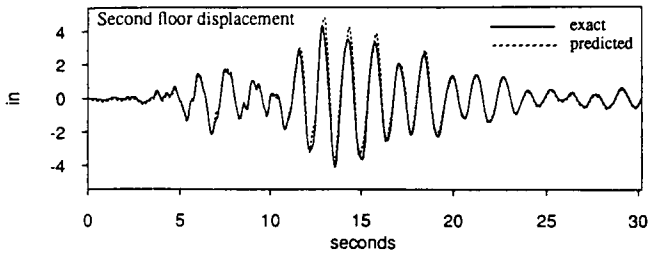

Figure 16. *Exact and predicted second floor response of 6 DOF structural model to Orion Blvd., 1971 (network recall).*

ing to the first 4.20 seconds of the El Centro record. The learning convergence rate of the network is shown in Figure 18. The training of the network was completed in 80 cycles with a Max error of 0.0118. The actual and predicted first- and second-floor displacement responses to the entire El Centro record are given in Figure 19. The network was then employed to predict the floor displacement response to the Orion Blvd. record and the corresponding predetermined optimal control forces at floor levels. The actual and predicted response time histories are presented in Figure 20. In both the El Centro and Orion Blvd. cases, the match between the network-predicted and the actual floor displacements is excellent. Based on these preliminary results, it is believed that with further refinement, neural network models could be equally effectively used in building structures subjected to multiple excitations.

CONCLUSIONS

The present study has shown that appropriately configured artificial neural networks can be effectively used in predicting the seismic response of multi-story buildings.

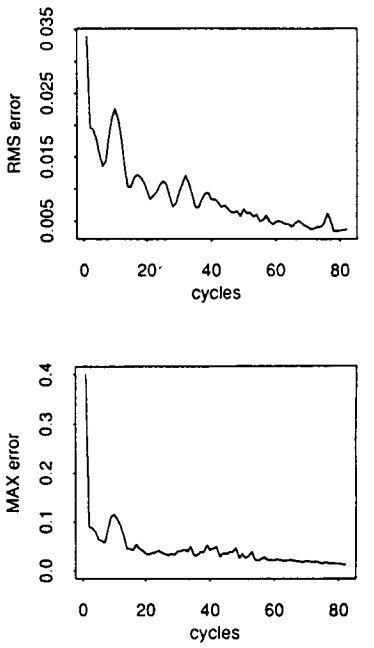

Figure 18. *Convergence of network learning (controlled 2 DOF case).*

The linear elastic dynamics of structural systems characterized by a few degrees of freedom and subjected to single or multiple excitations can be learned efficiently and accurately by means of simple three-layered neural networks in which the past response is fed back into the input nodes. The accuracy of the network prediction at the response peaks and crests is significantly improved by adding direct "linear connections" between the input and output layers and suppressing the activation function at the output nodes. For systems modeled with a large number of degrees of freedom, the ability of the basic three-layered network to pre-

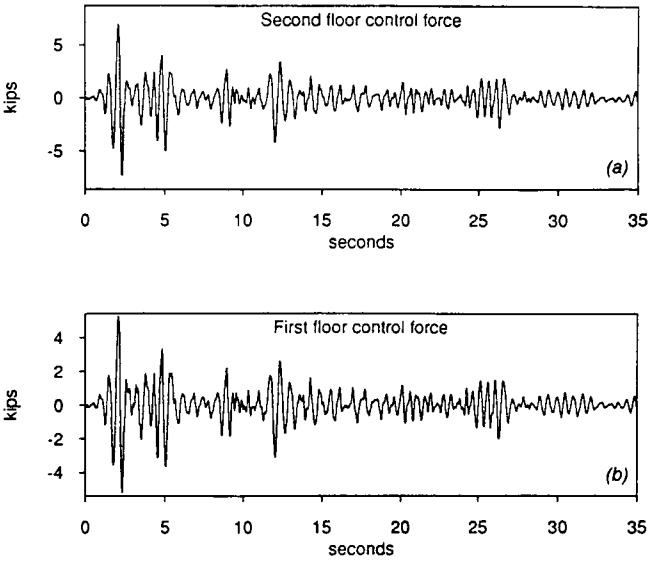

Figure 17. *Lateral control forces at floor levels (2 DOF case).*

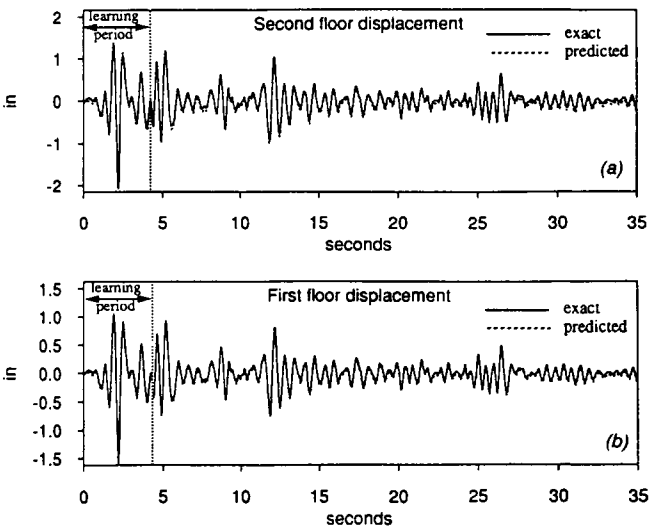

Figure 19. *Exact and predicted controlled 2 DOF structural response to El Centro, 1940 (learning period: 4.20 sec).*

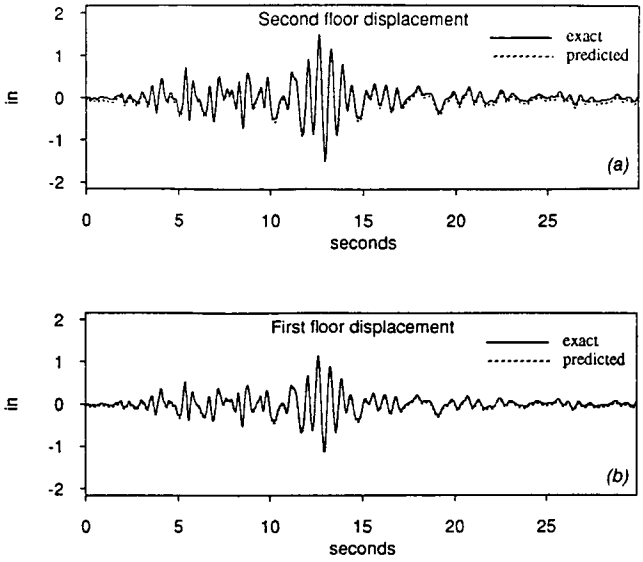

Figure 20. *Exact and predicted controlled 2 DOF structural response to Orion Blvd., 1971 (network recall).*

dict the response at a given degree of freedom is impeded when the past response of all degrees of freedom is fed back into the input layer. It is shown that learning efficiency and prediction accuracy are significantly improved by limiting the feedback from only the degrees of freedom that are adjacent to the one whose response is being predicted. This is not surprising as the dynamic response at each degree of freedom is goverened by the dynamic state of the adjacent degrees of freedom only. The three-layered network can thus be viewed as a basic building block of the neural network model of the entire structure, which is analogous to the substructure concept used in traditional structural analysis. The response at all degrees of freedom can then be predicted simultaneously by cascading the sub-networks together, which is the subject of continuing investigation.

REFERENCES

Agbabian, M. S., S. F. Masri, R. K. Miller and T. K. Caughey. 1991. "System Identification Approach to Detection of Structural Changes", *Journal of the Engineering Mechanics Division*, ASCE, 117(2).

Beck, J. L. and P. C. Jennings. 1980. "Structural Identification Using Linear Models and Earthquake Records", *Earthquake Engineering and Structural Dynamics*, 8:145–160.

Bialasiewicz, J. T. and T. T. Ho. 1991. "Neural Adaptive Identification and Control", *Proceedings of the 1991 International Conference on Artificial Neural Networks in Engineering*, St. Louis, Missouri, Nov. 10–12.

Bozich, D. J. and H. B. Mackay. 1991. "Vibration Cancellation Using Neurocontrollers", *Proceedings of the 1991 International Conference on Artificial Neural Networks in Engineering*, St. Louis, Missouri, Nov. 10–12.

Clough, R. W. and J. Penzien. 1993. *Dynamics of Structures, Second Edition*. McGraw-Hill.

Conte, J. P., R. O. Shelton and A. J. Durrani. 1992. "System Identification of Seismically Excited Structures Using Artificial Neural Networks", *Proceedings of the 1992 International Conference on Artificial Neural Networks in Engineering*, St. Louis, Missouri, Nov.

Elkordy, M. F., K. C. Chang and G. C. Lee. 1993. "Neural Networks Trained by Analytically Simulated Damage States", *Journal of Computing in Civil Engineering*, ASCE, 7(2).

Hoshiya, M. and E. Saito. 1984. "Structural Identification by Extended Kalman Filter", *Journal of the Engineering Mechanics Division*, ASCE, 110(12).

Kozin, F. and H. G. Natke. 1986. "System Identification Techniques", *Structural Safety*, 3:269–316.

Ljung, L. 1987. *System Identification—Theory for the User*. Prentice Hall.

Meirovitch, L. 1990. *Dynamics and Control of Structures*. John Wiley & Sons.

Nikolaou, M. and V. Hanagandi. 1991. "Input-Output Exact Linearization of Nonlinear Dynamical Systems Modeled by Recurrent Neural Networks", *Proceedings of the 1991 International Conference on Artificial Neural Networks in Engineering*, St. Louis, Missouri, Nov. 10–12.

Rehak, D. E., C. Thewalt and L. Doo. 1989. "Neural Network Approaches in Structural Mechanics Computations", *Computer Utilization in Structural Engineering, Structures Congress*, San Francisco, CA.

Rumelhart, D. E. and J. L. McClelland. 1988. *Parallel Distributed Processing, Volumes 1 and 2*. Cambridge, MA: MIT Press.

Singh, D. K., G. V. Kudav and S. A. Abdallah. 1991. "Modeling of Mechanical System Responses by Artificial Neural Networks", *Proceedings of the 1991 International Conference on Artificial Neural Networks in Engineering*, St. Louis, Missouri, Nov. 10–12.

Soong, T. T. 1990. *Active Structural Control—Theory and Practice*. Longman Scientific & Technical.

Wasserman, P. D. 1989. *Neural Computing—Theory and Practice*. New York: Van Nostrand Reinhold.

Wen, Y. K., J. Ghaboussi, P. Venini and K. Nikzad. 1992. "Control of Structures Using Neural Networks", *Proceeding of the U.S./Italy/Japan Workshop on Intelligent Structures*, Sorrento, Italy, July 13–14.