On-Line Learning Failure-Tolerant Neural-Aided Controller for Earthquake Excited Structures

Sriram Narasimhan, A.M.ASCE[1]; Sundaram Suresh[2]; Satish Nagarajaiah, M.ASCE[3]; and Narasimhan Sundararajan[4]

Abstract: This paper presents an on-line learning failure-tolerant neural controller capable of controlling buildings subjected to severe earthquake ground motions. In the proposed scheme the neural controller aids a conventional H_∞ controller designed to reduce the response of buildings under earthquake excitations. The conventional H_∞ controller is designed to reduce the structural responses for a suite of severe earthquake excitations using specially designed frequency domain weighting filters. The neural controller uses a sequential learning radial basis function neural network architecture called extended minimal resource allocating network. The parameters of the neural network are adapted on-line with no off-line training. The performance of the proposed neural-aided controller is illustrated using simulation studies for a two degree of freedom structure equipped with one actuator on each floor. Results are presented for the cases of no failure and failure of the actuator on each of the two floors under several earthquake excitations. The study indicates that the performance of the proposed neural-aided controller is superior to that of the H_∞ controller under no actuator failure conditions. In the presence of actuator failures, the performance of the primary H_∞ controller degrades considerably, since actuator failures have not been considered for the design. Under these circumstances, the neural-aided controller is capable of controlling the acceleration and displacement structural responses. In many cases, using the neural-aided controller, the response magnitudes under failure conditions are comparable to the performance of the H_∞ controller under no-failure conditions.

DOI: 10.1061/(ASCE)0733-9399(2008)134:3(258)

CE Database subject headings: Active control; Neutral networks; Failures; Earthquake resistant structures.

Introduction

The research in the field of active structural control during the last few decades has evolved into two general streams. On one hand, controllers have been designed using optimal control theories such as linear quadratic Guassian (LQG) or H_∞, and on the other hand controllers have been developed using techniques such as neural networks and fuzzy sets. Both these areas have demonstrated tremendous potential in addressing the problem of structural control under severe earthquake excitations. While the advantages are obvious, certain limitations have to be recognized in using optimal controllers and neural controllers. For example, an accurate system representation is essential for designing optimal controllers and the controller cannot accommodate sudden changes such as actuator or sensor failures. Neural networks designed using extensive off-line training cannot provide satisfactory performance under situations that have not been anticipated in the training phase. The main objective of this paper is to present a controller that consists of and draws upon the strengths of both the optimal and neural components. Several advantages of the proposed controller architecture are presented in this paper. These advantages include no a priori training for the neural network and failure-tolerance of the overall controlled system to partial actuator system failures.

Neural networks have been studied extensively in the areas of control and system identification of dynamical systems (Narendra and Parthasarathy 1990; Masri et al. 2000). There have been several studies on the application of neural networks for active control of structures subjected to earthquake excitations (Bani-Hani and Ghaboussi 1998; Chen et al. 1995; Ghaboussi and Joghataie 1995; Liut et al. 1999). The neural network architectures in the active structural control studies involve a back-propagation neural network (BPNN) that utilizes the error between the network output and the desired output to update the network parameters. One obvious limitation for structural control applications is the lack of information regarding the desired control output. In order to circumvent this problem, several strategies such as using an emulator network (Ghaboussi and Joghataie 1995) or a force-matching procedure (Liut et al. 1999) have been proposed. In these neural network controllers off-line training is required and the neural controller so designed is a stand-alone controller replacing the control algorithm. In addition, the size of the network is fixed beforehand. A neural network that does not involve BPNN has recently been proposed for active control of seismically excited

[1]Assistant Professor, Dept. of Civil & Environmental Engineering, 200 University Ave. W., Univ. of Waterloo, ON, Canada N2L 3G1 (corresponding author). E-mail: snarasim@uwaterloo.ca

[2]Research Fellow, School of Electrical and Electronic Engineering, Nanyang Technological Univ., Singapore. E-mail: ssundaram@ntu.edu.sg

[3]Professor, Depts. of Civil & Environmental Engineering, and Mechanical Engineering & Material Science, Rice Univ., 6100 S. Main St., Houston, TX 77005. E-mail: nagaraja@rice.edu

[4]Professor, School of Electrical and Electronic Engineering, Nanyang Technological Univ., Singapore. E-mail: ensundara@ntu.edu.sg

Note. Associate Editor: Erik A. Johnson. Discussion open until August 1, 2008. Separate discussions must be submitted for individual papers. To extend the closing date by one month, a written request must be filed with the ASCE Managing Editor. The manuscript for this paper was submitted for review and possible publication on July 12, 2006; approved on August 7, 2007. This paper is part of the ***Journal of Engineering Mechanics***, Vol. 134, No. 3, March 1, 2008. ©ASCE, ISSN 0733-9399/2008/3-258–268/$25.00.

structures (Madan 2005). This neural network is based on a counterpropagation network (Hecht-Nielsen 1987; Sirca and Adeli 2004). The feature sensitive hidden neuron centers are calculated using the unsupervised clustering algorithm without the aid of target control actions and the desired control signal is learned from on-line training. System failures such as actuator or sensor failures are not considered in all the aforementioned studies. For problems involving unknown excitations such as earthquakes, there are several advantages of using a neural network architecture that do not require training a priori. If the network parameters such as number of neurons in the hidden layer are a function of the complexity of the problem at hand, the burden of fixing the size of the network in advance does not exist. Finally, recognizing that a typical actively controlled building will contain several actuators and sensors, and associated with each is a finite failure probability, the controller must be tolerant to partial failure of these systems. This paper presents an on-line learning neural-aided H_∞ controller that requires no a priori training, whose parameters are adaptive and tolerant to partial actuator system failures.

Several active control strategies have been developed and shown to be effective for structural control (Narasimhan et al. 2006; Nagarajaiah and Narasimhan 2006; Narasimhan and Nagarajaiah 2006; Spencer and Nagarajaiah 2003; Yang et al. 2004). They include frequency domain controllers such as H_2 and H_∞ (Doyle et al. 1989; Yang et al. 2004) based on robust optimal control theory and typically involve a minimization of a norm of a transfer function (or a quadratic index in time domain) between the input excitation and the outputs to be regulated. H_2/LQG and H_∞ control strategies are robust for small perturbations in the system parameters, but cannot accommodate partial system failures such as loss of actuators or sensors. Since the aforementioned optimal control strategies have been adopted and widely used, it is attractive to develop a neural controller that augments the performance of the existing optimal controller, instead of replacing it completely. Such a control architecture can also be used when an existing control design is already in place.

System failures occur when components such as actuators malfunction, compared to system faults which include improper design (Stengel 1991). Since an actively controlled building consists of several actuators distributed over the entire structure area, it is conceivable that some actuators may malfunction during or prior to an earthquake event. Under conditions of system failures, the performance of the primary controller is compromised. In many cases, it may not be practical to provide redundant sets of actuators at each location throughout the structure. Actuator failures have been studied in the context of structural health monitoring and fault detection (Koh et al. 2005). Ankireddi and Yang (1991) proposed neural network architectures to detect and accommodate sensor failures (although outside the scope of the current study, sensor failures are another important class of system failures) and used it to improve the performance of a closed loop system. In the aforementioned study the neural network detects sensor failure and estimates the missing states, which in turn is used by the primary controller for feedback control. Prior training is needed for network performance and only sensor failures (and not actuator failures) are investigated in the aforementioned study.

Radial basis functions (RBFs) neural networks with Gaussian hidden units have been shown to possess good interpolation and generalization ability (Broomhead and Lowe 1988; Moody and Darken 1989). A sequential-learning RBF network called the minimal resource allocation network (MRAN) was developed by Lu et al. (1997, 1998). MRAN starts with a zero hidden neuron and has a growing, pruning, and parameter updating strategy using an extended Kalman filter (EKF) approach. For making MRAN faster for real-time applications, an extension to MRAN called extended MRAN (EMRAN) was developed by Li (Li et al. 2000). This network has been used for several applications such as function approximation and nonlinear system identification (Li et al. 2000). The main advantage of this architecture is its sequential learning ability. Hidden neurons are both added and removed based on the input patterns and learning mechanism to maintain a compact and a high performance network. EMRAN has been used in the context of auto-landing for fault-tolerant aircraft control applications (Li et al. 2004; Pashlikar et al. 2005). The system dynamics and the control objectives in an aircraft control problem are different from those in seismically excited structures. This is due to the nature of the control problem, namely, tracking versus regulation, and the characteristics of the input disturbances. In this study the neural controller aids a H_∞ primary controller to reduce the structural responses during both no-failure and under actuator failure conditions. Simulation results are presented for a suite of strong near-field historic earthquake records.

The paper is organized as follows. First, the structural system model subjected to earthquake excitation is presented. Then, the overall on-line controller architecture is outlined. Details of EMRAN are then presented along with the optimization procedure based on genetic algorithm (GA) used to derive the parameters of EMRAN. Finally, the on-line failure-tolerant learning neural network controller is presented followed by the simulation results for an example of a two degree of freedom structure model.

Structural System Model

The state space equations for an actively controlled linear planar structure having n floors with one degree of freedom per floor subjected to uniaxial earthquake excitation at the base can be written as follows:

$$\dot{\mathbf{x}}_{(2n\times 1)}(t) = \mathbf{A}_{(2n\times 2n)}\mathbf{x}_{(2n\times 1)}(t) + \mathbf{B}_{(2n\times n)}\mathbf{F}^{\text{act}}_{(n\times 1)} + \mathbf{E}_{(2n\times 1)}\ddot{U}_g(t) \tag{1}$$

where $\mathbf{x}$ consists of the states which are the displacements and velocity with respect to the ground; $\mathbf{F}^{\text{act}}(t)$=active control force; $\ddot{U}_g(t)$=ground acceleration; and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{E}$=system matrices that are defined as follows:

$$\mathbf{A}_{(2n\times 2n)} = \begin{bmatrix} \mathbf{0}_{(n\times n)} & \mathbf{I}_{(n\times n)} \\ -\mathbf{M}^{-1}_{(n\times n)}\mathbf{K}(n\times n) & -\mathbf{M}^{-1}_{(n\times n)}\mathbf{C}_{(n\times n)} \end{bmatrix}$$

$$\mathbf{B}_{(2n\times n)} = \begin{bmatrix} \mathbf{0}_{(n\times n)} \\ \mathbf{M}^{-1}_{(n\times n)}\mathbf{\Lambda}(n\times n) \end{bmatrix}$$

and

$$\mathbf{E}_{(2n\times 1)} = -\begin{bmatrix} \mathbf{0}_{(n\times 1)} \\ \mathbf{\Gamma}_{(n\times 1)} \end{bmatrix}$$

In the above equations, $\mathbf{M}$=diagonal mass matrix; $\mathbf{C}$=damping matrix; and $\mathbf{K}$=structure stiffness matrix. The earthquake excitation represented by $\ddot{U}_g$ excites the base of the structure and the vector of earthquake influence coefficients, $\mathbf{\Gamma}$, consists of those corresponding to all the floors. If actuators are

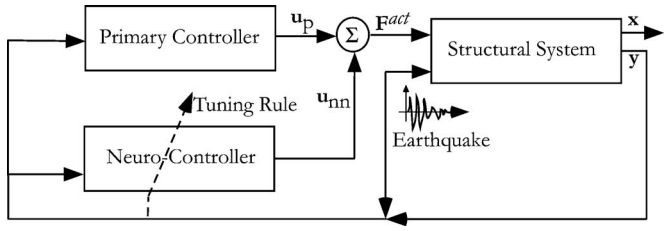

Fig. 1. Neural-aided H_∞ controller architecture

assumed on all floors, the matrix $\mathbf{\Lambda}$=identity matrix with those on the main diagonal corresponding to the actuator force locations at the degrees of freedom.

Design of On-Line Learning Neural Controller

The control strategy proposed in this paper is a variation of the feed-back error learning scheme proposed by others (Li et al. 2004; Gomi and Kawato 1990). The architecture of the on-line learning neural-aided controller is shown in Fig. 1. The proposed control architecture consists of an H_∞ primary controller in the inner loop and an on-line learning neural controller in the outer loop. The on-line learning neural controller augments the performance of the primary controller in order to achieve better structural performance and failure-tolerance. The neural network is based on an on-line learning architecture using RBF networks (Li et al. 2000, 2004) called EMRAN. Using a sequential learning algorithm, the network adds and prunes the number of neurons in the hidden layer depending on the input characteristics. This flexibility of the network results in compactness that is suited for real-time applications. The details of the primary and neural controller designs are presented in the following sections.

Design of H_∞ Primary Controller

The equations of motion presented in Eq. (1) is cast in the standard state space form (Doyle et al. 1989) as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_1\mathbf{w} + \mathbf{B}_2\mathbf{F}^{\text{act}}$$

$$\mathbf{z} = \mathbf{C}_1\mathbf{x} + \mathbf{D}_{11}\mathbf{w} + \mathbf{D}_{12}\mathbf{F}^{\text{act}}$$

$$\mathbf{y} = \mathbf{C}_2\mathbf{x} + \mathbf{D}_{21}\mathbf{w} + \mathbf{D}_{22}\mathbf{F}^{\text{act}} \tag{2}$$

In Eq. (2), $\mathbf{y}$ and $\mathbf{z}$=measurements and the outputs to be regulated; and $\mathbf{D}_{11}$, $\mathbf{D}_{12}$, $\mathbf{D}_{21}$, $\mathbf{D}_{22}$, $\mathbf{C}_1$, and $\mathbf{C}_2$=mapping matrices of appropriate dimensions. The measurement outputs are represented by $\mathbf{y}$, the outputs to be regulated by $\mathbf{z}$, and $\mathbf{w}=[\ddot{U}_g\ \boldsymbol{\nu}]$; where $\boldsymbol{\nu}$=measurement sensor noise. The purpose of the H_∞ control method is to minimize the ∞ norm of the transfer function from disturbance input to regulated output, $\mathbf{G}_{zw}$, and is written as

$$\|\mathbf{G}_{zw}(s)\|_\infty = \sup_\omega \|\bar{\sigma}(\mathbf{G}_{zw}(s))\| \leqslant \gamma \tag{3}$$

$\bar{\sigma}$=largest singular value of the transfer function; sup denotes the supremum; and γ=positive bound for the norm. The solution for the controller for the generalized regulator problem (Doyle et al. 1989; Green and Limebeer 1995) is given by

$$\mathbf{u} = -\mathbf{F}_\infty\hat{\mathbf{x}} \tag{4}$$

and the state estimator is given by

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}_2\mathbf{F}^{\text{act}} + \mathbf{B}_1\hat{\mathbf{w}} + \mathbf{J}_\infty\mathbf{L}_\infty(\mathbf{y} - \hat{\mathbf{y}}) \tag{5}$$

where

$$\hat{\mathbf{w}} = \gamma^{-2}\mathbf{B}_1^T\mathbf{K}_\infty\hat{\mathbf{x}}$$

and

$$\hat{\mathbf{y}} = \gamma^{-2}\mathbf{D}_{21}\mathbf{B}_1^T\mathbf{K}_\infty\hat{\mathbf{x}} + \mathbf{C}_2\hat{\mathbf{x}}$$

The terms $\hat{\mathbf{w}}$ and $\hat{\mathbf{y}}$=estimates of the worst case disturbance and output of the estimator. There exists a stabilizing controller if and only if there exists positive semidefinite solutions to the two Riccatti equations for $\mathbf{K}_\infty$ and $\mathbf{N}_\infty$ and the condition

$$\rho(\mathbf{K}_\infty\mathbf{N}_\infty) < \gamma^2 \tag{6}$$

where $\rho(\mathbf{A})$=spectral radius of $\mathbf{A}$ which is defined as the largest singular value of $\mathbf{A}$. The controller written in the packed matrix notation is

$$\mathbf{K}_{\text{sub}}(s) = \begin{bmatrix} \hat{\mathbf{A}}_\infty & \mathbf{J}_\infty\mathbf{L}_\infty \\ -\mathbf{F}_\infty & \mathbf{0} \end{bmatrix} \tag{7}$$

where

$$\mathbf{F}_\infty = (\mathbf{D}_{12}^T\mathbf{D}_{12})^{-1}(\mathbf{B}_2^T\mathbf{K}_\infty + \mathbf{D}_{12}^T\mathbf{C}_1)$$

$$\mathbf{L}_\infty = (\mathbf{N}_\infty\mathbf{C}_2^T + \mathbf{B}_1\mathbf{D}_{21}^T)(\mathbf{D}_{21}\mathbf{D}_{12}^T)^{-1}$$

and

$$\mathbf{J}_\infty = (\mathbf{I} - \gamma^{-2}\mathbf{N}_\infty\mathbf{K}_\infty)^{-1}$$

The terms $\mathbf{K}_\infty$ and $\mathbf{N}_\infty$=solutions to the controller and estimator Riccati equations given by

$$\mathbf{K}_\infty = \text{Ric}\begin{pmatrix} \mathbf{A} - \mathbf{B}_2\tilde{\mathbf{D}}_{12}\mathbf{D}_{12}^T\mathbf{C}_1 & \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^T - \mathbf{B}_2\tilde{\mathbf{D}}_{12}\mathbf{B}_2^T \\ -\tilde{\mathbf{C}}_1^T\tilde{\mathbf{C}}_1 & -(\mathbf{A} - \mathbf{B}_2\tilde{\mathbf{D}}_{12}\mathbf{D}_{12}^T\mathbf{C}_1)^T \end{pmatrix} \tag{8}$$

$$\mathbf{N}_\infty = \text{Ric}\begin{pmatrix} (\mathbf{A} - \mathbf{B}_1\mathbf{D}_{21}\tilde{\mathbf{D}}_{21}\mathbf{C}_2)^T & \gamma^{-2}\mathbf{C}_1^T\mathbf{B}_1 - \mathbf{C}_2^T\tilde{\mathbf{D}}_{21}\mathbf{C}_2 \\ -\tilde{\mathbf{B}}_1\tilde{\mathbf{B}}_1^T & -(\mathbf{A} - \mathbf{B}_1\tilde{\mathbf{D}}_{21}^T\tilde{\mathbf{D}}_{21}\mathbf{C}_2) \end{pmatrix} \tag{9}$$

where

$$\tilde{\mathbf{C}}_1 = (\mathbf{I} - \mathbf{D}_{12}\tilde{\mathbf{D}}_{12}\mathbf{D}_{12}^T)\mathbf{C}_1$$

$$\tilde{\mathbf{B}}_1 = \mathbf{B}_1(\mathbf{I} - \mathbf{D}_{21}^T\tilde{\mathbf{D}}_{21}\mathbf{D}_{21})$$

$$\tilde{\mathbf{D}}_{12} = (\mathbf{D}_{12}^T\mathbf{D}_{12})^{-1};\ \tilde{\mathbf{D}}_{21} = (\mathbf{D}_{21}\mathbf{D}_{21}^T)^{-1}$$

In order to achieve good controller performance, weighting filters for the input and outputs are introduced (Nagarajaiah and Narasimhan 2006; Narasimhan and Nagarajaiah 2006) into the formulation. The input excitation filter is used to provide information to the controller regarding the frequency content of the earthquake excitations (Narasimhan and Nagarajaiah 2006), and is represented in the frequency domain as follows

$$F(s) = \frac{8\zeta_g^2\omega_g^2 s^2}{[s^2 + 2\zeta_g\omega_g s + \omega_g^2]^2} \tag{10}$$

where ω_g=2π rad/s and ζ_g=0.6. The outputs are weighted using a first-order filter of the form, $W=a/(s+a)$, where, a=8.5 rad/s, which determines the rolloff frequency. The output filter is intended to regulate the responses in the fundamental mode of the structure.

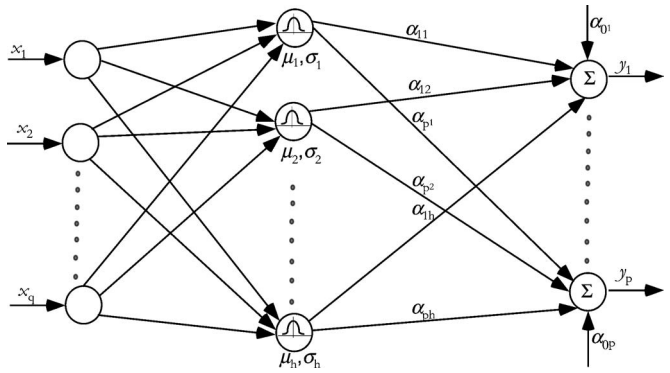

Fig. 2. Radial basis function network

The state space equations shown in Eq. (2) are augmented using both the input and output weighting filters and the H_∞ controller is designed for the augmented system (Narasimhan and Nagarajaiah 2006). The measurements contain accelerations on all floors and the earthquake excitation. The regulated outputs contain the interstory drifts and floor accelerations. For a more comprehensive treatment of H_∞ controller design for seismic control, the readers are referred to a previous study (Narasimhan and Nagarajaiah 2006). The computations involving the controller and estimator gains are performed using MATLAB (MATLAB 2000).

Neural Controller Design

In this section the neural network algorithm based on EMRAN is described followed by the neural network control architecture and the optimization procedures. EMRAN is a high-performance sequential learning algorithm for implementing a minimum RBF network developed by Li et al. (2000). EMRAN is a fully adaptive network with addition and pruning capabilities for the number of hidden neurons. The number of hidden neurons is a function of the complexity of the input signal and is not fixed at the start of the problem. The network inputs consist of the acceleration and displacement responses (obtained through integration from the accelerations) of the structure at the previous time-step. The network learns to control the structure in a sequential manner based on the inputs and the structure responses. The network parameters are updated based on structural responses without off-line training. The sequential learning ability is the major advantage in using the proposed network for on-line implementation.

A typical RBF network that consists of one hidden layer with connecting weights between the hidden layer and the output is shown in Fig. 2. The basis functions are Gaussian with center μ_k and width σ_k for a typical hidden neuron k, $k=1,2,\cdots,h$. Each output y_j, where $j=1,2,\cdots,p$=total outputs in the network, to the network inputs $\mathbf{x}_i$ at the ith instant can be written as

$$y^j = \alpha_{0j}[i] + \sum_{k=1}^{h} \alpha_{jk}[i]\phi_k(\mathbf{x}_i) \tag{11}$$

where α_{0j} and α_{jk} represent the bias and the connection weight to the jth output. The function $\phi(\mathbf{x}_i)$=Gaussian function represented by

$$\phi_k(\mathbf{x}_i) = \exp(-\|\mathbf{x}_i - \boldsymbol{\mu}_k[i]\|^2/(\sigma_k[i])^2) \tag{12}$$

The sequential learning network starts out with no hidden neurons and a new hidden neuron is added when the following criteria are met:

$$\|\mathbf{x}_i - \boldsymbol{\mu}_w[i]\| > \epsilon_i \tag{13}$$

where $\epsilon_i = \max[\epsilon_{\max} r^{i-1}, \epsilon_{\min}]$. Let

$$\mathbf{e}_i = \mathbf{y}_i - f(\mathbf{x}_i) \tag{14}$$

where $f(\mathbf{x}_i)$=desired output at the ith instant; and $\boldsymbol{\mu}_w$=center of the hidden unit closest ("winner" neuron) to $\mathbf{x}_i$. $\epsilon_{\max}$, $\epsilon_{\min}$, and r=constants that are determined using optimization procedures (to be discussed in the following section). The following additional criterion based on the root mean square error for a window of samples (S_w) is required to add a new hidden neuron:

$$e_{\text{rmse}} = \sqrt{\frac{\sum_{i=n-S_w+1}^{n} \mathbf{e}_i^2}{S_w}} \geqslant e_{\min} \tag{15}$$

where $e_{\min}$=minimum acceptable error threshold. The parameters for the added unit are given by

$$\boldsymbol{\alpha}_{h+1}[i] = \mathbf{e}_i, \quad \boldsymbol{\mu}_{h+1}[i] = \mathbf{x}_i, \quad \sigma_{h+1}[i] = \kappa\|\mathbf{x}_i - \boldsymbol{\mu}_w[i]\| \tag{16}$$

where κ=overlap factor that determines the overlap of the responses of the hidden units in the input space. When the criteria for adding a new hidden unit are not met, the parameters of the network, namely the connection weights, biases, centers, and widths of the network are updated for the ith step based on the values from the previous step using the EKF algorithm. In order to achieve faster performance for on-line applications, instead of updating all the network parameters using EKF, only the parameters of the neuron whose center is closest (in a norm sense) to the network input data $\mathbf{x}_i$ are updated. The closest neuron is called the "winner" neuron whose parameters ($\mathbf{NP}$)=connection weights, centers, and widths. The updated parameters of the winner neuron at the i sample are

$$\mathbf{NP}^w[i] = \mathbf{NP}^w[i-1] + \mathbf{K}^w[i]\mathbf{e}_i \tag{17}$$

where the Kalman gain matrix $\mathbf{K}^w[i]$ is given by

$$\mathbf{K}^w[i] = \mathbf{P}^w[i-1]\mathbf{B}^w[i](\mathbf{R}[i] + \mathbf{B}^w[i]^T\mathbf{P}^w[i-1]\mathbf{B}^w[i])^{-1} \tag{18}$$

where $\mathbf{B}^w[i] = \nabla_{\mathbf{w}^w} f(\mathbf{x}_i)$=gradient matrix of the function $f(\mathbf{x}_i)$ with respect to the parameter vector $\mathbf{w}^w[i]$ evaluated at $\mathbf{w}^w[i-1]$ and $\mathbf{R}[i]$=variance of the measurement noise. $\mathbf{P}^w[i]$=error of the covariance matrix, which is updated by

$$\mathbf{P}^w[i] = (\mathbf{I} - \mathbf{K}[i]\mathbf{B}^{wT}[i])\mathbf{P}^w[i-1] + q\mathbf{I} \tag{19}$$

where q=scalar that determines the allowed random step in the direction of the gradient vector. When the new hidden neuron is added to the network, the covariance matrix dimensionality increases to

$$\mathbf{P}^w[i] = \begin{bmatrix} \mathbf{P}^w[i-1] & \mathbf{0} \\ \mathbf{0} & P_0\mathbf{I} \end{bmatrix}$$

where P_0=scalar value representing the uncertainty in the initial parameters of the new hidden neuron.

In order to maintain a compact network, a pruning strategy is incorporated in the algorithm. Pruning of neurons ensures that the neurons that have been added in the past and have not been contributing significantly (based on a threshold parameter δ) to the network performance for a predefined period of time (N_w), are removed from the network. This results in a network that is computationally efficient and adapted to a fast on-line implementation. The on-line learning algorithm of EMRAN can be summarized in the following steps:

1. Compute the overall network output

$$y^j = \alpha_{0j}[i] + \sum_{k=1}^{h} \alpha_{jk}[i]\phi_k(\mathbf{x}_i) \tag{20}$$

2. Calculate the parameters required in the growth criterion

$$\epsilon_i = \max[\epsilon_{\max} r^{i-1}, \epsilon_{\min}] \tag{21}$$

$$\mathbf{e}_i = \mathbf{y}^i - f(\mathbf{x}_i) \tag{22}$$

Calculate the root mean square error for a given window of samples (S_w)

$$e_{\text{rmse}} = \sqrt{\frac{\sum_{i=n-S_w+1}^{n} \mathbf{e}_i^2}{S_w}} \tag{23}$$

3. Apply the criterion ($\|\mathbf{x}_i - \boldsymbol{\mu}_w[i]\| > \epsilon_i$ and $e_{\text{rmse}} => e_{\min}$) for adding or updating the network parameters. If the criteria are satisfied then add a new hidden neuron. Otherwise, update the network parameters for a winner neuron using the EKF algorithm.
4. Calculate the criterion $r^n < \delta$ (r^n=normalized contribution of the nth neuron for N_w subsequent samples) for pruning and delete the noncontributing neuron and reduce the dimensionality of EKF parameters.
5. Repeat steps 1–4 for all input data sequentially.

GA Optimization to Determine EMRAN Constants

For effective on-line learning, appropriate values for the following constants in EMRAN algorithm are determined through a GA optimization procedure: forward gain of the neural controller K_F, $\epsilon_{\max}$, $\epsilon_{\min}$, $e_{\min}$, r, κ, N_w, S_w, δ, P_0, q. The values for the aforementioned quantities are determined once for a given system and remain unchanged during simulations. In order to perform the optimization, the two degree of freedom structure used for the simulation study is excited by the fault-normal component of 1994 Northridge earthquake-Sylmar county record. The aforementioned excitation is not used to assess the controller performance subsequently. The controller parameters such as number of hidden neurons, weights, and centers are calculated on-line without a priori training.

The underlying cost function that minimizes the structural responses is not a smooth function of the network variables. Furthermore, due to the large number of variables it is computationally expensive to use gradient based search techniques. Under such situations, the GA based search methods are more suitable. GAs were developed by Holland (1975) in an attempt to explain the adaptive processes of natural systems and to design artificial systems based upon these natural systems. GA is a search algorithm based on the mechanism of natural selection that transforms a set of individuals (population of fixed length binary strings) into a new population (i.e., the next generation) using genetic operators such as crossover and mutation (Holland 1975; Goldberg 1989; Michalewicz 1994).

For any given optimization problem, one has to address the process of encoding the decision variable as a string and calculate its fitness value. The string representation is the process of encoding a potential search node (decision variables) as a string. The string representation not only depends on the structure of the problem in GA framework but also depends on genetic operators used in the algorithm. In earlier studies of genetic algorithms (Holland 1975; Goldberg 1989), the solutions were coded using binary representation. It was shown (Michalewicz 1994) that for numerical optimization problems floating point representation of solutions performs better than binary representation because it is more precise, more consistent, and lead to faster convergence. Hence, in this study, EMRAN constants are represented using real numbers. Reasonable values are chosen for upper and lower bounds for the constants in GA implementation. The GA optimizer searches between these bounds for the optimal solution. In genetic algorithm, the initial population of N search nodes is generated using the most common random generation procedure. The population size (N) is typically problem dependent and has to be determined through simulation. The fitness function typically consists of an objective function to be minimized.

The genetic operators maximize the fitness function, which is the negative of the root-mean-square error (RMSE) of structure response accelerations. The real-coded genetic algorithm based variable selection was implemented in MATLAB on a Pentium-IV machine. The GA variables used in the simulations are: mutation probability of 0.15; crossover probability of 0.6; selection probability of 0.08; maximum number of generations of 500; and population size of 30. It was observed that the performance
criteria were generally satisfied at around ten generations. The details of selection of GA parameters are outside the scope of the discussion in this paper and the readers are referred to Houck et al. (1995) for a detailed description. A general purpose GA optimization toolbox (Houck et al. 1995) is used to perform the simulations for the current study.

Implementation of On-Line Learning Neural Controller

The SIMULINK (MATLAB 2000) block diagram of the proposed on-line learning neural controller with primary H_∞ controller is shown in Fig. 3. The proposed control architecture consists of the H_∞ primary controller in the inner loop, and the EMRAN controller in the outer loop aiding the primary controller. Recognizing that the ultimate goal of the controller is to drive the response of the system as close to zero as possible, the structure floor acceleration, ground acceleration, and displacement responses are used for tuning the neural controller. The displacement responses are derived from the acceleration responses through numerical integration (Nagarajaiah and Narasimhan 2006). The network starts with no hidden neurons and quickly builds up the size of the network in response to seismic excitations during periods of intense shaking and prunes the number of neurons after the passage of the peak intensity periods of the earthquake. This results in a compact network that is suited for high performance on-line applications. The EMRAN constants obtained using GA optimization are listed in Table. 1.

An important aspect that needs to be considered in on-line control is the time delay due to the control force calculations. The primary advantage in this control architecture is that only the parameters of the nearest hidden neuron are updated. The computations required to update the parameters of one hidden neuron is $O(n^3)$, where n=number of parameters for the nearest neuron (center, width, and weights=8), which means the total computational burden at each step is 8^3 floating point operations (FLOPS) (1 FLOP=$1e^{-06}$ s), which is less than the sampling time of 0.005 s). Control output calculation (forward pass) is $O(h)$, where h=number of hidden neurons. The main advantage of this network architecture is to keep h minimal, resulting in a compact network. Hence, the forward computational burden is minimal and is less than 0.005 s, the time-step used for simulations.

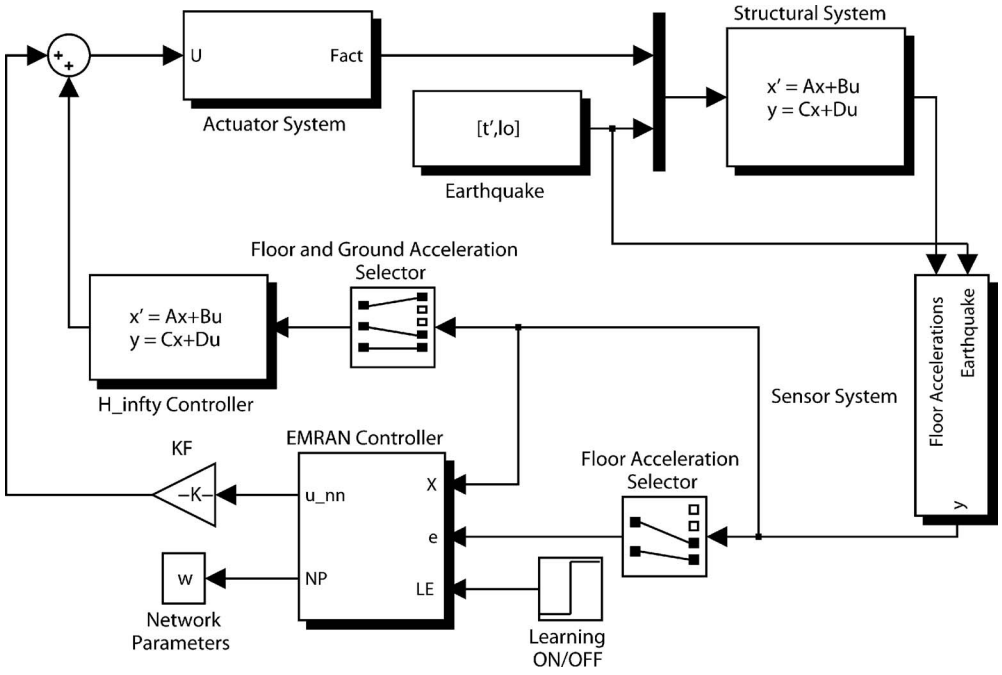

Fig. 3. Controller architecture used for implementation in SIMULINK

Table 1. GA Optimized Values of EMRAN Control Parameters

Parameter	K_F (kN)	ϵ_{max}	ϵ_{min}	e_{min}	r	κ	N_w	S_w	δ	P_0	q
Value	10.5	2	0.1	0.0005	0.996	0.921	200	5	0.002	1.2	0.0002

Simulation Study and Discussion of Results

The structure considered for the simulation study is a two story planar shear structure with a mass of 10,000 kg s on each floor ($m_{i=1,2}$=10,000 kg s) as shown in Fig. 4. The damping coefficients are $c_{i=1,2}$=5,657 N s/m and the stiffness for each floor is $k_{i=1,2}$=2,000 kN/m. The location of the devices and earthquake influence vector are represented by $\Lambda=\mathbf{I}_{2\times 2}$ and $\Gamma=[1\ 1]^T$. Two actuators, one on each floor, with a saturation value of 55,000 N each are used for control. The actuator dynamics are modeled using a first-order linear filter with a time constant of 5. The natural frequencies of the structure are 1.39 and 3.64 Hz. A total of six acceleration records are used for simulating the response of the structure. They are 1994 Northridge–Newhall (fault-normal 360 and fault-parallel 90 components), 1995 Kobe earthquake (JMA East-West and North-South components), and 1992 Erzikan (fault-normal and fault-parallel components) earthquake. Three cases are considered in the simulation study: (1) no actuator failure; (2) case of dead actuator on one of the floors; and (3) on-line failure of actuator on one of the floors. For cases (2) and (3), actuator failure on the first floor (called Actuator 1) and second floor (called Actuator 2) are considered separately and results are presented for each failure scenario. In order to evaluate the performance of the controller, five performance measures are presented in tabular form. They are the peak base shear (F_{bs}), peak floor acceleration ($\ddot{x}_{fl}$), peak interstory drift (x_{max}), root mean square displacement of the second floor (x^{2fl}_{rms}), and the root mean square acceleration ($\ddot{x}^{2fl}_{rms}$) of the second floor. The normalized performance measures for the corresponding quantities are denoted by $\bar{F}_{bs}$, $\bar{x}_{fl}$, $\bar{x}_{max}$, $\bar{x}^{2fl}_{rms}$, and $\bar{\ddot{x}}^{2fl}_{rms}$. The results presented in the tables in this section are normalized with respect to the corresponding quantities of the baseline H_∞ controller with no failure. The exception are the results of the baseline H_∞ controller that are normalized with respect to the uncontrolled responses.

Neural-Aided Controller with No Actuator Failure

The structural responses for the uncontrolled case (open loop responses) are shown in Table 2. The results for the baseline H_∞ controller with no actuator failure are presented in Table 3 along with the values (in parens) when normalized with respect to the uncontrolled values as shown in Table 2. The results of the neural-aided controller without actuator failures are presented in Table 4. The results in Tables 2–4 are shown in terms of the five

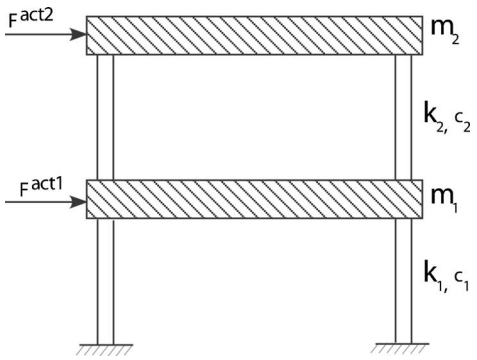

Fig. 4. Two-story model used for numerical simulations

Table 2. Uncontrolled Responses of Structure

Earthquake	Case	F_{bs} (kN)	$\ddot{x}_{fl}$ (g)	x_{max} (cm)	x_{rms}^{2fl} (cm)	$\ddot{x}_{rms}^{2fl}$ (g)
Newhall	Fault-normal	495.9	3.22	15.80	13.73	1.07
	Fault-parallel	221.6	1.49	7.29	7.21	0.57
Kobe	Fault-normal	595.0	3.88	19.02	17.36	1.36
	Fault-parallel	487.6	3.04	14.93	15.03	1.17
Erzinkan	Fault-normal	162.3	1.03	5.07	4.83	0.38
	Fault-parallel	285.0	1.75	8.56	7.57	0.59

Table 3. Results for H_∞ Controller without Actuator Failure

Earthquake	Case	F_{bs} (kN) ($\bar{F}_{bs}$)	$\ddot{x}_{fl}$ (g) ($\bar{\ddot{x}}_{fl}$)	x_{max} (cm) ($\bar{x}_{max}$)	x_{rms}^{2fl} (cm) ($\bar{x}_{rms}^{2fl}$)	$\ddot{x}_{rms}^{2fl}$ (g) ($\bar{\ddot{x}}_{rms}^{2fl}$)
Newhall	Fault-normal	306.9 (0.62)	2.11 (0.66)	8.44 (0.53)	4.01 (0.29)	0.41 (0.38)
	Fault-parallel	144.9 (0.65)	1.30 (0.88)	5.46 (0.75)	1.69 (0.23)	0.18 (0.33)
Kobe	Fault-normal	265.3 (0.45)	1.74 (0.45)	6.61 (0.35)	3.41 (0.20)	0.37 (0.27)
	Fault-parallel	250.7 (0.51)	1.53 (0.50)	5.45 (0.37)	3.03 (0.20)	0.32 (0.27)
Erzinkan	Fault-normal	109.5 (0.67)	0.73 (0.71)	2.45 (0.48)	1.82 (0.38)	0.14 (0.39)
	Fault-parallel	167.9 (0.59)	1.13 (0.65)	4.25 (0.50)	2.87 (0.38)	0.29 (0.48)

Note: The quantities shown within the parens are normalized with respect to the uncontrolled responses.

Table 4. Results for Neural-Aided H_∞ Control without Actuator Failures Normalized by Baseline H_∞ Responses without Failure

Earthquake	Case	$\bar{F}_{bs}$	$\bar{\ddot{x}}_{fl}$	$\bar{x}_{max}$	$\bar{x}_{rms}^{2fl}$	$\bar{\ddot{x}}_{rms}^{2fl}$
Newhall	Fault-normal	0.90	0.90	0.80	0.70	0.62
	Fault-parallel	0.82	0.86	0.91	0.84	0.74
Kobe	Fault-normal	0.62	0.71	0.81	0.80	0.65
	Fault-parallel	0.79	0.76	0.77	0.77	0.57
Erzinkan	Fault-normal	0.91	0.90	0.90	1.00	0.55
	Fault-parallel	0.64	0.65	0.84	0.63	0.48

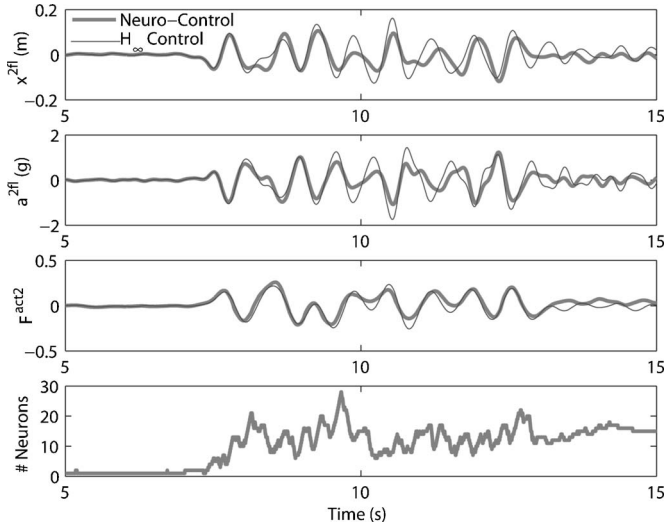

Fig. 5. Time histories for second floor displacement, second floor acceleration, second floor actuator force (normalized by weight of structure), and number of neurons for Kobe-NS component under conditions of no actuator failures

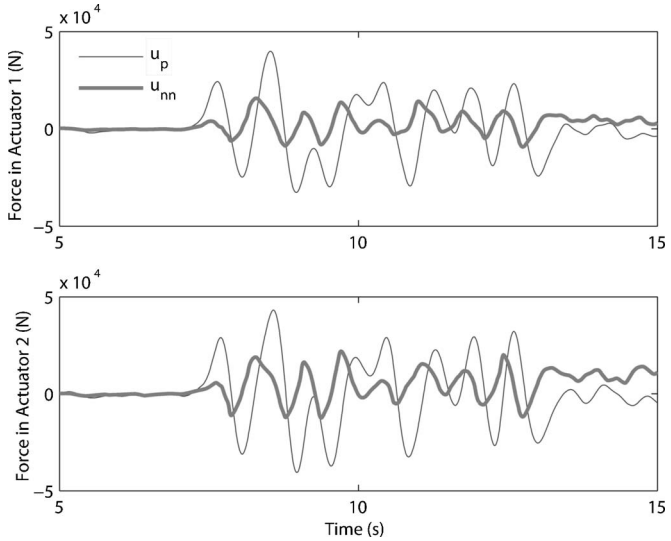

Fig. 6. Comparison of actuator force contribution from H_∞ primary controller u_p, and neural network u_{nn} in each actuator, for Kobe-NS component under conditions of no actuator failure

Table 5. Results for Neuro-H_∞ with Dead Actuator Number 2—All Quantities Shown Are Normalized by Baseline H_∞ Responses without Failure Presented in Table 3

Earthquake	Case	$\bar{F}_{bs}$	$\ddot{\bar{x}}_{fl}$	$\bar{x}_{max}$	$\bar{x}^{2fl}_{rms}$	$\ddot{\bar{x}}^{2fl}_{rms}$
Newhall	Fault-normal	1.27 (1.34)	1.16 (1.33)	1.43 (1.63)	1.40 (2.26)	1.14 (1.87)
	Fault-parallel	1.13 (1.24)	1.00 (1.06)	1.17 (1.24)	1.60 (2.44)	1.28 (1.94)
Kobe	Fault-normal	1.27 (1.96)	1.41 (2.10)	1.82 (2.71)	1.95 (3.52)	1.51 (2.78)
	Fault-parallel	1.45 (1.56)	1.43 (1.63)	1.96 (2.23)	1.73 (2.96)	1.36 (2.37)
Erzinkan	Fault-normal	1.13 (1.18)	1.16 (1.19)	1.69 (1.74)	1.21 (1.78)	1.27 (1.91)
	Fault-parallel	1.18 (1.50)	1.07 (1.35)	1.39 (1.76)	1.26 (1.91)	1.07 (1.64)

Note: Quantities inside parens show the corresponding values for the stand-alone H_∞ controller with dead actuator No. 2.

performance measures presented earlier. The performance measures shown in Table 4 are normalized with the corresponding values of the baseline H_∞ controller without failures. In Table 4, values of performance measures less than one are an improvement over the baseline H_∞ controller. As can be observed from the results presented in Table 4, the neural-aided controller results in substantial improvements over the baseline H_∞ controller. The improvements in the peak base shear range from 9 to nearly 38% in the case of north-south component of Kobe earthquake. The improvement in the magnitude of peak floor accelerations ranges from 9.5 to 34.9% in the case of the fault-parallel component of the Erzinkan earthquake. Similarly, the peak interstory drift ranges from 9.3 to 23.2% reduction in the case of the east-west component of the Kobe earthquake. The root mean square quantities of the acceleration of the second floor show substantial improvements over the baseline case for all earthquakes (range from 26 to 51.6% in the case of the fault-parallel component of the Erzinkan earthquake). Notable improvements for root mean square displacements are also observed for most of the earthquakes. The response time histories of second floor displacement (x^{2fl}), and acceleration (a^{2fl}) are shown in Fig. 5 for the case of the north-south component of the Kobe earthquake. Also included in Fig. 5 are the time histories of the control force in Actuator 2 (F^{act2}, normalized by the total weight of the structure) for both the baseline case and the neural control case, and the number of neurons generated by the algorithm. The peak and root mean square response reductions in the floor displacements and the floor accelerations are evident from the time histories. It can also be observed that the magnitude of force in Actuator 2 in both the baseline and the neural-aided case is comparable. The EMRAN tuning algorithm is efficient in learning the on-line force needed to reduce the responses beyond what is achievable using H_∞ control alone. The number of neurons in the hidden layer reaches only a peak value of 28, which is practically realizable for on-line implementation. The number of neurons in the hidden layer drops to 15 from a peak of 28 after 15 s showing evidence of the pruning action after the passage of periods of large earthquake intensity. The relative magnitudes of the two components of the total control force in each actuator, F^{act}, the H_∞ optimal control force u_p, and the control force from the neural network, u_{nn} are shown in Fig. 6. The quantity F^{act2} in Fig. 5 is obtained by the summation of the forces in the two components u_p and u_{nn} for Actuator 2 in Fig. 6, and normalizing it by the total weight of the structure. The peak control force that can be generated by the neural component is 10,500 N (K_F in Table 1), which is roughly 20% of the saturation value of 55,000 N corresponding to the peak value for the total control force in each actuator.

Case of Dead Actuator

The results for the case of the dead actuator on the second floor in terms of the performance measures are presented in Table 5. The performance measures are normalized with respect to values obtained using the baseline H_∞ controller (without failures). Such a normalization was performed because it is intuitive to observe how the neural-aided controller performs when compared to no-failure conditions with the H_∞ controller alone. The results for the case of the stand-alone H_∞ controller with a dead actuator on the second floor are shown in parens next to the corresponding neural-control results in Table 5. As expected, all the quantities in Table 5 are greater than one, which means the performance degrades due to actuator failure compared to the baseline H_∞ controller. However, with the neural-aided controller, the effect of performance degradation due to actuator failure is minimized compared to the case when the neural aid is not available. The peak base shear, the peak floor acceleration, and the maximum drift are reduced by a few percent to nearly 50% in the case of the fault-normal component of Kobe earthquake. The effect of the neural control is clearly evident for root mean square quantities, where the values are reduced by almost 80% for the case of the fault-normal component of the Kobe earthquake. One general conclusion that can be made from the results is that the performance of the neural controller is significantly better when the earthquake has multiple strong pulses and the maximum pulse

Table 6. Results for Neuro-H_∞ with Dead Actuator Number 1—All Quantities Shown Are Normalized by Baseline H_∞ Responses without Failure Presented in Table 3

Earthquake	Case	$\bar{F}_{bs}$	$\ddot{\bar{x}}_{fl}$	$\bar{x}_{max}$	$\bar{x}^{2fl}_{rms}$	$\ddot{\bar{x}}^{2fl}_{rms}$
Newhall	Fault-normal	1.12 (1.18)	1.08 (1.20)	1.00 (1.16)	0.86 (1.21)	0.78 (1.13)
	Fault-parallel	0.98 (1.13)	1.00 (1.09)	1.04 (1.10)	0.94 (1.16)	0.88 (1.10)
Kobe	Fault-normal	0.80 (1.31)	0.91 (1.23)	0.98 (1.24)	0.96 (1.53)	0.84 (1.39)
	Fault-parallel	1.12 (1.25)	1.11 (1.26)	1.05 (1.22)	0.93 (1.46)	0.82 (1.33)
Erzinkan	Fault-normal	1.06 (1.13)	1.04 (1.12)	1.12 (1.18)	0.84 (1.04)	0.72 (1.19)
	Fault-parallel	0.75 (1.25)	0.82 (1.18)	0.89 (1.19)	0.73 (1.41)	0.64 (1.36)

Note: Quantities inside parens show the corresponding values for the stand-alone H_∞ controller with dead actuator No. 1.

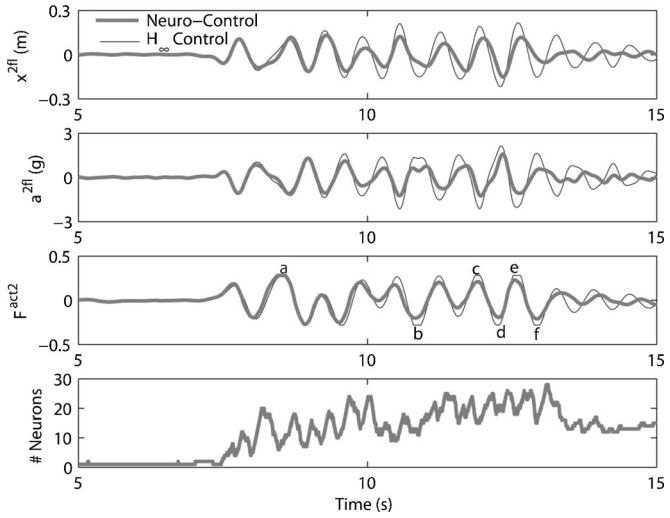

Fig. 7. Time histories for second floor displacement, second floor acceleration, second floor actuator force (normalized by weight of structure), and number of neurons for Kobe-NS component for dead actuator on first floor

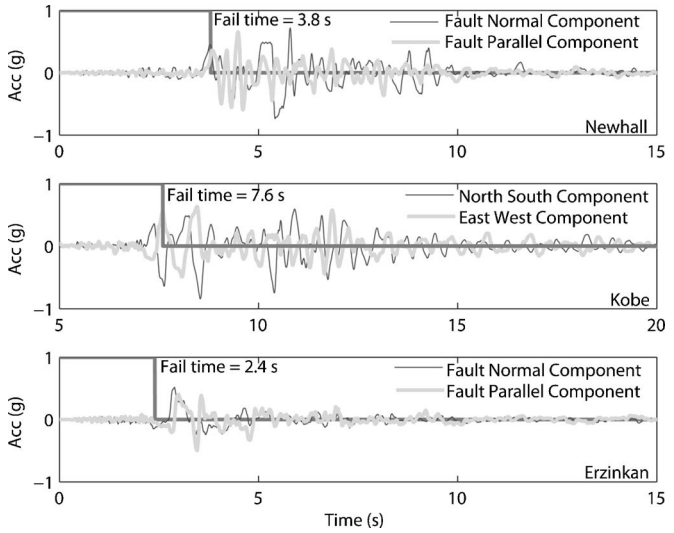

Fig. 8. Time of failure of actuators used in simulations corresponding to six earthquake records

(contributing to the peak response) follows a series of relatively smaller pulses, such as that for the Kobe earthquake. However, if the maximum pulse occurs during the early stages of the earthquake the novelty in the excitation leads to a relatively weak performance by the neural network in reducing the peak response quantities. However, the network is still capable of reducing the root mean square responses as in doing so it has sufficient time to learn the dynamics on-line. It must be emphasized that although all the quantities shown in Table 5 are greater than one, with one corresponding to the baseline H_∞ case with no failure, all the quantities shown in Table 5 are lower than the corresponding uncontrolled values presented in Table 2.

The results for the case of the dead actuator on the first floor is shown in Table 6. As in the previous case, the performance measures are normalized with respect to values obtained using the baseline H_∞ controller (without failures). It can be readily observed from the results presented in Table 6 that it is possible to achieve better control action when the actuator on the second floor is intact. The majority of the performance measures in Table 6 corresponding to the neural-aided case are close to or less than one, which is the baseline H_∞ control case without failures. In comparison, the values corresponding to the H_∞ control without neural aid are significantly greater than one showing the superior performance of the neural control. In percentage terms the case of H_∞ controller with actuator failure results in peak response quantities that are 6–66% greater than the neural control case, and 23–113% greater than the neural control case for the root mean square quantities. As with the case of the dead actuator on the second floor although the quantities shown in Table 6 are greater than one, with one corresponding to the baseline H_∞ case with no failure, all the quantities shown in Table 6 are lower than the corresponding uncontrolled values presented in Table 2.

The response time histories of second floor displacement and acceleration are shown in Fig. 7 for the case of the north-south component of the Kobe earthquake when the dead actuator is on the first floor. The reductions in the displacement and acceleration responses are clearly evident from Fig. 7. The control force in Actuator 2 for the case of the H_∞ controller and the neural-aided controller are also shown in Fig. 7. The locations of actuator force saturation are indicated by the letters a–f. As a result of the failure of one of the actuators, the H_∞ controller is rendered less effective during peak demand periods due to the actuator saturation. With the addition of the neural network, the total control force is generally bounded within the saturation limits, thus resulting in effective control action.

Online Failure of Actuators

The performance of the neural-aided controller is investigated for the case of on-line failure of actuators. The actuators failed at different times for the earthquakes simulating the situation wherein an actuator on a particular floor may be active for a period of time, but cease to function arbitrarily. In all the cases

Table 7. Results for Neural-Aided H_∞ Controller with Online Failure of Actuator Number 2—All Quantities Shown Are Normalized by Baseline H_∞ Responses without Failure Presented in Table 3

Earthquake	Case	$\bar{F}_{bs}$	$\ddot{\bar{x}}_{fl}$	$\bar{x}_{max}$	$\bar{x}^{2fl}_{rms}$	$\ddot{\bar{x}}^{2fl}_{rms}$
Newhall	Fault-normal	1.26 (1.34)	1.16 (1.33)	1.42 (1.63)	1.41 (2.28)	1.14 (1.88)
	Fault-parallel	1.11 (1.23)	1.00 (1.06)	1.17 (1.24)	1.62 (2.47)	1.28 (1.96)
Kobe	Fault-normal	1.30 (1.97)	1.43 (2.11)	1.85 (2.73)	2.03 (3.57)	1.58 (2.82)
	Fault-parallel	1.43 (1.50)	1.47 (1.56)	2.02 (2.14)	1.73 (2.90)	1.36 (2.32)
Erzinkan	Fault-normal	1.08 (1.13)	1.12 (1.17)	1.64 (1.71)	1.15 (1.71)	1.20 (1.83)
	Fault-parallel	1.15 (1.50)	1.03 (1.36)	1.34 (1.76)	1.21 (1.92)	1.02 (1.64)

Note: Quantities inside parens show the corresponding values for the stand-alone H_∞ controller with online failure of actuator No. 2.

Table 8. Results for Neuro-H_∞ with Online Failure of Actuator Number 1—All Quantities Shown Are Normalized by Baseline H_∞ Responses without Failure Presented in Table 3

Earthquake	Case	$\bar{F}_{bs}$	$\ddot{\bar{x}}_{fl}$	$\bar{x}_{max}$	$\bar{x}^{2fl}_{rms}$	$\ddot{\bar{x}}^{2fl}_{rms}$
Newhall	Fault-normal	1.11 (1.21)	1.07 (1.21)	1.00 (1.18)	0.87 (1.24)	0.78 (1.15)
	Fault-Parallel	0.96 (1.13)	0.99 (1.10)	1.04 (1.12)	0.95 (1.17)	0.88 (1.11)
Kobe	Fault-normal	0.81 (1.32)	0.90 (1.25)	1.02 (1.26)	0.99 (1.57)	0.86 (1.42)
	Fault-parallel	1.16 (1.27)	1.13 (1.30)	1.06 (1.28)	0.95 (1.47)	0.83 (1.34)
Erzinkan	Fault-normal	1.02 (1.09)	0.98 (1.05)	1.01 (1.08)	0.92 (1.04)	0.70 (1.18)
	Fault-parallel	0.79 (1.25)	0.84 (1.17)	0.96 (1.19)	0.82 (1.41)	0.65 (1.35)

Note: Quantities inside parens show the corresponding values for the stand-alone H_∞ controller with online failure of actuator No. 1.

the actuator was made to fail prior to the arrival of the peak earthquake pulse (for Erzinkan), or during a relatively large pulse (for Newhall and Kobe). The time of failure for actuators used for the on-line failure case are shown in Fig. 8. As with the dead actuator, results are presented separately for the case when Actuator 1 fails and when Actuator 2 fails and are normalized with respect to the values obtained from the baseline H_∞ controller with no failures. The results in terms of performance indices for on-line failure of Actuator 2 are presented in Table 7. The results for the on-line failure of Actuator 1 are presented in Table 8. Based on the results presented in Tables 7 and 8, the performance of the neural-aided controller is comparable to the corresponding case of the dead actuator (in Tables 5 and 6). This is primarily due to the relatively fast on-line learning capability of the neural-aided controller. The response time histories of second floor displacement and acceleration are shown in Fig. 9 for the case of the north-south component of the Kobe earthquake as a result of actuator failure on the first floor. The reductions in the displacement and acceleration responses are evident from Fig. 9. The total number of neurons in the hidden layer is 32 which is three more than for the dead actuator and case of no failures. As with the dead actuator case in Fig. 8, the actuator saturates for the H_∞ control case at locations indicated by letters a–g in Fig. 9, resulting in performance degradation. As observed in the case of the dead actuator, the effect of actuator saturation is mitigated by the introduction of the neural controller in the control architecture. As with the case of the dead actuator, although the quantities shown in Tables 7 and 8 are greater than one, with one corresponding to the baseline H_∞ case with no failure, all the quantities shown in Tables 7 and 8 are lower in magnitude compared to the corresponding uncontrolled values presented in Table 2. The benefit of neural control is evident from the results of Table 8, where the responses (shown in brackets) for the failure case without neural aid are all significantly greater than one, whereas with neural aid the magnitudes are less than or close to one, which corresponds to the baseline H_∞ controller without failure.

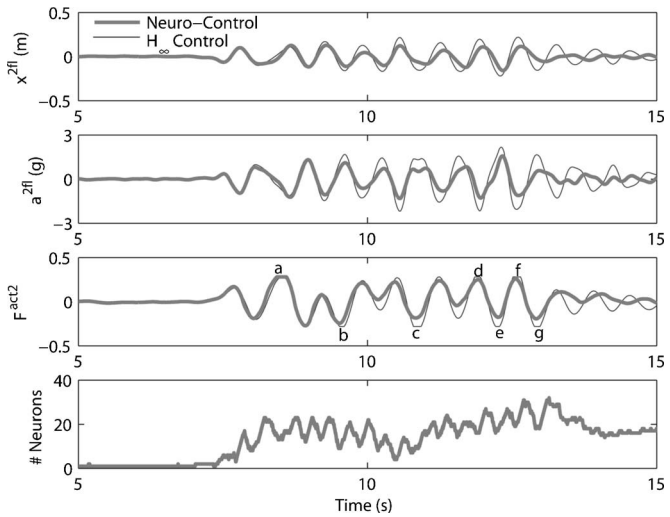

Fig. 9. Time histories for second floor displacement, second floor acceleration, second floor actuator force, and number of neurons for Kobe-NS component for on-line actuator failure on first floor

Conclusions

An on-line learning neural-aided H_∞ controller was presented and shown to be effective in reducing the response of structures subjected to earthquake excitations. In addition to providing superior performance to a conventional H_∞ controller, the neural-aided controller can handle actuator failures effectively. In all cases of actuator failure it is observed that the magnitude of the responses is less than the corresponding uncontrolled values. Results are presented for the cases of no actuator failure, and for the cases of actuator failures in terms of peak and root mean square performance measures. The advantage of the proposed control architecture is its ability to learn on-line with no a priori training. The proposed neural-aided controller performs effectively both during normal operating conditions and during conditions of actuator system failures. For all the cases simulated in this study the network created is compact, with relatively few neurons in the hidden layer during peak periods of excitation. Since the neural controller is added to an existing H_∞ controller, the architecture can be adapted for applications where a control design is already in place with minimum additional design components.

Acknowledgments

The support and funding provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through their Discovery Grants Program, and the Department of Civil and Environmental Engineering, University of Waterloo, is gratefully acknowledged by the first writer. The writers would also like to thank the reviewers for their valuable comments to improve the quality of the paper.

References

Ankireddi, S., and Yang, H. (1999). "Neural networks for sensor fault correction in structural control." *J. Struct. Eng.*, 125(9), 1056–1064.

Bani-Hani, J., and Ghaboussi, J. (1998). "Nonlinear structural control using neural networks." *J. Eng. Mech.*, 124(3), 319–327.

Broomhead, D., and Lowe, D. (1988). "Multivariable functional interpolation and adaptive networks." *Complex Syst.*, 2(3), 321–355.
Chen, H., Tsai, G., Qi, J., Yang, F., and Amini, F. (1995). "Neural network for structure control." *J. Comput. Civ. Eng.*, 9(2), 168–176.
Doyle, J., Glover, K., Khargonekar, P., and Francis, B. (1989). "State-space solutions to standard H_2 and H_∞ control problems." *IEEE Trans. Autom. Control*, 34(8), 831–847.
Ghaboussi, J., and Joghataie, A. (1995). "Active control of structures using neural networks." *J. Eng. Mech.*, 121(4), 555–567.
Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, New York.
Gomi, H., and Kawato, M. (1990). "Learning control for a closed loop system using feedback-error-learning." *Proc., 29th Conf. on Decision and Control*, Vol. 6, Honolulu, IEEE, 3289–3294.
Green, M., and Limebeer, D. (1995). *Linear robust control*, Prentice-Hall, Englewood Cliffs, N.J.
Hecht-Nielsen, R. (1987). "Counterpropagation networks." *Appl. Opt.*, 26(23), 4979–4984.
Holland, H. (1975). *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, Mich.
Houck, C., Jones, J., and Kay, M. (1995). "A genetic algorithm for function optimization: A matlab implementation." *Rep. No. NCSU-IE TR 95-09*, North Carolina State Univ., Raleigh, N.C.
Koh, B., Li, Z., Dharap, P., Nagarajaiah, S., and Phan, M. (2005). "Actuator failure detection through interaction matrix formulation." *J. Guid. Control Dyn.*, 28(5), 895–901.
Li, Y., Sundararajan, N., and Saratchandran, P. (2000). "Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems." *Proc. IEEE*, 147(4), 476–484.
Li, Y., Sundararajan, N., Saratchandran, P., and Wang, Z. (2004). "Robust neuro-H_∞ controller design for aircraft auto-landing." *IEEE Trans. Aerosp. Electron. Syst.*, 40(1), 158–167.
Liut, D. A., Matheu, E., Singh, M., and Mook, D. (1999). "Neural-network of building structures by a force-matching training scheme." *Earthquake Eng. Struct. Dyn.*, 28(12), 1601–1620.
Lu, Y., Sundararajan, N., and Saratchandran, P. (1998). "Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm." *IEEE Trans. Neural Netw.*, 9(2), 308–318.
Lu, Y. W., Sundararajan, N., and Saratchandran, P. (1997). "A sequential learning scheme for function approximation using minimal radial basis neural networks." *Neural Comput.*, 9(2), 1–18.
Madan, A. (2005). "Vibration control of building structures using self-organizing and self-learning neural networks." *J. Sound Vib.*, 287(4–5), 759–784.
Masri, S., Smyth, A., Chassiakos, A., Caughey, T., and Hunter, N. (2000). "Application of neural networks for detection of changes in nonlinear systems." *J. Eng. Mech.*, 126(7), 666–676.
MATLAB. (2000). *MATLAB software*, The Math Works, Inc., Natick, Mass.
Michalewicz, Z. (1994). *Genetic algorithms + data structures = evolution programs*, AI Series, Springer, New York.
Moody, J., and Darken, C. J. (1989). "Fast learning in network of locally-tuned processing units." *Neural Comput.*, 1(1), 281–294.
Nagarajaiah, S., and Narasimhan, S. (2006). "Smart base isolated benchmark building. Part II: Phase I sample controllers for linear isolation system." *Struct. Health Monit.*, 13(3), 589–604.
Narasimhan, S., and Nagarajaiah, S. (2006). "Smart base isolated buildings with variable friction systems: H_∞ controller and novel SAIVF device." *Earthquake Eng. Struct. Dyn.*, 35(8), 920–942.
Narasimhan, S., Nagarajaiah, S., Gavin, H., and Johnson, E. A. (2006). "Base isolated benchmark building. Part I: Problem definition." *Struct. Health Monit.*, 13(2), 573–588.
Narendra, K. S., and Parthasarathy, K. (1990). "Identification and control of dynamical systems using neural networks." *IEEE Trans. Neural Netw.*, 1(1), 4–27.
Pashlikar, A., Sundararajan, N., and Saratchandran, P. (2005). "A fault-tolerant neural aided controller for aircraft auto-landing." *Aerosp. Sci. Technol.*, 10(1), 49–61.
Sirca, J., and Adeli, H. (2004). "Counterpropagation neural network model for steel girder bridge structures." *J. Bridge Eng.*, 9(1), 55–65.
Spencer, B. F., and Nagarajaiah, S. (2003). "State of the art of structural control." *J. Struct. Eng.*, 129(7), 845–856.
Stengel, B. J. (1991). "Intelligent failure-tolerant control." *IEEE Control Syst. Mag.*, 11(4), 14–23.
Yang, J. N., Lin, S., and Jabbari, F. (2004). "H_2-based control strategies for civil engineering structures." *J. Struct. Control*, 10(3–4), 205–230.