Engineering Structures 32 (2010) 2477–2487

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct

Fault-tolerant adaptive control of nonlinear base-isolated buildings using EMRAN

Sundaram Suresh [a], Sriram Narasimhan [b,*], Satish Nagarajaiah [c], Narasimhan Sundararajan [d]

[a] School of Computer Eng., Nanyang Technological University, Singapore
[b] Department of Civil & Env. Eng., 200 University Avenue W., University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
[c] Departments of Civil & Env. Engrg. and Mech. Engrg.& Mat. Sci., Rice University, USA
[d] School of Electrical and Electronic Eng., Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 27 April 2007
Received in revised form
1 April 2010
Accepted 12 April 2010
Available online 14 May 2010

Keywords:
Adaptive control
EMRAN
Neural networks
Smart base-isolated buildings
Active control

ABSTRACT

This paper presents a direct adaptive fault-tolerant neural control scheme for the active control of nonlinear hysteretic base-isolated buildings using the recently developed Extended Minimal Resource Allocation Network (EMRAN). EMRAN is a learning algorithm in which the structure of the neural controller is adapted on-line based on the input–output data. EMRAN starts with no hidden neurons and calculates the number of hidden neurons based on growing/pruning criteria. If the criteria are not met, then the parameters of the network are adjusted using an Extended Kalman Filter (EKF). The constants associated with the growing/pruning criteria and EKF are estimated using Genetic Algorithm (GA) optimization. The advantage of the proposed control architecture is its ability to learn on-line with no *a priori* training. Most of the existing studies in structural control using neural networks require computationally intensive off-line training. Consequently, once the network parameters are learnt, the parameters remain fixed. Such procedures require an accurate mathematical model of the system. These issues are addressed in the current controller scheme by utilizing the on-line adaptation capabilities of the neural networks. The advantages of on-line adaptation are demonstrated using the controller's capability to handle actuator failures and system uncertainties. Performance of the proposed control scheme is evaluated using the recently developed nonlinear three-dimensional base-isolated benchmark structure incorporating lateral–torsional superstructure behavior and the biaxial interaction of the nonlinear bearings in the isolation layer. Results show that the proposed controller scheme can achieve the desired performance objectives under both partial actuator failure conditions and large uncertainties associated with the system's parameters.

1. Introduction

Structural control for linear structures is well understood, and many controllers based on linear optimal control theory have been proposed in the literature, e.g., [1–3]. On the other hand, control of nonlinear structures poses significant challenges. Researchers have resorted to a wide range of methods such as linearization and nonlinear control (e.g., sliding mode control). While these methods have produced several reliable and effective controllers, a growing interest in the control of uncertain and fault-tolerant nonlinear systems has sparked significant research in adaptive control theory. Structures are typically characterized by large uncertainties in estimating the system's properties, and this problem is especially exemplified in nonlinear structures such as friction or hysteretic base-isolated buildings. In addition, structural properties and operating loads change with time, which necessitates including adaptivity in the control designs. Linearized methods may not provide the desirable properties during severe earthquakes (during large excursions into the nonlinear range). As with the linear controller design methods, nonlinear control methods also require an accurate description of system's dynamics. In contrast, adaptive control techniques can operate even if the nonlinear dynamics are not explicitly known *a priori*, and can adapt to the changing dynamics of the system.

Because of their nonlinear function approximation capabilities, neural networks (NNs) have demonstrated a significant potential for the adaptive control of nonlinear systems. Neural networks typically consist of multiple layers of fixed nonlinearities (e.g., sigmoid-type) and adjustable weights that are embedded within the nonlinear functions. Using concepts of universal approximation theorem [4], multi-layer networks with sufficient number of hidden units have been shown to approximate general nonlinear functions. Such networks have been studied extensively by researchers for both system identification and control [5–13].

* Corresponding author. Tel.: +519 888 4567x38081; fax: +519 888 4349.
E-mail addresses: SSundaram@ntu.edu.sg (S. Suresh), snarasim@uwaterloo.ca, snarasim@civmail.uwaterloo.ca (S. Narasimhan), Satish.Nagarajaiah@rice.edu (S. Nagarajaiah), ensundara@ntu.edu.sg (N. Sundararajan).

0141-0296/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engstruct.2010.04.024

The main idea of an extended minimal resource allocation network (EMRAN) approach is to allocate new computational units (neurons), which are chosen to be Gaussian, depending on the input characteristics [14–16]. Based on resource allocation networks [17], each computational unit added has a local response in the region of the input space (localization). Due to their localization properties, Gaussian units explicitly store information regarding the input characteristics [18,19], as opposed to simply using them for update purposes. Since the number of neurons in the hidden layer is a function of the complexity of the problem at hand, the necessity of fixing the size of the network in advance does not exist. EMRAN starts with zero hidden neurons and has a growing, pruning and parameter updating strategy using an Extended Kalman Filter (EKF) algorithm. For making the computations faster for real-time applications, instead of updating the parameters of all the neurons, only the parameters of the nearest neighbor neuron are updated. This results in a network with a fast sequential learning ability. This type of network (EMRAN) has been used for several applications such as function approximation and nonlinear system identification [16]. Hidden neurons are both added and removed based on the input patterns and learning mechanisms to maintain a compact and high-performance network.

To date, most of the adaptive control strategies for seismically-excited base-isolated buildings proposed in the literature involve back-stepping [20–22]. The aforementioned studies have shown that adaptive control can be very effective for uncertain hysteretic systems. Back-stepping employs Lyapunov synthesis to determine nonlinear controllers in a recursive manner. Neural networks (NNs) have been used extensively for active seismic structural control applications [7–12] and there are several similarities between these studies and adaptive control theory. For example, the back-propagation algorithm [23] used for network training can also be used to adaptively tune the controller parameters on-line; however, the major difference is that the tuning in adaptive control is performed on-line, while several of these structural control studies adopt learning-then-control approach. Early studies on structural control using NNs [8,9,7] employed a two step architecture; an identifier (called an emulator) and a controller. The function of the emulator is to calculate the control error based on plant output error, which is used to train the controller subsequently. Once the controller is trained, the number of hidden neurons and weights is fixed. Such identifier-based control methods require significant training data and the performance of the controller is directly linked to the quality of the identifier. If the identifier is developed based on rough approximations of the plant model, the quality of the controller is likely to be degraded. Subsequent studies in structural control [10–12] have investigated the application of NNs for linear structures without an emulator, while retaining a learning-then-control approach. The resulting architecture still remains fixed (hidden neurons), developed using off-line training data. In addition, aspects such as system parameter uncertainty and failures have not been considered in these studies.

In the current study, by casting the control problem in an adaptive control framework, the free parameters of the network, such as the number of Gaussian units, their centers and widths, are adapted on-line. The controller structure does not involve an identifier. Instead, the controller tunes itself based on the deviations of the measured outputs from the desired values. The principal problem of the unavailability of the correct network output is inferred from the observed plant's behavior. The measurements that contain the necessary information to correct these tracking errors are developed based on adaptive systems theory [24,25]. The resulting controller adapts on-line, as opposed to adapt-then-control as found in method utilized thus far in the literature.

There are two important aspects considered in this study. The first element is the controller's robustness in the presence of system parameter uncertainties. Theoretically, since the network's parameters such as the number of neurons, centers, widths and weights are all initialized to zero, one may argue that the controller is inherently independent (or, in other words, robust) of system parameters. However, as will be shown in the subsequent discussion, there is a number of constants associated with the EKF algorithm and the growing and pruning criteria employed in the architecture. These constants are system dependent, and are calculated using optimization algorithms (which require a mathematical model of the system). The optimization is conducted using a perturbed linearized model, in order to demonstrate the robustness of the controller to system uncertainties.

Another important aspect is the possibility that some of the actuators may fail to actuate in the event of an earthquake. Other researchers have studied sensor and actuator failures in the context of fault detection and isolation (FDI) e.g., [26–28]. The process of FDI often introduces significant complexity in the control architecture, such as estimation, which for uncertain nonlinear systems is not trivial. Even assuming FDI is achieved, establishing control logic under different failure scenarios is not practical for large scale civil structural systems. In aircraft control, researchers have investigated re-configurable controllers, e.g., [29], that do not require FDI. The controllers adapt themselves based on the measured performance under failure conditions, thus avoiding the cumbersome process of FDI. Recently [30], the authors have presented a failure-tolerant H_∞ controller aided by a neural network. In the aforementioned study, a H_∞ controller was used as a baseline controller for a linear system, and the failure-tolerance of the H_∞ controller was enhanced using a neural network. Results showed that the neural-aided controller is capable of enhancing the performance and the failure tolerance capabilities of the H_∞ controller. Subsequently, experimental studies were performed at Rice University to demonstrate the effectiveness of the neural-aided controller (EMRAN architecture) to adjust to on-line sensor failures [31].

In this paper, EMRAN based control architecture is used to control a nonlinear base-isolated benchmark building [32]. Performance of the algorithm is investigated under three scenarios of actuator failures, without the use of a baseline controller. A comprehensive suite of six near-fault earthquakes (due to large relatively-long period pulses, near-fault earthquakes pose a high risk of damage for base isolated structures [32] is used to evaluate the controller's performance. Since the proposed methodology utilizes active devices, it is assumed that a power supply is available during the seismic event.

The paper is organized as follows. The mathematical formulation of a three-dimensional base-isolated benchmark structure is presented first, followed by the adaptive control scheme and the parameter update laws. The results of the simulation studies on the benchmark structure are presented next followed by the conclusions.

2. Structural and isolation model

The structure considered (Fig. 1) in this study is a nonlinear three-dimensional base-isolated building [32], that is based on a full-scale benchmark structure located in Los Angeles, California. The benchmark structure is an eight-storey, steel-braced framed building, that is L-shaped in plan (Fig. 2), and has dimensions of 82.4-m long and 54.3-m wide. The isolators are connected between the ground and a rigid concrete base slab. The superstructure is modeled as a three-dimensional linear elastic system. Floor slabs and the base are assumed to be rigid in plane. The superstructure and the base are modeled using three master degrees of freedom

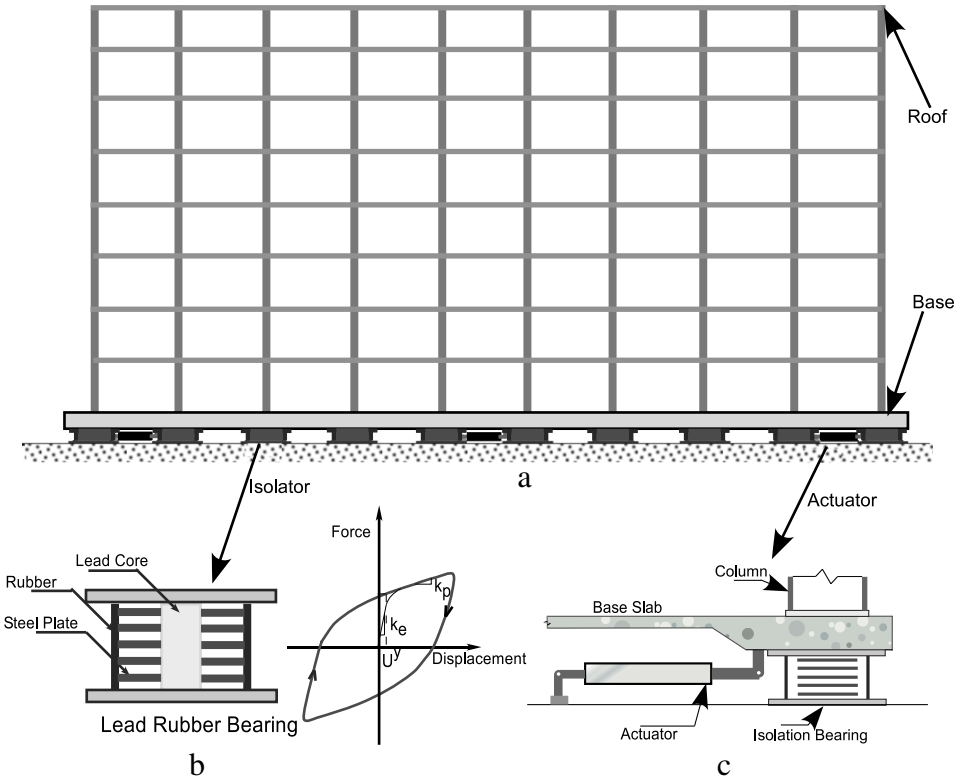

Fig. 1. (a) Elevation view of the eight-storey base-isolated building, (b) Lead–rubber-bearing isolation system, (c) Typical actual placement.

(DOF) per floor at the center of mass. The combined model of the superstructure (24 DOF) and isolation system (3 DOF) consists of 27 degrees of freedom. All twenty four modes in the fixed base case are used in modeling the superstructure. The superstructure damping ratio is assumed to be 5% in all fixed-base modes (un-isolated case).

2.1. Isolation model

A combination of 32 linear elastomeric bearings and 62 non-linear lead rubber bearings, called LRB (linear elastomeric bearing with a lead core) constitutes the isolation system considered in this paper. The plan view showing the layout of the isolation elements is shown in Fig. 2. The linear elastomeric bearings are modeled using linear spring stiffnesses and linear viscous dampers.

The bi-axial hysteretic behavior of lead–rubber bearings is modeled using the biaxial interaction equations of the hysteretic model as follows [33]:

$$U^y \begin{Bmatrix} \dot{z}_x \\ \dot{z}_y \end{Bmatrix} = \alpha \begin{Bmatrix} \dot{U}_{bx} \\ \dot{U}_{by} \end{Bmatrix} - \mathbf{Z}_w \begin{Bmatrix} \dot{U}_{bx} \\ \dot{U}_{by} \end{Bmatrix}$$
$$\mathbf{Z}_w = \begin{bmatrix} z_x^2(\gamma sgn(\dot{U}_{bx} z_x) + \beta) & z_x z_y(\gamma sgn(\dot{U}_{by} z_y) + \beta) \\ z_x z_y(\gamma sgn(\dot{U}_{bx} z_x) + \beta) & z_y^2(\gamma sgn(\dot{U}_{by} z_y) + \beta) \end{bmatrix} \quad (1)$$

where z_x and z_y are dimensionless hysteretic variables that are bounded by ± 1. α, β and γ are dimensionless quantities. U_{bx}, U_{by} and $\dot{U}_{bx}$, $\dot{U}_{by}$, represent the displacements and velocities in the x and y directions, respectively, at the isolation bearing or device and U^y is the yield displacement. The forces, f, mobilized in the elastomeric isolation bearings or control devices can be modeled by an elastic–viscoplastic model with strain hardening

$$f_x = k_p U_{bx} + c_v \dot{U}_{bx} + \left(k_e - k_p\right) U^y z_x \quad (2)$$

$$f_y = k_p U_{by} + c_v \dot{U}_{by} + \left(k_e - k_p\right) U^y z_y \quad (3)$$

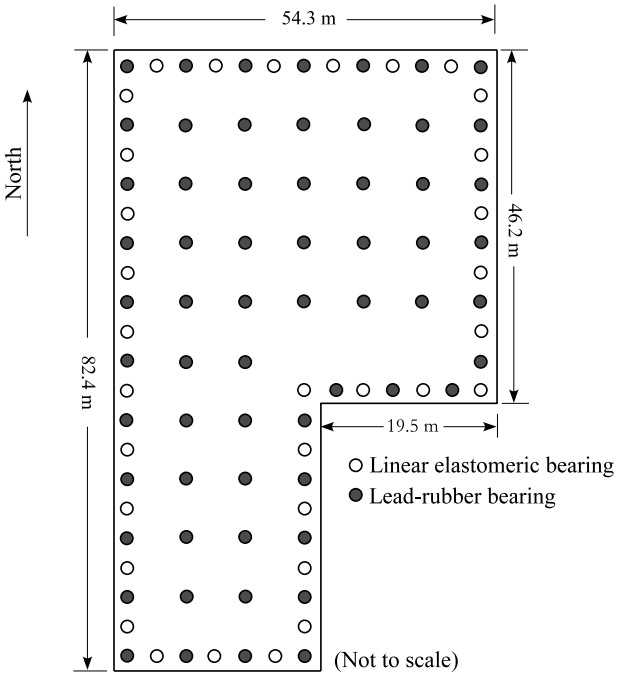

Fig. 2. Plan view of the eight-storey base-isolated building.

where k_e = pre-yield stiffness, k_p = post-yield stiffness, c_v = viscous damping coefficient of the elastomeric bearing or control device, U^y is the yield displacement. The isolation system's parameters have been specified in part III of the benchmark definition paper [34]. Note that the authors of the aforementioned study [34] have made a simplification to the LRB isolation system properties by assuming that the hysteretic behavior follows a bi-linear law. However, in this paper, the nonlinear behavior is accounted for by use of Eqs. (2) and (3).

The equations of motion for the linear superstructure are expressed as,

$$\mathbf{M}_{n\times n}\ddot{\mathbf{U}}_{n\times 1} + \mathbf{C}_{n\times n}\dot{\mathbf{U}}_{n\times 1} + \mathbf{K}_{n\times n}\mathbf{U}_{n\times 1} = -\mathbf{M}_{n\times n}\mathbf{R}_{n\times 3}(\ddot{\mathbf{U}}_g + \ddot{\mathbf{U}}_b)_{3\times 1} \quad (4)$$

in which, n is three times the number of floors (excluding the base), $\mathbf{M}$ is the superstructure's mass matrix, $\mathbf{C}$ is the superstructure's damping matrix in the fixed base case, $\mathbf{K}$ is the superstructure's stiffness matrix in the fixed base case and $\mathbf{R}$ is the matrix of earthquake influence coefficients, i.e. the matrix of displacements and rotations at the center of mass of the floors resulting from a unit translation in the X and Y directions and unit rotation at the center of mass of the base. Furthermore, $\ddot{\mathbf{U}}$, $\dot{\mathbf{U}}$ and $\mathbf{U}$ represent the floor's acceleration, velocity and displacement vectors relative to the base, $\ddot{\mathbf{U}}_b$ is the vector of base accelerations relative to the ground and $\ddot{\mathbf{U}}_g$ is the vector of ground accelerations.

The equations of motion for the base are given by:

$$\mathbf{R}^T_{3\times n}\mathbf{M}_{n\times n}\left[(\ddot{\mathbf{U}})_{n\times 1} + \mathbf{R}_{n\times 3}(\ddot{\mathbf{U}}_g + \ddot{\mathbf{U}}_b)_{3\times 1}\right]_{n\times 1} + \mathbf{M}_{b_{3\times 3}}(\ddot{\mathbf{U}}_g + \ddot{\mathbf{U}}_b)_{3\times 1} + \mathbf{C}_{b_{3\times 3}}\dot{\mathbf{U}}_{b_{3\times 1}} \quad (5)$$

$$+ \mathbf{K}_{b_{3\times 3}}\mathbf{U}_{b_{3\times 1}} + \mathbf{f}_{b_{3\times 1}}\left(\mathbf{U}_b, \dot{\mathbf{U}}_b, \theta\right) + \mathbf{f}_{c_{3\times 1}} = 0 \quad (6)$$

where, θ contains the isolation system parameters, $\mathbf{f}_b$ consists of f_x, f_y and f_r, which are the forces in the nonlinear isolation system at the center of mass of the base in the x, y, and rotational directions, respectively. f_r is calculated by transforming the spatially distributed f_x and f_y to the center of mass of the base. $\mathbf{M}_b$ is the diagonal mass matrix of the rigid base, $\mathbf{C}_b$ is the resultant damping matrix of viscous isolation elements, $\mathbf{K}_b$ is the resultant stiffness matrix of elastic isolation elements, $\mathbf{f}_b$ is the vector containing the nonlinear bearing forces and $\mathbf{f}_c$ is the vector containing the control forces. Using, $\mathbf{X} = \{\mathbf{U}^T \quad \mathbf{U}_b^T \quad \dot{\mathbf{U}}^T \quad \dot{\mathbf{U}}_b^T\}^T$, the system's dynamics can be formulated as

$$\dot{\mathbf{X}}(t) = \mathcal{F}\left(\mathbf{X}, \theta, t\right) + \mathbf{B}\mathbf{F}_c(t) + \mathbf{E}\ddot{\mathbf{U}}_g(t) \quad (7)$$

where, $\mathcal{F}$ is a nonlinear vector function. The nonlinear equations for the bearings are solved using the unconditionally stable semi-implicit Runge–Kutta method. Readers are referred to the benchmark definition paper [32] for a detailed discussion on the solution procedures.

3. Adaptive control law formulation

Let us consider the nonlinear base-isolated structure in Eq. (7) represented by a general nonlinear system in discrete form:

$$\mathbf{X}(k+1) = \mathcal{F}[\mathbf{X}(k), \boldsymbol{\theta}] + \mathbf{B}\mathbf{F}_c(k) + \mathbf{E}\ddot{\mathbf{U}}_g(k),\ \mathcal{F}[\mathbf{0}, \boldsymbol{\theta}] = \mathbf{0}$$
$$\mathbf{y}(k) = \mathcal{G}[\mathbf{X}(k), \boldsymbol{\theta}] + \mathbf{B}\mathbf{F}_c(k),\ \mathcal{G}[\mathbf{0}, \boldsymbol{\theta}] = \mathbf{0} \quad (8)$$

where, $\mathcal{F}[\cdot]$ and $\mathcal{G}[\cdot]$ are smooth, nonlinear and continuous functions. Let $\mathbf{X} \in \Re^n$ represent the n-dimensional state vector, and $\ddot{\mathbf{U}}_g \in \Re^q$ be the q-dimensional earthquake disturbance, $\mathbf{F}_c \in \Re^p$ is the p-dimensional control input, and $\mathbf{y} \in \Re^m$ is the m-dimensional output. The functions, $\mathcal{F} : \Re^n \times \Re^p \times \Re^q \longrightarrow \Re^n$ and $\mathcal{G} : \Re^n \times \Re^p \longrightarrow \Re^n$ are the one-step ahead transition function and output function respectively.

The objective of the problem is to determine the control input, $\mathbf{F}_c^*(k)$ such that the plant's output, $\mathbf{y}$ follows the desired response, $\mathbf{y}_d$ accurately, i.e.,

$$\mathbf{y}_e = \|\mathbf{y} - \mathbf{y}_d\| \leq \epsilon \quad (9)$$

where, ϵ is a small positive constant.

Under the assumption that the linearized system represented by Eq. (8) is observable, it follows from the implicit function theorem [24,35] that there exists a desired control force $\mathbf{F}_c^*$ of the form:

$$\mathbf{F}_c^*(k) = \bar{\mathcal{G}}\left(\mathbf{F}_c(k-1), \ldots, \mathbf{F}_c(k-n), \mathbf{y}(k-1), \ldots, \mathbf{y}(k-n), \ddot{\mathbf{U}}_g(k), \ldots, \ddot{\mathbf{U}}_g(k-n), \mathbf{y}_d(k+1)\right) \quad (10)$$

where, $\bar{\mathcal{G}}$ is a smooth nonlinear map. The above form of a control law is known to exist and is unique [24]. Eq. (10) can be simplified further if we assume that the system's output follows the desired trajectory [36], and the desired responses are assumed to be zero. The simplified control form then becomes:

$$\mathbf{F}_c^*(k) = \bar{\mathcal{G}}\left(\mathbf{y}(k-1), \ldots, \mathbf{y}(k-n), \ddot{\mathbf{U}}_g(k), \ldots, \ddot{\mathbf{U}}_g(k-n)\right) \quad (11)$$

which can be written as,

$$\mathbf{F}_c^*(k) = \bar{\mathcal{G}}\left(\mathbf{v}\right) \quad (12)$$

where, $\mathbf{v}$ consists of past measurements and present and past values of the ground accelerations. Eq. (12) means that if the mapping $\bar{\mathcal{G}}$ is known, then the desired control force $\mathbf{F}_c^*(k)$ can be calculated using n past values of the response measurements, and $n+1$ current and past values of the earthquake disturbance. Since the function map $\bar{\mathcal{G}}$ is unknown, estimating the control force $\mathbf{F}_c^*(k)$ for a given desired response $\mathbf{y}_d$ is not possible. However, since the relationship given by Eq. (12) exists, $\bar{\mathcal{G}}$ can be approximated using a neural network with a finite number of hidden neurons [24]. The update laws for the adaptive parameters in the neural network can then be derived using the deviation of the structural response from the desired response. Hence, the nonlinear function $\bar{\mathcal{G}}$ can only be approximated.

The adaptive control force, after dropping the superscript * (the superscript is dropped in all subsequent discussions for simplicity) is then given by,

$$\mathbf{F}_c = \mathbf{u}_{ad} + \boldsymbol{\epsilon}_1 \quad (13)$$

where, $\mathbf{u}_{ad}$ is the estimated adaptive control force, and $\boldsymbol{\epsilon}_1$ is the approximation error. Using universal approximation theorem [4], the approximation error ($\boldsymbol{\epsilon}_1$) can be minimized arbitrarily by proper selection of the number of neurons (h^*). Hence, we can say that the approximation error is bounded by a constant $\boldsymbol{\epsilon}_{h^*}$, where,

$$\boldsymbol{\epsilon}_{h^*} = \sup_{t\in\Re^+} \|\boldsymbol{\epsilon}_1\|. \quad (14)$$

After ensuring an adequate number of kernels in the algorithm (h^*), the adaptive control problem becomes a problem of estimating the free parameters. It must be recognized that the number of neurons affects the performance of the controller. A method that has been used by several researchers in the past e.g., [24], is to fix the number of neurons and its adaptive parameters *a priori*. However, this is a cumbersome process that requires a detailed mathematical model. Therefore, here the on-line learning capabilities of EMRAN are used for on-line estimation of the number of hidden neurons and its adaptive parameters. EMRAN starts with no hidden neurons and adds/prunes the hidden neurons using the responses of the structure. Due to the online learning ability, the proposed EMRAN controller can handle a large number of uncertainties in the structural system parameters and can also perform effectively under system failures (such as the actuator failures investigated in this study). Details of the proposed controller architecture and parameter update laws are discussed next.

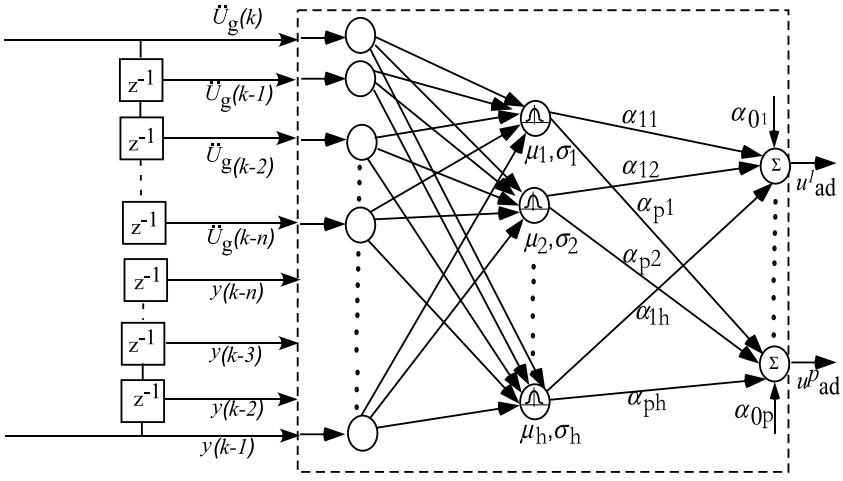

Fig. 3. Neural network architecture.

3.1. Adaptive update laws using gaussian functions and Extended Kalman Filter algorithm (EKF)

The adaptive controller developed in this paper is based on EMRAN [15,16]. EMRAN architecture consists of an input layer, a hidden layer and an output layer as shown in Fig. 3. The inter-connection weights only exist between the hidden and the output layer. Gaussian units are used in the hidden layer as activation functions because of their localization properties (each neuron responds only to a specified region of the input space). The main advantage of EMRAN is the explicit storage of information from the input data, instead of merely using the information to update parameters. For the case of the Gaussian network, the adaptive control force can be written as a linear combination of Gaussian activation functions, with centers $\boldsymbol{\mu}$ and width σ. The output of each Gaussian activation function in the hidden layer is given by:

$$z^j = \exp^{-\frac{\|\mathbf{v}-\boldsymbol{\mu}_j\|^2}{2(\sigma_j)^2}} \tag{15}$$

where $\mathbf{v}$ are the s-dimensional inputs to the Gaussian activation functions, μ is the center of the Gaussian, and σ is the width. Each output of the network is then given by,

$$u^i_{ad} = \sum_{j=1}^{h} \alpha_{ij} z^j + \alpha_{oi} \tag{16}$$

where α_{ij} and α_{oi} are inter-connection weights and biases at the output layer, respectively.

The sequential learning network starts out with no hidden units. A new hidden unit is added when the following criteria are met (at any kth time instant):

$$\|\mathbf{v}[k] - \boldsymbol{\mu}_w[k]\| > \epsilon_1[k] \tag{17}$$

$$\|\mathbf{y}_e[k]\|^2 \geq \epsilon_2 \tag{18}$$

$$J_{rmse} = \sqrt{\frac{\sum_{k=n-S_w+1}^{n} \|\mathbf{y}_e[k]\|^2}{S_w}} \geq \epsilon_3 \tag{19}$$

where $\epsilon_1[k] = \max\left[\epsilon_{max}\gamma^{k-1}, \epsilon_{min}\right]$, γ is a decay constant between 0 and 1 representing the scale of resolution [17], $\boldsymbol{\mu}_w$ is the center of the hidden unit closest ('winner' neuron) to $\mathbf{v}[k]$, $\epsilon_{1,2,3}$ are the appropriate thresholds, and S_w is the length of the sliding window. The first criterion compares the distance between the new observation and all the existing nodes, and the second criterion determines if the existing nodes are insufficient to produce a reasonable network output. The third criterion checks whether the network meets the required sum-squared-error specification for the past S_w network outputs, and is used to control the effect of noise from over-fitting the hidden neurons. The algorithm begins with ϵ_{max}, the largest scale of interest in the input space, and decays until it reaches ϵ_{min}. The parameters for the added unit are given by,

$$\boldsymbol{\alpha}_{j+1}[k] = \mathbf{y}_e[k-1],\ \boldsymbol{\mu}_{j+1}[k] = \mathbf{v}[k],$$
$$\sigma_{j+1}[k] = \kappa\|\mathbf{v}[k] - \boldsymbol{\mu}_w[k]\| \tag{20}$$

where, κ is a overlap factor that determines the overlap of the responses of the hidden units in the input space. When the criteria for adding a new unit to the hidden layer are not met, the parameters of the network, namely the connection weights, biases (assumed zero), centers and widths of the network are updated for the kth step based on the values from previous step using the Extended Kalman Filter (EKF) algorithm. In order to achieve faster performance for on-line applications, instead of updating all of the network parameters using EKF, only the parameters of the neuron whose center is closest (in a norm sense) to the network input data $\mathbf{v}$ are updated. The closest neuron is called the 'winner' neuron whose parameters ($\mathbf{NP}$) are the connection weights, centers and widths. The updated parameters of the winner neuron at the kth instant are as follows:

$$\mathbf{NP}^w[k] = \mathbf{NP}^w[k-1] + \mathbf{K}^w[k]\|\mathbf{y}_e[k-1]\| \tag{21}$$

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

$$\mathbf{K}^w[k] = \mathbf{P}^w[k-1]\mathbf{B}^w[k]\left(\mathbf{R}[k] + \mathbf{B}^w[k]^T\mathbf{P}^w[k-1]\mathbf{B}^w[k]\right)^{-1} \tag{22}$$

where, $\mathbf{B}^w[k] = \nabla_{\mathbf{w}}\mathbf{u}_{ad}$ is the gradient matrix of $\mathbf{u}_{ad}$ with respect to the parameter vector $\mathbf{w}[k]$ evaluated at $\mathbf{w}[k-1]$, and $\mathbf{R}[k]$ is the variance of the measurement noise. $\mathbf{P}^w[k]$ is the error of the covariance matrix, which is updated using:

$$\mathbf{P}^w[k] = (\mathbf{I} - \mathbf{K}^w[k]\mathbf{B}^w[k]^T)\mathbf{P}^w[k-1] + q\mathbf{I} \tag{23}$$

where q is a 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

$$\mathbf{P}^w[k] = \begin{bmatrix} \mathbf{P}^w[k-1] & \mathbf{0} \\ \mathbf{0} & P_0\mathbf{I} \end{bmatrix} \tag{24}$$

where P_0 is a 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. In this approach the outputs from each hidden unit are first normalized with respect to the maximum value of the outputs of all the hidden units. Then the units whose normalized output falls below δ for a period of N_w are removed from the network. This results in a network that is computationally efficient and adapted to a fast on-line implementation [15].

3.2. Implementation of adaptive neural controller

All the simulations for this study have been conducted in MATLAB [37]. The input vector **v** consists of base displacement, velocity and ground accelerations (vector of size $3 + 3 + 2 = 8$). For the nominal case, the base acceleration response at the center of mass of the base is used as the learning signal for the neural controller. The velocity responses are derived from the acceleration responses using a filter that mimics the operation of integration [38]. Although the weights, number of Gaussian units and associated parameters are adapted on-line without off-line training, there is a number of constants associated with the EKF algorithm and growing and pruning criteria. These constants are problem-dependent and have to be determined off-line. Given the dimensionality of the problem, trial-and-error approach is not practical. Hence, numerical optimization methods, such as genetic algorithms (GAs) are utilized.

GA based optimization is particularly powerful because the underlying cost function that minimizes the structural responses is not a smooth function of the network parameters. Due to the large number of variables it is computationally expensive to use gradient-based search techniques. The genetic algorithm (GA) is perhaps the most well-known of all evolution-based search techniques. GAs were developed by Holland [39] in an attempt to explain the adaptive processes of natural systems and to design artificial systems based upon these natural systems. The 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 [39–41]. Since a genetic algorithm maximizes the fitness function, the negative of the root mean square error (RMSE) of structure accelerations is used as a fitness. 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 10; and population size of 10 (chosen using trial-and-error). The details of the GA algorithm are outside the scope of the discussion in this paper and the readers are referred to [42] for a detailed description. A general purpose GA optimization toolbox [42] is used to perform the simulations for the current study.

A mathematical system model is necessary to perform the optimization. In this study, it is assumed that only a linearized model (linear isolation system) of the base-isolated building is available. Additionally, the mass, stiffness and damping parameters are perturbed by 25%. The mass and damping were reduced by 25%, whereas, the stiffness was increased by 25%. The linearized values of the isolation system are assumed to be the post-yield stiffness of the isolation system as described in the part II of the benchmark programs [38]. TurkBolu earthquake is used to provide the excitation input to the structure. A flowchart illustrating various steps in the EMRAN algorithm is shown in Fig. 4. For GA optimization, a perturbed model excited by TurkBolu earthquake is used, and the results of the optimization are presented in Table 1.

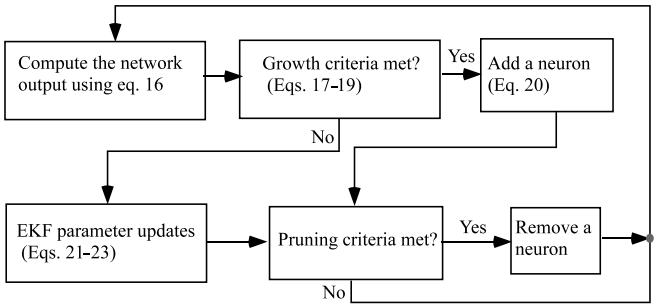

Fig. 4. Flowchart of the EMRAN algorithm.

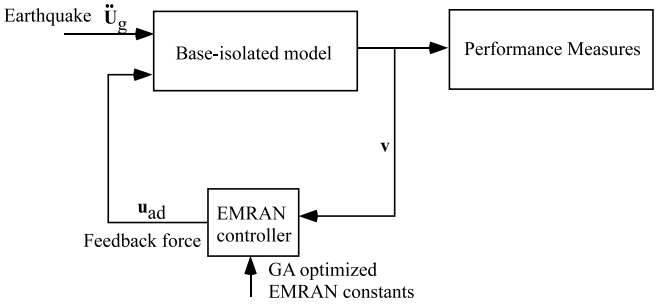

Fig. 5. Schematic diagram of the controller's implementation.

A schematic diagram of the EMRAN controller implementation is shown in Fig. 5. For implementation, the un-perturbed base-isolated model [32] is utilized. 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. The controller prunes the number of neurons after the passage of the peak intensity periods of the earthquake.

An important aspect that needs to be considered for an effective on-line control application is the computational burden due to EMRAN. To this end, recent experiments [31] have demonstrated the feasibility of using EMRAN for on-line control applications. The primary advantage in the proposed 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 is the number of parameters for the nearest neuron. Which means that the total computational burden at each step is 8^3 floating point operations (FLOPS) (1 FLOP $= 1e^{-06}$ s), which is relatively small. Control output calculation (forward pass) is $O(h)$, where h is the number of hidden neurons.

3.3. Actuator failures

One centralized on-line controller generates specified forces both in the x and y directions (distributed equally to all actuators) for the nominal case. To counteract actuator failures, eight de-centralized controllers are implemented on-line, corresponding to each actuator pair's location. Assuming that a healthy actuator can be modeled using a first order filter, failures are simulated by modifying the time constant (changing to a very small value) in the transfer function. For example, the actuator model transfer function of the form $\frac{K}{s+t_c}$ is used in this study, where t_c is the time constant and K is the forward gain of the actuator. For the nominal case, $t_c = 5$ is consistent with large actuators likely to be used in civil applications ($t_c = 20$ is commonly used for aircraft control actuators); whereas, for failures, $t_c = 0.05$. Physically, low time constants simulate a loss of hydraulic fluid to the actuators.

Table 1
GA optimized values of EMRAN constants.

Parameter	ϵ_{max}	ϵ_{min}	γ	ϵ_2	ϵ_3	δ	N_w	S_w	κ	P_0	q
Value	0.9404	0.917	0.967	0.432	0.448	0.067	500	250	0.874	1.0	0.05

Table 2
Performance indices.

Peak base shear[a]	Peak structure shear[a]	Peak base displacement[a]
$J_1 = \frac{\max_t \\|V_0(t)\\|}{\max_t \\|\hat{V}_0(t)\\|}$	$J_2 = \frac{\max_t \\|V_1(t)\\|}{\max_t \\|\hat{V}_1(t)\\|}$	$J_3 = \frac{\max_t \\|x_b(t)\\|}{\max_t \\|\hat{x}_b(t)\\|}$
Peak interstorey drift[a]	Peak floor acceleration[a]	Peak control force
$J_4 = \frac{\max_{t,f} \\|d_f(t)\\|}{\max_{t,f} \\|\hat{d}_f(t)\\|}$	$J_5 = \frac{\max_{t,f} \\|a_f(t)\\|}{\max_{t,f} \\|\hat{a}_f(t)\\|}$	$J_6 = \frac{\max_t \\|f_d(t)\\|}{\max_t \\|V_0(t)\\|}$
RMS base displacement[a]	RMS floor acceleration[a]	
$J_7 = \frac{\max_i \sigma_d(t)}{\max_i \sigma_{\hat{d}}(t)}$	$J_8 = \frac{\max_f \sigma_a(t)}{\max_f \sigma_{\hat{a}}(t)}$	–

[a] The denominator consists of the corresponding response quantity in the uncontrolled case; f = floor number, 1, ..., 5; t = time, $0 \le t \le T$; $\langle\cdot\rangle$ = inner product; $\\|\cdot\\|$ = vector magnitude; i = isolator number; V_o, V_1 = base and structural shears; x_b = base displacement; d_f = inter-storey drift; a_f = floor acceleration; f_d = total force in the control devices; σ_d and σ_a = RMS base displacement and floor acceleration; W = weight of the structure, 202,000 kN; $\hat{}$ = corresponding response quantity in the uncontrolled case.

4. Results of the simulation study and discussion

In order to evaluate the performance of the controller, a set of indices [32] shown in Table 2 are used. These indices are used to assess the performance of the controller only. The indices J_1 through J_5 measure the peak values of base shear, structural shear, base displacement, inter-storey drift and floor accelerations, respectively. These values are normalized by their respective uncontrolled values, which refer to the case when the control device is disconnected from the structure. The performance index J_6 measures the maximum control force (normalized with respect to the peak base shear in the controlled structure) developed in the device, or in other words, measures the peak control demand. The objective of the control design is to ensure that the values of all performance indices are as small as possible. However, in some cases, larger values of J_6 may also represent low peak base shears; in which case the results must be interpreted accordingly. The indices, J_7 and J_8 measure the RMS values of base displacement and acceleration normalized by their uncontrolled values. Note that the indices are used for evaluation purposes only, and are not used in the tuning or on-line controller implementation.

4.1. Results of the adaptive neural controller: nominal case

The results of the controller in terms of the performance indices are presented in Table 3 for the nominal case. The nominal case refers to no actuator failures. All the performance indices (except J_6, which is zero for uncontrolled case) are less than 1, indicating that the controlled performance is better than the uncontrolled performance. The peak base shears (J_1) are reduced between 26% and 44% compared to the uncontrolled case for all cases of earthquakes considered; the lower value corresponds to the Rinaldi and the higher value corresponds to the Erzinkan earthquake. Similar magnitudes of reductions are achieved using the adaptive controller in the case of peak structural shears (J_2).

The critical parameters for the control of the base-isolated buildings are the base displacements (J_3), inter-storey drifts (J_4) and floor accelerations (J_5). The base displacements govern design of the isolators; whereas, the inter-storey drifts and superstructure accelerations govern the structural member designs and design of components. The peak base displacements (J_3) are reduced in the range of 32%–48% compared to the uncontrolled case for all cases considered. The inter-storey drifts (J_4) are reduced in the range of 6%–40% (the higher reductions are achieved for Erzinkan earthquake), while the peak floor accelerations (J_5) are reduced by approximately 5%–34% for all earthquakes (as with inter-storey drifts, the higher reductions achieved for Erzinkan earthquake). The RMS displacement and acceleration (J_7 and J_8) are also reduced considerably for the adaptive controller case compared to the uncontrolled values.

Time-history results for the Kobe earthquakes are presented in Fig. 6. Time histories of the base acceleration, roof acceleration, base displacement and the control force are presented along with the force–displacement loops for the total force at the isolation level (transformed to the center of mass of the base). The time histories of the variation of the number of neurons are presented in Fig. 7 for Newhall, Sylmar, Rinaldi and Kobe earthquakes. The growing and pruning of the number of neurons is evident; growing occurs during peak periods of ground motion, and the number of neurons is subsequently pruned after the passage of the primary earthquake pulses. Without the pruning action, the number of neurons would reach the maximum (100, for this study), and remain at that level, leading to increased computational burden.

4.2. Results of the adaptive neural controller: failure cases

Three actuator failure cases (FC1, FC2 and FC3) as shown in Fig. 8 are investigated in this study. In all three cases, it is assumed that one half of the total actuation capacity is lost at the beginning of the earthquake. In other words, eight of the sixteen actuators fail. In the first case (FC1), it is assumed that all the interior actuators fail. In the second case (FC2), the actuators located in the corners of the building fail. In the last failure case (FC3), the failed actuators are located randomly, including failure of any one actuator in a given pair. Note that for FC3, there is one location with actuator failures in both x and y directions, denoted by the 'X'; the rest of the locations have only one actuator fail either in x or in y direction and indicated by thin rectangles. A major difference must be noted regarding the implementation of the controller for the nominal and failure cases. Unlike the nominal case, where a single controller is used, the failure cases consist of eight decentralized (on-line) controllers, one at each actuator location. No attempt is made to locate and isolate the actuator faults. Since, any attempt to locate and isolate

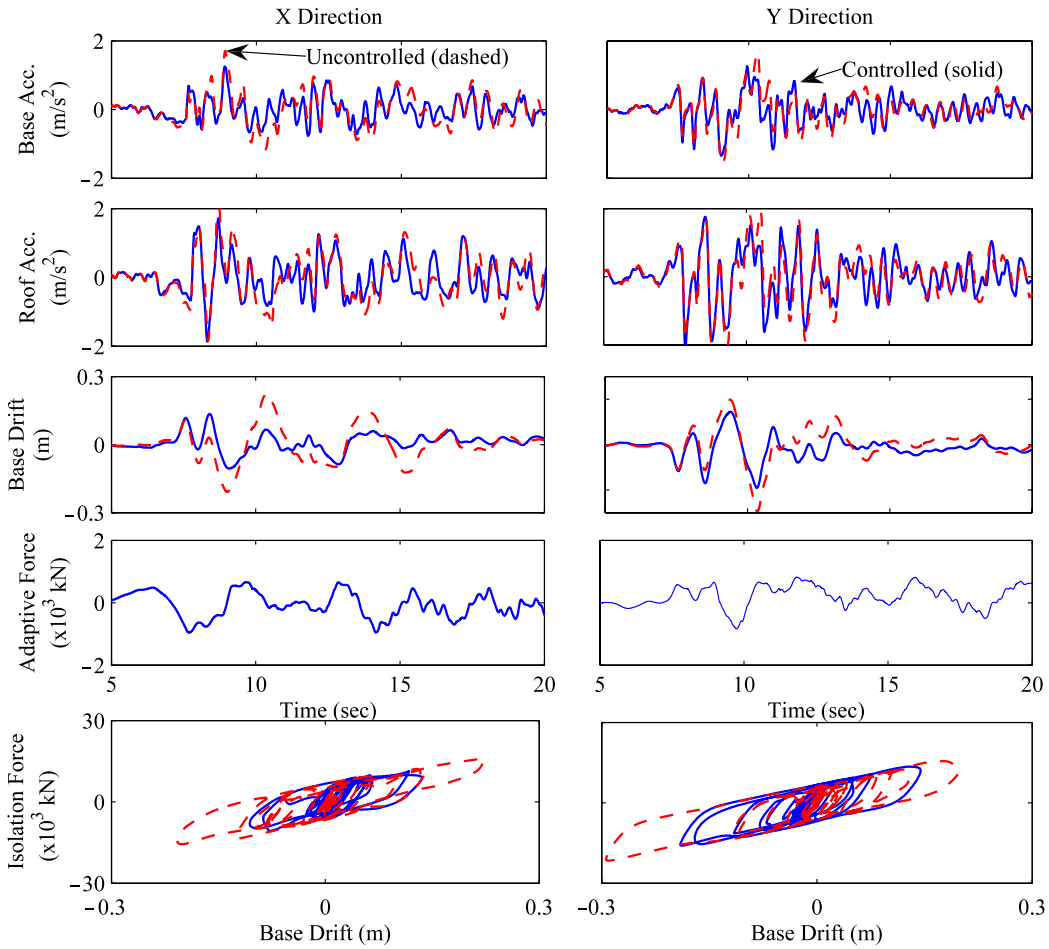

Fig. 6. Results of neural controller for Kobe earthquake — Response time histories of base and roof accelerations, adaptive force commanded by the actuator at the CM of the base, and total isolation force at the center of mass of base.

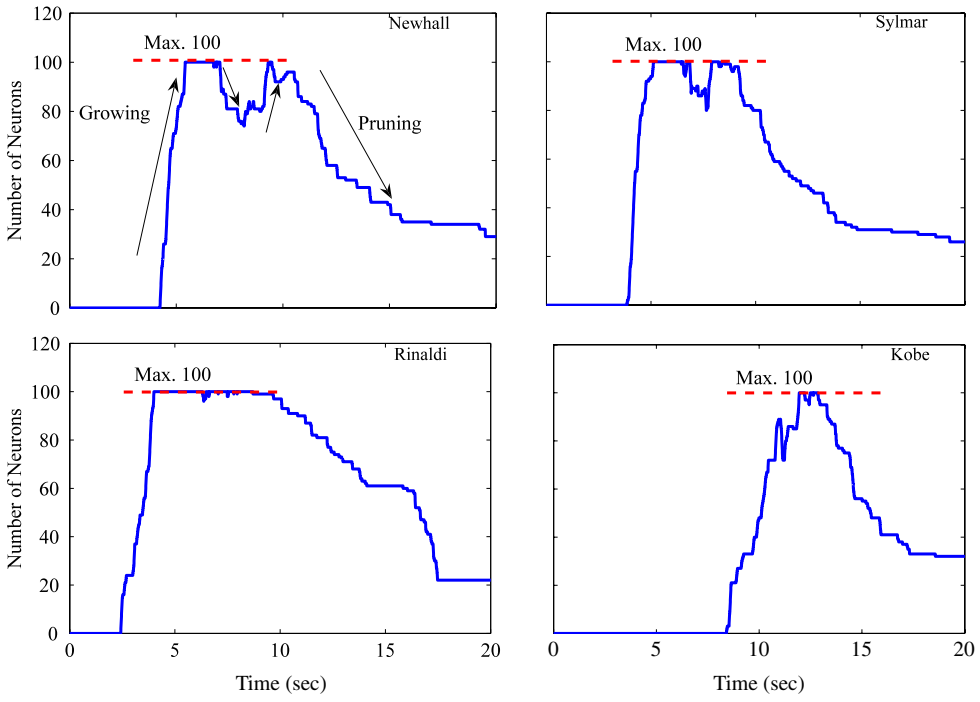

Fig. 7. Time histories of the number of neurons for Newhall, Sylmar, Rinaldi and Kobe earthquakes.

Table 3
Performance indices for adaptive controller: nominal case.

Performance index	Description[a]	Newhall	Sylmar	El Centro	Rinaldi	Kobe	Erzinkan
J1	Peak base shear	0.667	0.684	0.730	0.740	0.643	0.558
J2	Peak structural shear	0.625	0.666	0.790	0.731	0.732	0.549
J3	Peak base drift	0.650	0.598	0.524	0.684	0.649	0.537
J4	Peak storey drift	0.604	0.621	0.937	0.771	0.770	0.594
J5	Peak floor acceleration	0.733	0.738	0.953	0.809	0.891	0.662
J6	Peak control force	0.497	0.476	0.657	0.489	0.398	0.739
J7	RMS base drift	0.695	0.651	0.643	0.765	0.512	0.366
J8	RMS floor acceleration	0.818	0.595	0.968	0.760	0.912	0.467

[a] Uncontrolled values are equal to 1 for all performance indices, except for J_6.

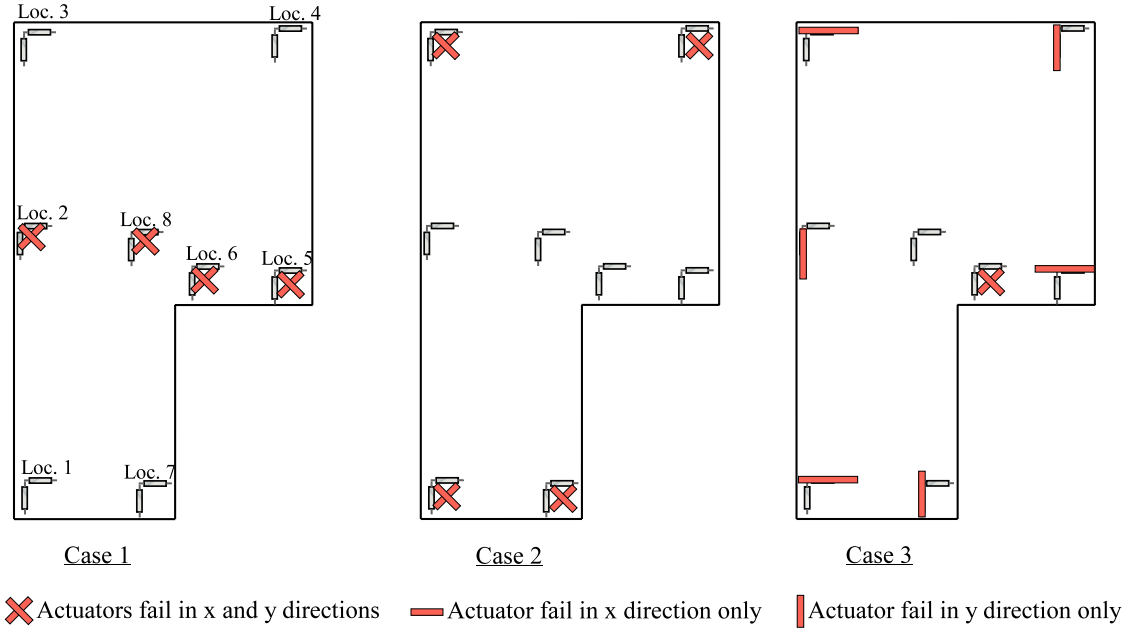

Fig. 8. Actuator failure cases considered in the simulation study.

faults would involve estimation and elaborate fault logic, which, for a nonlinear system with uncertainties is cumbersome. For the de-centralized case, the input measurements (same as that for the nominal case) are obtained locally at all the actuator locations.

The results of the controller for the failure cases in terms of the performance indices previously described are listed in Table 4. From the results, it is readily observed that all the performance indices are less than 1, which corresponds to the uncontrolled case. Hence, the controllers are capable of achieving the performance objectives even under actuator failures. Generally, it can be seen that the results (Table 4) from all three failure cases are comparable. The peak shears (base and structural, J_1 and J_2) are reduced between 12% and 30% for the earthquakes considered. The peak base displacement (J_3) is reduced between 12% and 45%. Improvements in the inter-storey drifts and the peak floor accelerations (J_4 and J_5) are also achieved (between 3% and 21%) under failure conditions. The root mean squared displacements and accelerations (J_7 and J_8) are also reduced considerably beyond their uncontrolled values. The time history comparisons between FC2 and no-failures for the base acceleration, roof acceleration, base displacement and the force displacement loops corresponding to the total isolation force at the center of mass of the base are shown in Fig. 9.

Time histories of the adaptive control force for the Kobe earthquake under the three failure scenarios is shown in Fig. 10. For all the failure cases, the actuator force (in the healthy actuator) in the vicinity of one failure location (only one representative location shown) is shown. For example, for FC1 (see Fig. 8), the actuator force in the y direction at location 3 in the vicinity of the failure location is shown in Fig. 10(a). The actuator forces for the nominal case (decentralized controllers with no failures), are also shown in the same figure. The reconfiguration capability of the controller is evident from the variations (both magnitude and time-wise variations) between the nominal and failure control forces. In all the failure cases, an actuator saturation value of 3000 kN was used; however, with the exception of the Erzinkan earthquake, the peak control force in all cases did not reach the saturation value.

5. Conclusions

A direct adaptive control scheme based on EMRAN was presented and shown to be effective in reducing the response of a nonlinear base-isolated structure subjected to earthquake excitations. The performance of the adaptive controller is verified on a fully coupled three-dimensional nonlinear benchmark base-isolated structure. The results are presented in terms of a comprehensive set of performance indices. In addition to providing good response reductions under system parameter uncertainties, it is shown that the neural controller can reconfigure itself under actuator failures and provide good control performance. The advantage of the proposed control architecture is its ability to learn on-line with no *a priori* training. Although there are several controller parameters that are system dependent, an approximate system representation is sufficient to calculate these parameters.

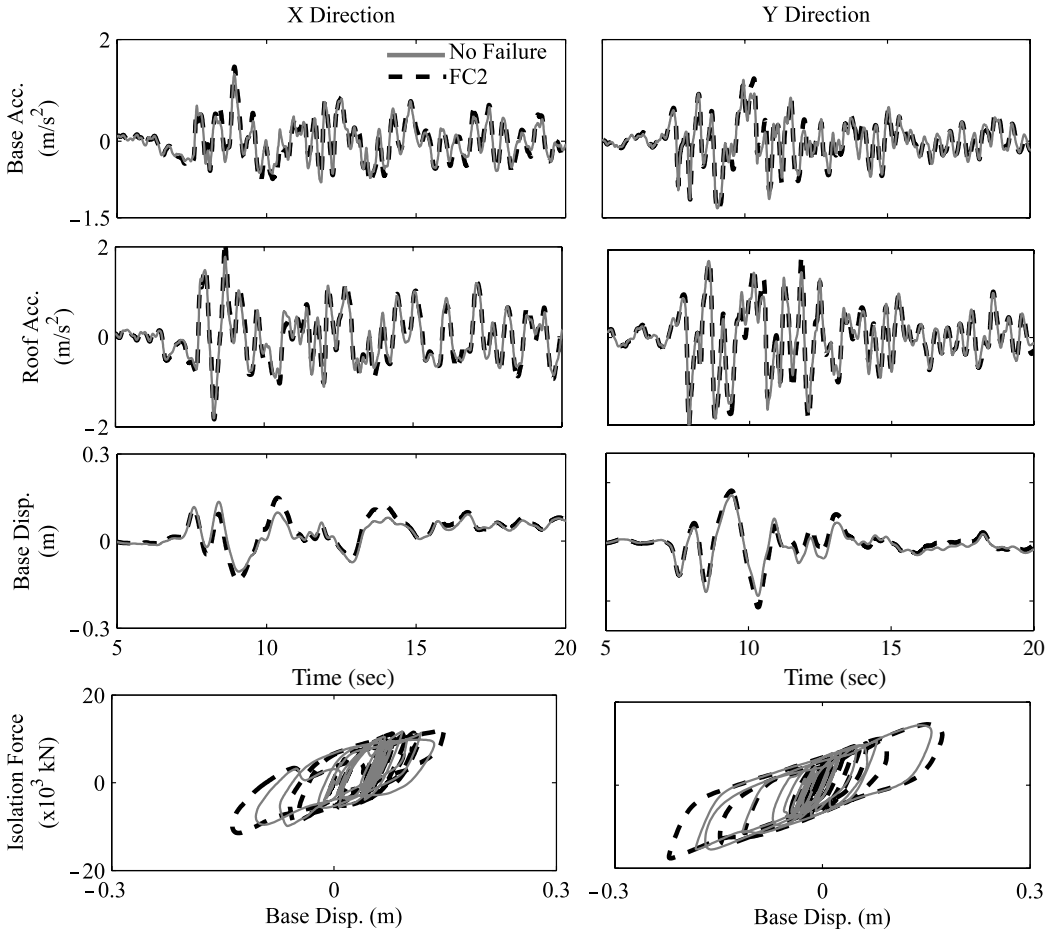

Fig. 9. Results of neural controller for Kobe earthquake — Base and roof accelerations, base displacement and total isolation force at the center of mass of base.

Table 4
Performance indices for adaptive controller under different failure scenarios.

Performance index	Case[a]	Newhall	Sylmar	El Centro	Rinaldi	Kobe	Erzinkan
J1	FC1	0.820	0.804	0.820	0.873	0.816	0.717
	FC2	0.813	0.798	0.788	0.869	0.779	0.701
Peak base shear	FC3	0.819	0.814	0.817	0.871	0.814	0.723
J2	FC1	0.807	0.824	0.782	0.873	0.826	0.717
	FC2	0.799	0.818	0.770	0.869	0.790	0.706
Peak structural shear	FC3	0.806	0.832	0.779	0.871	0.824	0.718
J3	FC1	0.782	0.740	0.542	0.727	0.794	0.651
	FC2	0.817	0.746	0.565	0.752	0.769	0.687
Peak base drift	FC3	0.775	0.719	0.549	0.726	0.795	0.638
J4	FC1	0.802	0.760	0.943	0.911	0.825	0.763
	FC2	0.790	0.753	0.961	0.906	0.833	0.752
Peak storey drift	FC3	0.796	0.764	0.969	0.910	0.829	0.770
J5	FC1	0.936	0.865	0.982	0.950	0.851	0.820
	FC2	0.915	0.860	0.986	0.944	0.856	0.791
Peak floor acceleration	FC3	0.929	0.867	0.982	0.947	0.853	0.823
J6	FC1	0.225	0.247	0.374	0.164	0.247	0.384
	FC2	0.231	0.233	0.353	0.152	0.337	0.375
Peak control force	FC3	0.225	0.253	0.338	0.158	0.270	0.371
J7	FC1	0.668	0.626	0.759	0.634	0.744	0.402
	FC2	0.699	0.640	0.588	0.680	0.897	0.452
RMS base drift	FC3	0.677	0.610	0.889	0.665	0.869	0.401
J8	FC1	0.915	0.725	0.975	0.830	0.960	0.620
	FC2	0.902	0.705	0.963	0.829	0.939	0.602
RMS floor acceleration	FC3	0.911	0.724	0.986	0.829	0.955	0.625

[a] FC1, FC2 and FC3 refer to failure cases 1, 2 and 3, respectively. Uncontrolled values are equal to 1 for all performance indices, except for J_6, which is 0 for uncontrolled case.

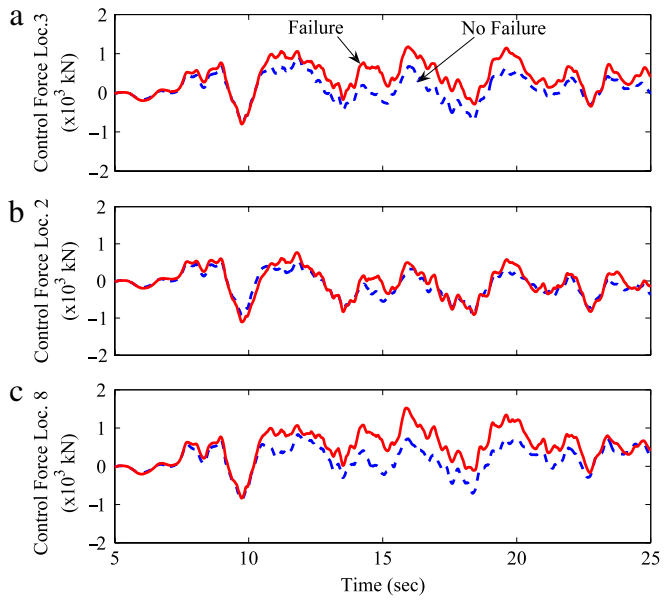

Fig. 10. Time histories of the adaptive control forces for Kobe earthquake in dominant direction: (a) failure case 1, (b) failure case 2, and (c) failure case 3.

Acknowledgements

The first and the second authors would like to thank Natural Sciences and Engineering Research Council of Canada (NSERC) — Discovery Grants Program, for their financial support to conduct this study. This study was partly undertaken during the first author's visit to the University of Waterloo in Spring 2007.

References

[1] Spencer BF, Nagarajaiah S. State of the art of structural control. J Struct Eng, ASCE 2003;129(7):845–56.
[2] Yang JN, Wu J, Reinhorn A, Riley M. Control of sliding-isolated buildings using sliding-mode control. J Struct Eng, ASCE 1995;122:179–86.
[3] Yang JN, Lin S, Jabbari F. H_∞ based control strategies for civil engineering structures. J Struct Control 2004;10:205–30.
[4] Cybenko G. Approximations by superpositions of sigmoidal functions. Math Control Signals Systems 1989;2:303–14.
[5] Masri S, Smyth A, Chassiakos A, Caughey T, Hunter N. Application of neural networks for detection of changes in nonlinear systems. J Eng Mech 2000; 126(7):666–76.
[6] Pei J, Smyth A. New approach to designing multilayer feed-forward neural network architecture for modeling nonlinear restoring forces. i: formulation. J Eng Mech 2006;132(12):1290–300.
[7] Bani-Hani J, Ghaboussi J. Nonlinear structural control using neural networks. J Eng Mech 1998;124(2):319–27.
[8] Chen H, Tsai G, Qi J, Yang F, Amini F. Neural network for structure control. J Comput Civil Eng 1995;9(2):168–76.
[9] Ghaboussi J, Joghataie A. Active control of structures using neural networks. J Eng Mech 1995;121(4):555–67.
[10] Liut DA, Matheu E, Singh M, Mook D. Neural-network of building structures by a force-matching training scheme. Earthquake Eng & Str Dyn 1999;28: 1601–20.
[11] Lee HJ, Yang G, Jung H, Spencer B, Lee I. Semi-active neuro-control of a base-isolated benchmark structure. Structural Control and Health Monitoring 2006; 13:682–92.
[12] Madan A. General approach for training back-propagation neural networks in vibration control of multi-degree of freedom structures. J Comput Civil Eng 2006;20:247–60.
[13] Sirca J, Adeli H. Counterpropagation neural network model for steel girder bridge structures. J Bridge Eng 2004;9(1):55–65.
[14] Lu YW, Sundararajan N, Saratchandran P. A sequential learning scheme for function approximation using minimal radial basis neural networks. Neural Comput 1997;9(2):1–18.
[15] Lu Y, Sundararajan N, Saratchandran P. Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. IEEE Trans Neural Netw, IEEE 1998;9(2):308–18.
[16] Li Y, Sundararajan N, Saratchandran P. Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems. IEEE Proc Control Theory Appl, IEEE 2000;147(4):476–84.
[17] Platt J. A resource-allocating network for function approximation. Neural Comput 1991;3(2):213–25.
[18] Broomhead D, Lowe D. Multivariable functional interpolation and adaptive networks. Complex Syst 1988;2:321–55.
[19] Moody J, Darken CJ. Fast learning in network of locally-tuned processing units. Neural Comput 1989;1:281–94.
[20] Ikhouane F, Manosa V, Rodellar J. Adaptive control of hysteretic structural system. Automatica 2004;41:225–31.
[21] Luo N, Rodellar J, Sen DL, Vehi J. Output feedback sliding mode control of base isolated structures. J Franklin Inst 2000;337:555–77.
[22] Pozo F, Ikhouane F, Pujol G, Rodellar J. Adaptive backstepping control of hysteretic base-isolated structures. J Vib Control 2006;12.
[23] Rumelhart G, Hinton G, Williams R. In: Rumelhart, McClelland, editors. Learning internal representations by error propagation. Parallel distributed processing, vol. 1. Cambridge (MA): MIT Press; 1986.
[24] Narendra KS. Neural networks for control: theory and practice. Proc IEEE 1996; 84(10):1385–406.
[25] Sanner RM, Slotine JE. Gaussian networks for direct adaptive control. IEEE Trans Neural Netw 1992;3(6):837–63.
[26] Ankireddi S, Yang H. Neural networks for sensor fault correction in structural control. J Struct Eng 1999;125(9):1056–64.
[27] Koh B, Li Z, Dharap P, Nagarajaiah S, Phan M. Actuator failure detection through interaction matrix formulation. J Guidance Control Dyn 2005;28(5):895–901.
[28] Li Z, Koh BH, Nagarajaiah S. Detecting sensor failure via decoupled error function and inverse input–output model. J Eng Mech, ASCE 2007;133(11): 1222–8.
[29] Pashlikar A, Sundararajan N, Saratchandran P. A fault-tolerant neural aided controller for aircraft auto-landing. Aerosp Sci Technol 2005;10(1):49–61.
[30] Narasimhan S, Suresh S, Nagarajaiah S, Sundararajan N. On-line learning failure-tolerant neural-aided controller for earthquake excited structures. J Eng Mech 2008;134(3):258–68.
[31] Contreras M, Nagarajaiah S, Narasimhan S. Fault tolerant neural-aided controller for multi degree of freedom structures experiencing online sensor failure. Adv Sci Technol 2008;56:247–52.
[32] Narasimhan S, Nagarajaiah S, Gavin H, Johnson EA. Base isolated benchmark building part I: problem definition. J Struct Control Health Monitoring 2006; 13(2):573–88.
[33] Nagarajaiah S, Reinhorn AM, Constantinou MC. Nonlinear dynamic analysis of 3-d-base-isolated structures. J Struct Eng, ASCE 1991;117(7):2035–54.
[34] Erkus B, Johnson EA. Smart base isolated benchmark building part III: a sample controller for bilinear isolation. J Struct Control Health Monitoring 2006;13(3): 605–25.
[35] Leontaritis IJ, Billings SA. Input–output parametric models for non-linear systems part I: deterministic non-linear systems. Int J Control 1985;41(2): 303–28.
[36] Suresh S, Omkar S, Mani V, Sundararajan N. Direct adaptive neural flight controller for f-8 fighter aircraft. J Guidance Control Dyn 2006;29(2):454–64.
[37] MATLAB. The Mathworks, Inc., Natick, Massachusetts; 2000.
[38] Nagarajaiah S, Narasimhan S. Base isolated benchmark building part II: phase I sample controllers for linear isolation systems. J Structural Control Health Monitoring 2006;13(2):589–604.
[39] Holland H. Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press; 1975.
[40] Goldberg D. Genetic algorithms in search, optimization and machine learning. New York: Addison-Wesley; 1989.
[41] Michalewicz Z. Genetic algorithms + data structures = evolution programs. AI series, New York: Springer-Verlag; 1994.
[42] Houck C, Jones J, Kay M. A genetic algorithm for function optimization: a matlab implementation. ACM Trans Math Software 1996;22:1–14.