Real time detection of stiffness change using a radial basis function augmented observer formulation

2011 Smart Mater. Struct. 20 035013

(http://iopscience.iop.org/0964-1726/20/3/035013)

Smart Mater. Struct. **20** (2011) 035013 (14pp) doi:10.1088/0964-1726/20/3/035013

Real time detection of stiffness change using a radial basis function augmented observer formulation

M Contreras[1], S Nagarajaiah[1,2,4] and S Narasimhan[3]

[1] Department of Civil and Environmental Engineering, Rice University, Houston, TX 77005, USA
[2] Department of Mechanical Engineering and Materials Science, Rice University, Houston, TX 77005, USA
[3] Department of Civil and Environmental Engineering, University of Waterloo, ON, N2L 3G1, Canada

E-mail: Satish.Nagarajaiah@rice.edu

Received 31 July 2010, in final form 18 January 2011
Published 9 February 2011
Online at stacks.iop.org/SMS/20/035013

Abstract
Existing methods for structural health monitoring pose a formidable challenge to real time implementation due to the significantly large computational loads. The proposed algorithm is suitable for online applications because it maintains good pattern recognition capabilities while possessing a computationally compact network topology. This study employs the computational efficiency of single layer radial basis function (RBF) approximaters to create a subspace capable of isolating faults in multi-degree of freedom systems which involve coupled and uncoupled stiffness changes in real time. The RBF network transforms the displacement–time history of the varying plant into a decoupled output space which is then compared to a baseline healthy observer which undergoes the same decoupling transformation. The online comparison of the output of the time varying plant and the healthy observer in a decoupled subspace comprises the observer based error function. The error function is shown to not only detect the existence of faults, but also isolate these faults in real time in the presence of base excitation. The method is validated for systems that experience earthquake induced damage, as well as an experimental system using a semi-active independent variable stiffness device which is capable of varying system stiffness in real time. By simply observing the displacement–time history responses, the RBF augmented observer formulation is capable identifying changes in the stiffness at each degree of freedom.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

The task of detecting real time stiffness change belongs to a subclass of problems dealing with fault detection and isolation (FDI). System stiffness variation can be regarded as a fault, and detection of that variation, representing structural damage, is critical to users or occupants. For systems which have many different types of faults, it is necessary to attempt the more challenging task of fault isolation, which determines the type of fault. Observer based formulations are a particular type of analytical redundancy strategy that have a longstanding history of demonstrating the capability to perform FDI in real time. In reference to the work of White [30], Massoumnia [19], Douglas [10], Liberatore [18], and very recently Chen and Nagarajaiah [2–4], most current research focuses on novel techniques to select a feedback detection filter matrix that attempts to decouple the closed loop observer error residual; however, few studies implement open loop observer formulations which strive to decouple the system output before creating an error residual. By using an interaction matrix formulation, input error functions, input–

[4] Address for correspondence: Department of Civil and Environmental Engineering, Rice University, 6100 Main Street, MS-318, Houston, TX 77005, USA.

output inverse models, and time segmented least squares to decouple time history signals, Koh *et al* [14, 15], Dharap *et al* [9, 8], Li *et al* [17] and Nagarajaiah *et al* [23] were able to perform real time damage detection within an open loop observer framework. The work presented in this paper seeks to unprecedentally combine open loop observer methods with radial basis function (RBF) networks to (1) alleviate the requirements associated with the design of a feedback detection filter matrix and (2) extend the previous signal decoupling capabilities for open loop systems by further generalizing the framework for subspace isolation. The main idea of the proposed work is to achieve separation in a transformed function subspace, rather than directly in the signal space. This transformation is achieved using the function approximation capabilities of artificial neural networks (ANN).

RBF networks are well known for their feedforward computational efficiency and require relatively small training data sets compared to other neural network methods such as back propagation [20]. These networks are also less susceptible to problems with non-stationary inputs due to the adaptive centers of the radial basis functions evaluated in the hidden processing elements. Their agility in providing succinct pattern recognition while not being computationally cumbersome make RBF networks prime candidates for real time fault detection and isolation.

Neural networks are ultimately a biologically inspired phenomena in which researchers have tried to mimic the topological flow of data across the complex network of axons and neurons located in the brain. Their applications has shown copious practical advantages over conventional methods for dynamic mappings of input–output data [26]. The mathematical representation of data flow is done primarily by a series of weighted summations and function evaluations occurring at specified processing elements, or neurons. Particularly, the RBF neural network is comprised of a single hidden layer of neurons consisting of locally tuned or locally sensitive neurons that produces an output tuned to some region of the input space. Furthermore, each hidden layer neuron is localized and decreases as a function of the distance of the input from that neuron's receptive field center. This localized tuning is made possible by the use of Gaussian basis functions, giving rise to the name RBF network. The output is a linear combination of the hidden layer function evaluation. RBF networks gain their computational advantage in that they rely heavily on resource allocation—proper selection of RBF parameters and robust weight updating of the output layer's linear units. This item of critical importance has been targeted in the past and the following paragraph addresses plausible solutions.

Platt developed a sequential learning algorithm, called the resource allocation network (RAN) [27]. In this algorithm, the number of hidden neurons is a function of the complexity of the problem and a new hidden neuron is added if the new data exhibits novelty, as defined by certain conditions. When the novelty conditions are not satisfied, the network parameters are adjusted using a least mean square (LMS) algorithm. This algorithm results in a compact network

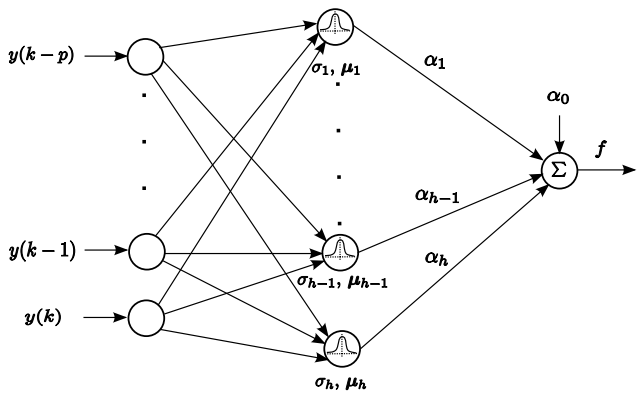

Figure 1. Radial basis function network topology.

compared to fixed-size networks, and learns quickly for real time applications. Kadirkamanathan and Niranjan replaced the LMS adaptation of network parameters by the extended Kalman filter (EKF) and showed through several examples that the resulting network, RANEKF, was even more compact than RAN [13].

In EKF and RANEKF methods, once a hidden neuron is created, it can never be removed. Because of this, RAN and RANEKF could produce networks in which some hidden units although active initially, may subsequently end up contributing little to the network output. For example, this phenomenon occurs in identification problems of nonlinear systems with changing dynamics resulting in a network with numerous inactive hidden neurons, as the dynamics that caused their creation initially become non-existent. Further the pruning can also help avoid the problem of over parameterization, which is a problem when neural networks are employed in control type formulations [29].

The extended minimal resource allocation network (EMRAN) has been developed by Li *et al* to employ the pruning concept [16]. The methodology behind the EMRAN formulation is to dynamically estimate the number of neurons in the hidden layer as a function of the complexity of the input data. This is accomplished through a combination of growing and pruning of hidden layer neurons. These additions make it suitable for online convergence while maintaining a compact network. Applications of the EMRAN formulation for online adaptive sequential learning applications involving building structures have been presented by Narasimhan *et al* [24] and Contreras *et al* [5, 6].

The compact RBF–EMRAN has been implemented in this study to decouple time history response by isolating relevant information to a separable subspace. The plant output and observer outputs are transformed into a RBF network subspace. Then, the residual error between the plant output and the observer output is constructed in the transformed domain. It will be shown through numerical simulations and experimental observations that the proposed methodology is effective in isolating the location and the onset of faults.

2. Network learning architecture

In the architecture illustrated by figure 1, input computations commence by first evaluating a general Gaussian function, ϕ_j,

Table 1. EMRAN initialized parameters.

$\epsilon_{\max}$	$\epsilon_{\min}$	ϵ_2	ϵ_3	γ	κ	N_w	S_w	δ	P_0	q
0.9699	0.1147	0.0328	0.0024	0.1911	0.6587	5.0	5.0	0.15	17.8613	25.0

at each neuron of which, h neurons comprises a single hidden layer having dimension $\boldsymbol{\phi} \in \Re^h$:

$$\phi_j = \exp^{\frac{-\|\mathbf{y}-\boldsymbol{\mu}_j\|^2}{(\sigma_j)^2}} \tag{1}$$

where $\mathbf{y} \in \Re^{p+1}$ is the vector comprising the inputs space, $\boldsymbol{\mu}_j \in \Re^{p+1}$ is the center of each Gaussian function, and $\sigma_j \in \Re^1$ is the standard deviation. The amount of points in the input space, p, is included here for generality and details regarding its determination will be addressed later in section 4.3. The weighted summation of the fully connected output layer produces scalar network output, $f \in \Re^1$, and is described by the following:

$$f = \alpha_0 + \sum_{j=1}^{h} \alpha_j \phi_j \tag{2}$$

where $\alpha_0 \in \Re^1$ is a scalar bias which assists in function mapping.

The network parameters are strategically selected through the training process based on several factors related to the error, $e \in \Re^1$, that exists between the training or desired network output, $f_d \in \Re^1$, and the actual network output produced by equation (2). The discretized training error is defined as:

$$e[k] = f[k] - f_d[k] \tag{3}$$

for the kth observation which is a particular data point at which the network parameters are updated or adapted. For the damage detection scheme proposed in this paper, f_d is provided *a priori* by training data sets, and the kth observation refers to the kth instant in time. Regarding the strategic update and selection of the network parameters based on the training error, they have been omitted here for clarity and readers are referred to [24] for a detailed explanation.

3. Network initialization

The EMRAN algorithm requires constants listed in table 1 to be initialized so that the network is suitable for online stiffness isolation. Selecting the appropriate parameters, involves the optimization of a complicated function which may have many local minima. Due to the large number of variables and the complexity of the function space, a gradient based method was not employed. Instead, a genetic algorithm (GA) which was developed to mimic the energy minimization that occurs in the evolutionary process of natural selection was utilized. GAs were developed by Holland in 1975 and they implement techniques such as genetic crossover and mutation to determine future populations of individuals [12]. By representing an individual by a coded string, in this case the EMRAN parameters with fixed bounds on their variance, future generations of the string further optimize those constants.

A GA produces robust populations by maximizing an overall fitness function, which in this case is the root mean squared value of error produced from equation (3). Since the RMS value of equation (3) is always positive and the error is desired to be small, the following quantity or fitness function is maximized, $-[e[k]]^2$, to ensure an error close to zero. An initial population size (number of search nodes) are selected and by maximizing the fitness function, the GA produces EMRAN parameters that will allow the RBF network to optimally approximate the dynamics of the systems introduced in the following sections. The genetic mutation, crossover, selection probabilities, maximum number of generations, and population size can be found in [24].

It is important to note that once the EMRAN parameters are selected by the GA they remain fixed for a given application. The RBF network implemented in the proposed formulation is fairly robust to the initial EMRAN parameters. However, the use of the GA contributes a rigorous determination of those values, thereby ensuring network convergence. The values produced by the GA can be found in table 1.

4. RBF augmented observer formulation

Two factors that hinder the application of open loop observers for FDI are (1) their susceptibility to produce unbounded residuals and (2) the necessary creation of separable fault detection subspaces for the system output [11]. RBF networks resolve these two shortcomings, and by augmenting an open loop observer in the configuration proposed in figure 2, they can be adapted for online applications. Details are provided in this section to help elucidate why RBF networks are effective in generating separable subspaces.

4.1. Damage detection via bounded error residual

By implementing the proposed formulation, the output of a faulty system and the output of a healthy observer are both transformed to a nonlinear radial basis function subspace. It is the error residual produced by comparing these two quantities in real time that is capable of detecting damage. The section 4.2 will go through some of the key assumptions necessary for detecting damage. Since the damage detection relies heavily on the error function producing either zero or nonzero finite output, it is important to ensure under no circumstances does its output tend toward infinity. Topics related to bounding of the error function and limitation on system type will be addressed.

Suppose some faulty dynamic system can be characterized by the following linear time invariant (LTI) state space

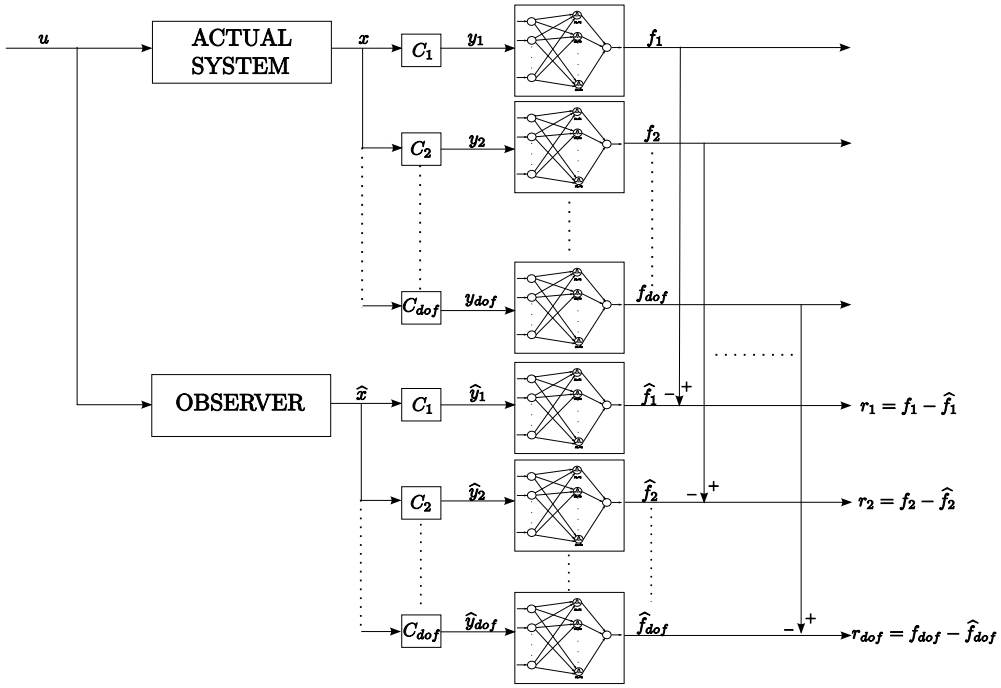

Figure 2. RBFN augmented observer formulation.

representation where $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathcal{Y}$, $\mathbf{u} \in \mathcal{U}$ with $n = \dim(\mathcal{X})$, $m = \dim(\mathcal{Y})$, and $r = \dim(\mathcal{U})$.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \sum_{i=1}^{q_n} \mathbf{F}_i m_i(t) \qquad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{4}$$

$\mathbf{A}$ is an $n \times n$ system state transmission matrix, $\mathbf{B}$ is an $n \times r$ input influence matrix, and $\mathbf{C}$ is an $m \times n$ output influence matrix. It was shown by Beard [1], that by defining, $\mathbf{F}_i$ as an $n \times 1$ fault direction vector, and $m_i(t)$ as the ith arbitrary scalar function of time ($i = 1, 2, \ldots, q$) for q different faults, actuator, sensor, and structural faults can be described when $m_i(t) \neq 0$ [1].

Assuming an accurate observer of the healthy system can be created (i.e. $\hat{\mathbf{A}} = \mathbf{A}$, $\hat{\mathbf{B}} = \mathbf{B}$, and $\hat{\mathbf{C}} = \mathbf{C}$), the following expression for $\hat{\mathbf{x}} \in \mathcal{X}^n$ and $\hat{\mathbf{y}} \in \mathcal{Y}^m$ describes the estimated state for identical bounded input, $\mathbf{u}(t)$, without the effect of faults.

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) \qquad \hat{\mathbf{y}}(t) = \mathbf{C}\hat{\mathbf{x}}(t). \tag{5}$$

For the proposed formulation presented in this paper, the $\mathbf{C}$ matrix is equivalent to $\mathbf{C}_{\text{dof}}$ which is defined as:

$$\mathbf{C}_{\text{dof}} \Rightarrow C_{1j} = \begin{cases} 1 & \forall\, j = \text{dof} \\ 0 & \forall\, j \neq \text{dof} \end{cases} \tag{6}$$

where the dof subscript indicates the degree of freedom or floor where measurements are being taken. Considering a particular degree of freedom's output for the faulty system, $y = y_{\text{dof}}$ produced by $\mathbf{C} = \mathbf{C}_{\text{dof}}$, the baseline observer in equations (5) will produce corresponding output $\hat{y} = \hat{y}_{\text{dof}}$. By storing $p + 1$ points of data, a vector $\mathbf{y} \in \Re^{p+1}$ and $\hat{\mathbf{y}} \in \Re^{p+1}$ for the kth instant can be created.

$$\begin{aligned} \mathbf{y} &= [y_{\text{dof}}(k-p) \cdots y_{\text{dof}}(k-1)\ y_{\text{dof}}(k)]' \\ &= [\mathbf{C}_{\text{dof}}\mathbf{x}(k-p) \cdots \mathbf{C}_{\text{dof}}\mathbf{x}(k-1)\ \mathbf{C}_{\text{dof}}\mathbf{x}(k)]' \end{aligned} \tag{7}$$

$$\begin{aligned} \hat{\mathbf{y}} &= [\hat{y}_{\text{dof}}(k-p) \cdots \hat{y}_{\text{dof}}(k-1)\ \hat{y}_{\text{dof}}(k)]' \\ &= [\mathbf{C}_{\text{dof}}\hat{\mathbf{x}}(k-p) \cdots \mathbf{C}_{\text{dof}}\hat{\mathbf{x}}(k-1)\ \mathbf{C}_{\text{dof}}\hat{\mathbf{x}}(k)]'. \end{aligned} \tag{8}$$

The RBFN hidden layer then performs a basis transformation on $\mathbf{y}$ and $\hat{\mathbf{y}}$ described by equation (1), for a function Φ: $\mathbf{y} \to \boldsymbol{\phi}$, where $\boldsymbol{\phi} \in \Re^h$. The total network output can be described by function composition, $(f \circ \Phi)(\mathbf{y})$: $\Re^{p+1} \to \Re^1$, adapted from equation (2) and shown below for f and $\hat{f}$,

$$f = \alpha_0 + \sum_{j=1}^{h} \alpha_j \phi_j(\mathbf{y}) \tag{9}$$

$$\hat{f} = \alpha_0 + \sum_{j=1}^{h} \alpha_j \phi_j(\hat{\mathbf{y}}). \tag{10}$$

Due to the properties of unscaled Gaussian functions with fixed centers and standards of deviation (see equation (1)), the following will hold true.

$$0 < \phi_j(\mathbf{y}) \leqslant 1 \qquad \forall\, \mathbf{y} \in \Re^{p+1}. \tag{11}$$

From equation (11), it follows that f and $\hat{f}$ remain in some ball of radius, $\beta_1, \beta_2 \in \beta$. This is conveyed by the following expressions:

$$\left| \alpha_0 + \sum_{j=1}^{h} \alpha_j \phi_j(\mathbf{y}) \right| \leqslant \beta_1 \qquad \forall\, \mathbf{y} \in \Re^{p+1} \tag{12}$$

and,

$$\left|\alpha_0 + \sum_{j=1}^{h} \alpha_j \phi_j(\hat{\mathbf{y}})\right| \leqslant \beta_2 \qquad \forall\, \mathbf{y} \in \Re^{p+1} \tag{13}$$

since $(f \circ \Phi)(\mathbf{y})$ is invariant to input from $\hat{\mathbf{y}}$, the error residual, r, shown in equation (14) is also bounded.

$$|r| = |f - \hat{f}| \leqslant \beta \qquad \forall\, \mathbf{y} \in \Re^{p+1}. \tag{14}$$

Now that the error residual has been shown to be bounded, it is only plausible that it produces either zero values, or nonzero finite values. The bounded condition on the residual is necessary, but not sufficient for fault detection. The formulation requires a further assumption that the state transition matrix, $\mathbf{A}$, be stable. For the structural systems presented in this paper this is true; however, there exist systems which are input to state stable, or stabilizable by input, of which this formulation is not capable of detecting damage. Reasons as to why the system must be stable are elucidated further in section 4.2.

4.2. Fault isolation using the error residual

The solutions for both the faulty system and the observer are as follows:

$$\mathbf{x}(t) = \mathrm{e}^{\mathbf{A}t\mathbf{x}(0)} + \int_0^t \mathrm{e}^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)\,\mathrm{d}\tau - \sum_{i=1}^{q_n} \int_0^t \mathrm{e}^{\mathbf{A}(t-\tau)}\mathbf{F}_i m_i(\tau)\,\mathrm{d}\tau \tag{15}$$

$$\hat{\mathbf{x}}(t) = \mathrm{e}^{\mathbf{A}t}\hat{\mathbf{x}}(0) + \int_0^t \mathrm{e}^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)\,\mathrm{d}\tau. \tag{16}$$

It can be seen for the systems considered in this study (stable $\mathbf{A}$ and bounded, scalar $u(t)$) the transient terms, $\mathrm{e}^{\mathbf{A}t}\mathbf{x}(0)$ and $\mathrm{e}^{\mathbf{A}t}\hat{\mathbf{x}}(0)$ will tend to zero and the remaining terms tend toward constant values, provided by their Taylor expansions, yielding expressions (17) and (18). It should be noted that the following derivation is also valid for systems with modeled noise. For simplicity, it is not included here, but the end result still holds true. For the details regarding the treatment of noise for FDI, readers are referred to the work of Douglas on robust H_∞ bounded detection filters [10].

$$\mathbf{x} = [\psi_0\mathbf{A}^0\mathbf{B} + \psi_1\mathbf{A}^1\mathbf{B} + \psi_2\mathbf{A}^2\mathbf{B} + \cdots] - \sum_{i=1}^{q_n}[\Gamma_{i0}\mathbf{A}^0\mathbf{F}_i + \Gamma_{i1}\mathbf{A}^1\mathbf{F}_i + \Gamma_{i2}\mathbf{A}^2\mathbf{F}_i + \cdots] \tag{17}$$

$$\hat{\mathbf{x}} = [\psi_0\mathbf{A}^0\mathbf{B} + \psi_1\mathbf{A}^1\mathbf{B} + \psi_2\mathbf{A}^2\mathbf{B} + \cdots] \tag{18}$$

where in equations (17) and (18), ψ_l and Γ_{il}, are defined by,

$$\psi_l = \int_0^t \frac{(t-\tau)^l}{l!} u(\tau)\,\mathrm{d}\tau \tag{19}$$

$$\Gamma_{il} = \int_0^t \frac{(t-\tau)^l}{l!} m_i(\tau)\,\mathrm{d}\tau. \tag{20}$$

The solutions in equations (17) and (18) only differ by the contribution of the fault term and can be rewritten as follows,

$$\mathbf{x} = \mathbf{\Delta}_B - \mathbf{\Delta}_F \tag{21}$$

$$\hat{\mathbf{x}} = \mathbf{\Delta}_B \tag{22}$$

where

$$\mathbf{\Delta}_B = \sum_{l=1}^{\infty} \psi_l \mathbf{A}^l \mathbf{B} \tag{23}$$

and

$$\mathbf{\Delta}_F = \sum_{i=1}^{q_n} \boldsymbol{\delta}_{F_i} = \sum_{i=1}^{q_n}\sum_{l=1}^{\infty} \Gamma_{il}\mathbf{A}^l\mathbf{F}_i. \tag{24}$$

By plugging in the equations (21) and (22) into equations (4) and (5), the output of a particular degree of freedom ($\mathbf{C} = \mathbf{C}_{\mathrm{dof}}$) for both the system and the observer can be written by the expressions below, respectively.

$$y_{\mathrm{dof}} = \mathbf{C}_{\mathrm{dof}}(\mathbf{\Delta}_B - \mathbf{\Delta}_F) \tag{25}$$

$$\hat{y}_{\mathrm{dof}} = \mathbf{C}_{\mathrm{dof}}(\mathbf{\Delta}_B). \tag{26}$$

The network input vector for both the faulty system and the observer are constructed according to equations (7) and (8) and propagated through the Gaussian function as shown below. For simplicity of illustration, scalar, μ_j, is assumed.

$$\phi_j = \exp\left[\frac{-||\mathbf{C}_{\mathrm{dof}}\mathbf{\Delta}_B - \mathbf{C}_{\mathrm{dof}}\mathbf{\Delta}_F - \mu_j||^2}{\sigma_j^2}\right] \tag{27}$$

$$\hat{\phi}_j = \exp\left[\frac{-||\mathbf{C}_{\mathrm{dof}}\mathbf{\Delta}_B - \mu_j||^2}{\sigma_j^2}\right]. \tag{28}$$

By assuming the following expression regarding μ_j,

$$\mu_j = \mu_B - c_{F_m} \tag{29}$$

where μ_B is defined to be some scalar center capable of localizing the input space, and c_{F_m} is a center capable of distancing appropriately perturbations of the $\mathbf{F}_m$th fault relative to all faults, $\mathbf{F}_i$. Equations (27) and (28) can be simplified for systems which are adequately tuned to the input, (i.e. $\mathbf{C}_{\mathrm{dof}}\mathbf{\Delta}_B - \mu_B = 0$).

$$\phi_j = \exp\left[\frac{-||c_{F_m} - \mathbf{C}_{\mathrm{dof}}\mathbf{\Delta}_F||^2}{\sigma_j^2}\right] \tag{30}$$

$$\hat{\phi}_j = \exp\left[\frac{-||c_{F_m}||^2}{\sigma_j^2}\right]. \tag{31}$$

Equation (30) can be expressed as a summation of $\mathbf{F}_i$ fault terms with P_i faulty magnitude scaling, by the following:

$$\phi_j = \exp\left[\frac{-||c_{F_m} - P_1\mathbf{C}_{\mathrm{dof}}\boldsymbol{\delta}_{F_1} - P_2\mathbf{C}_{\mathrm{dof}}\boldsymbol{\delta}_{F_2} - \cdots||^2}{\sigma_j^2}\right]. \tag{32}$$

It is desired that equation (32) express only damaged values ($0 < \phi_j < 1$) in the presence of the $\mathbf{F}_m$th fault. For this to be true the substitution, $\sigma_j = \sigma_{F_m}$, is made. Next, c_{F_m} and σ_{F_m} must be selected such that the following condition is fulfilled:

$$\frac{c_{F_m} - P_m\mathbf{C}_{\mathrm{dof}}\boldsymbol{\delta}_{F_m}}{\sigma_{F_m}} \ggg \frac{P_i\mathbf{C}_{\mathrm{dof}}\boldsymbol{\delta}_{F_i}}{\sigma_{F_m}} \qquad \forall\, i \neq m \tag{33}$$

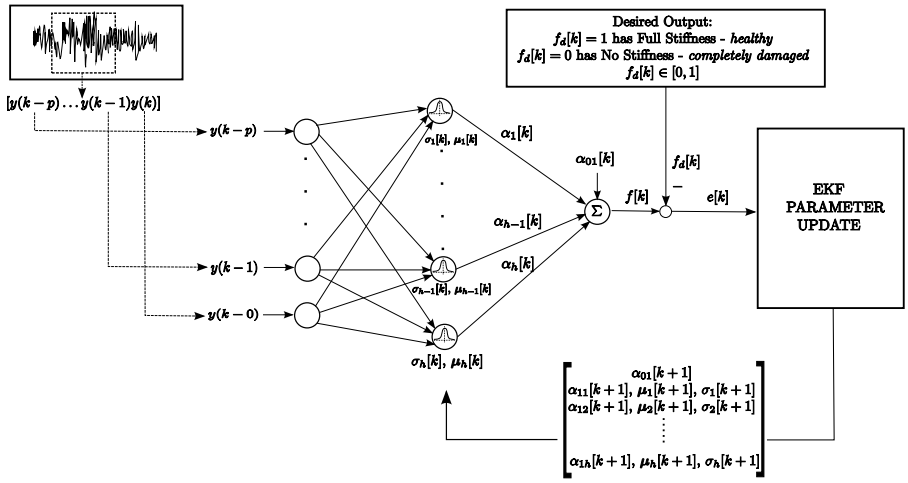

Figure 3. Single iteration of recursive training scheme.

where $\frac{P_i \mathbf{C}_{\text{dof}} \delta_{F_i}}{\sigma_{F_m}}$ is sufficiently small. If equations (29) and (33) are true, the proposed detection filter equation (14) is sufficient for fault detection and isolation. More explicitly, for fault $\mathbf{F}_m$, even if there exists the presence of faults $\mathbf{F}_i$, $r_j = \alpha_j(\phi_j - \hat{\phi}_j) \rightarrow 0$, as $\mathbf{F}_m \rightarrow \mathbf{0}$. This ensures that the contribution of the jth hidden layer neuron is maximized when a particular fault is present and consequently produces a nonzero weighted network output which indicates damage. Finally, it is worth noting that $\mathbf{C}_{\text{dof}}$ must be constructed such that it corresponds to the degree of freedom in which damage occurs for the formulation to function properly (e.g. $\mathbf{C}_1$ is capable of detecting damage between the first and second floor, $\mathbf{C}_2$ is capable of detecting damage between the second and third floor, and so on). If this is not the case, the linear combination of $\mathbf{C}_{\text{dof}}$ and equation (24) will be zero yielding equation (30) equivalent to equation (31) for both $\mathbf{F}_m$ present and $\mathbf{F}_m$ absent. This equality incorrectly produces a healthy zero residual when fault, $\mathbf{F}_m$, is present.

Therefore, the solution to the fault isolation problem given the proposed RBFN augmented observer formulation lies in adequately tuning the centers, $\boldsymbol{\mu}_j$, and standards of deviation, σ_j, of h number of basis functions such that the $\mathbf{F}_m$th faulty contribution to the error residual can be separated from all remaining faults that may or may not exist in the system. It is a further proposition of this study that this desired level of tuning and separation can be accomplished by a generalized training scheme described in detail in section 4.3.

Specifically regarding tuning, due to the excellent localization properties of RBF networks with adequate training data sets, it is possible to isolate faults that are sufficiently distinct. This statement is verified using both numerical and experimental results. Further, theoretical justification for separation capability of the RBFs is provided by Cover's theorem on the separability of patterns [7]. In Cover's theorem on the separability of patterns, he was able to prove that a pattern classification problem cast in a nonlinear space of higher dimension is more likely to be separable than in a low dimensional space [7]. It is illustrated by the following validations that given an adequate training set, and maximum dimensionality of the RBF hidden layer, separation is not only possible, but robust toward a variety of inputs and damage scenarios.

4.3. Training the network for fault isolation

As shown previously, the fault isolation characteristic of the RBFN relies on the selection of RBF parameters. The training scheme presented in this section has been shown to (1) be general enough to apply to a wide class of problems and (2) adequately tune the RBF parameters such that they can isolate coupled and uncoupled damage.

In order to create the necessary mapping from displacement to separable fault detection space, the neural network has to be trained offline using perturbations of the system model. The systems in this study are subjected to varying levels of damage by changing the stiffness in each DOF. Corresponding displacement data is generated. The training input to the RBF network is comprised of $p+1$ preceding displacement values in each DOF at a certain instant k. The desired training output is the value of the stiffness term known *a priori* that corresponds to those $p+1$ displacements.

Regarding the determination of a suitable value for p, this depends on the amount of network updates needed during training with minimum criteria $p \geqslant n$, where n is the dimension of the observed system output. Since the network updates in this study occur at $p+1$, $2p+1$, $3p+1$, ... data points and assuming sections 5.1, 5.2 and 6 examples have the same total amount of data points, larger p values produce less network updates during training. Larger values for p are reserved for the system that exhibits a steady state response (section 6). The exact dimension of p for the examples in this study is determined on a trial and error basis; however, a minimum selection of $p = n = 1$ will produce the most amount of network updates, but it is always the least computationally efficient selection that produces robust results.

The learning process in a single iteration of training is as follows (see figure 3). First, $p+1$ displacements preceding and including instant, k, comprise input vector, $\mathbf{y}$, and $\mathbf{y}$ is fed into the RBF network. Based on the size of the network's

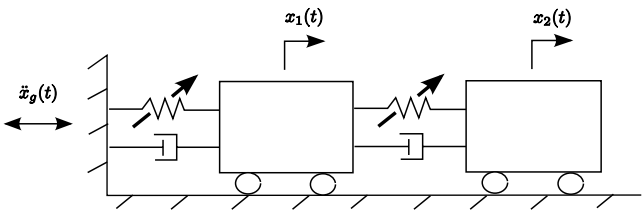

Figure 4. Two degree of freedom system with variable stiffness.

hidden layer and value of the weights, the RBF network then computes an output. Next, the computed output is compared to the desired output which varies between 0—no stiffness- and 1—full stiffness. If the error between network output and desired training output is large the tuning parameters are updated optimally using the EKF algorithm outlined in [24]. The training process moves to another instant, k, and performs the same computations. At that instant, if the error between computed and desired output is sufficiently small, then the weights and parameters cease to be updated. The RBF network is now ready to be implemented for testing.

It should be noted that during this training process, it is not necessary for the exact testing damage case to be shown to the network. The main motivation for the training process is that after seeing a general variety of damage scenarios, the RBFN can adequately classify in real time a test pattern similar, but not exactly the same, to those patterns seen during training.

5. Simulation using earthquake input

Before verifying the formulation experimentally, the method is first implemented analytically on a two degree of freedom mass–spring–damper system shown in figure 4 and a ten degree of freedom mass–spring–damper system shown in figure 5. Both models were subjected to coupled fault scenarios in the presence of the Mexico City (1985 EW) and Jiji (NS) quake cases and uncoupled fault scenarios in the presence of the Mexico City (1985 EW) and Jiji (NS) quake cases. The results in both cases are presented in subsequent sections. During certain pulses of the quake excitations, both systems experience a fault scenario signified by a decrease in stiffness in selected degrees of freedom. The formulation is shown to detect both coupled and uncoupled stiffness changes in the simulations.

Throughout the remainder of this paper the state space representation for a general multi-degree of freedom system subjected to input excitation and whose output is displacements at each DOF is implemented (described below by equations (34)).

Table 2. Two DOF system properties.

ω (rad s^{-1})	ζ (%)
$\omega_1 = 7.67$	$\zeta_1 = 1.8$
$\omega_2 = 19.92$	$\zeta_2 = 10.2$

$$\begin{aligned} \dot{\mathbf{x}}_{2n\times 1}(t) &= \mathbf{A}_{2n\times 2n}\mathbf{x}_{2n\times 1}(t) + \mathbf{B}_{2n\times 1}u(t) \\ \mathbf{y}_{n\times 1}(t) &= \mathbf{C}_{n\times 2n}\mathbf{x}_{2n\times 1}(t). \end{aligned} \tag{34}$$

The state variable, $\mathbf{x}_{2n\times 1}$, consists of relative displacement, $\mathbf{x}_{n\times 1}$, and relative velocity, $\dot{\mathbf{x}}_{n\times 1}$, for all degrees of freedom. $u(t)$ is the input acceleration and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are state space matrices defined as follows:

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

$$\mathbf{B}_{2n\times 1} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{\Gamma}_{n\times 1} \end{bmatrix}$$

and

$$\mathbf{C}_{n\times 2n} = [\,\mathbf{I}_{n\times n} \quad \mathbf{0}_{n\times n}\,]$$

where $\mathbf{M}$, $\mathbf{C}^{\text{damp}}$ and $\mathbf{K}$ are matrices for mass, damping, and stiffness, respectively. $\mathbf{\Gamma}$ is a vector of unity influence coefficients corresponding to the degrees of freedom in which the system is receiving input acceleration.

It is commonly known that the linear time invariant state space representation in equations (34) can describe a general mass–spring–damper structure or shear type building of arbitrary, n, degrees of freedom. By setting $n = 2$ and $n = 10$ respectively, the proper dimension of the simulation models displayed in figures 4 and 5 are set and their time history responses to input can be simulated. The state space models are used to represent both the actual system and the healthy observer in both the $n = 2$ and 10 validations. Damage is induced by varying the stiffness values of the appropriate DOF in matrix, $\mathbf{K}$, during select time intervals and simulating the time varying response.

5.1. Results for 2-DOF mass–spring–damper system

The effectiveness of the RBF algorithm is demonstrated using a 2-DOF system subjected to the earthquake cases mentioned previously. The 2-DOF system in figure 4 has the modal parameters found in the table 2. This model is used to generate proper training time histories that validate the FDI capabilities of the RBFN formulation. The network is trained as described in section 4.3. During training, a sinusoidal

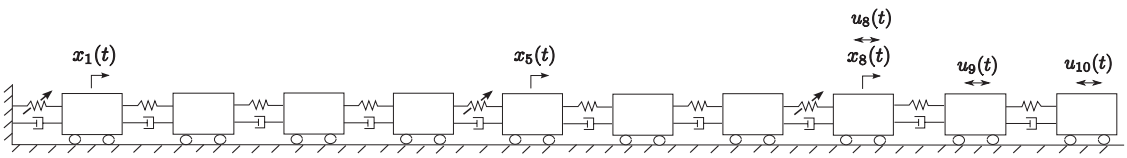

Figure 5. Ten degree of freedom system with variable stiffness.

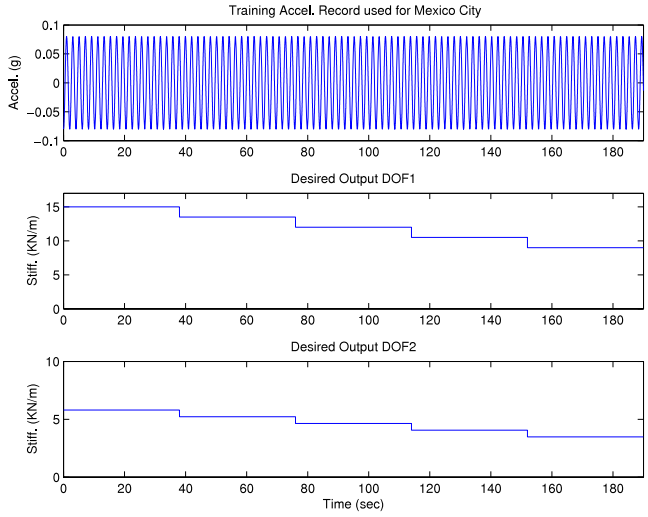

Figure 6. Training Acc. and desired stiffness output for Mexico City.

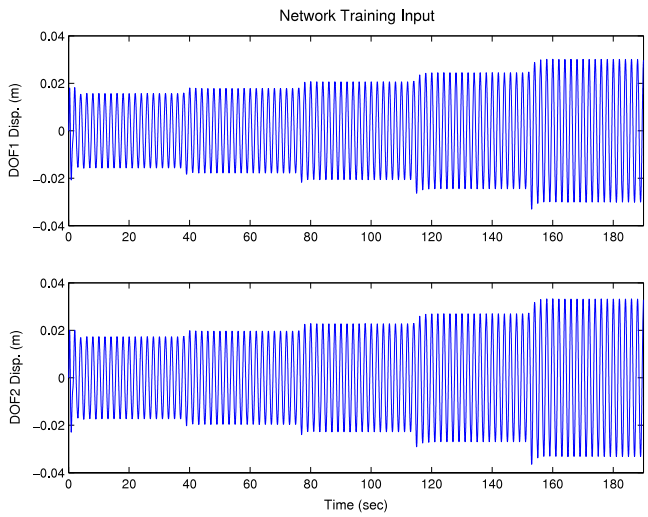

Figure 7. Network training input produced by Acc. in figure 6 (top).

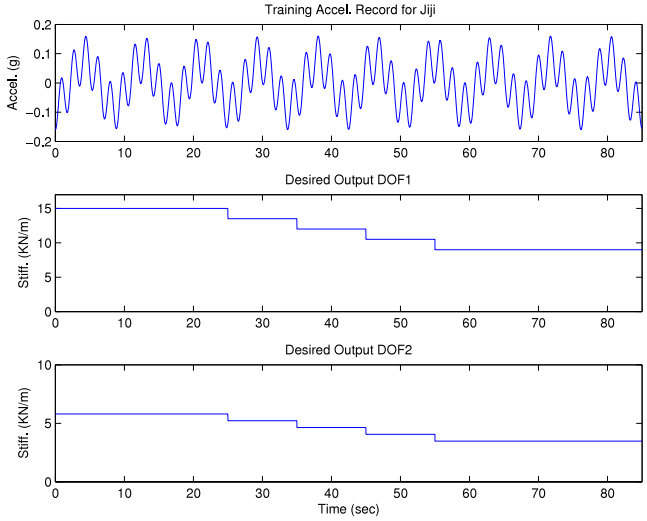

Figure 8. Training Acc. and desired stiffness output for Jiji.

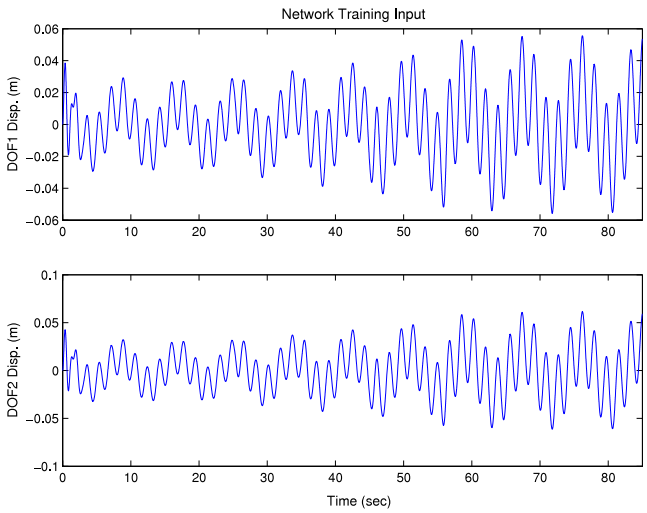

Figure 9. Network training input produced by Acc. in figure 8 (top).

excitation is used to drive the system while the stiffness in each degree of freedom is damaged an incremental amount. To facilitate with the localization process, equation (29), the frequencies of the training excitation are based on the dominant frequencies of excitation that the system will experience during testing. These dominant frequencies are determined beforehand from site specific response spectra for earthquake excitations and Fourier spectrum for harmonic excitations. The frequencies of the training excitations are as follows: 0.5 Hz for the Mexico City quake and 0.1 and 0.6 Hz for the Jiji quake. The Mexico City and Jiji training functions were used consistently with both 2-DOF and 10-DOF models. The resulting displacement response serves as training input to the network and the incremental damage as desired output. The sinusoidal excitation, damage scenarios, and resulting response used to train the network in the case of the Mexico City verification are shown in figures 6 and 7. The sinusoidal excitation, damage scenarios, and resulting response used to train the network for the Jiji verification are shown in figures 8 and 9.

After training, the weights are set. The test damage scenarios are meant to mimic damage in a building with stiffness losses occurring at designated time instances corresponding to larger pulses within the quake record. Regarding the selection of the dimension of the input vector, it is selected such that $\mathbf{y} \in \Re^{11}$ at instant k for $p = 10$ and is consistent for both training and testing. The simulation sampling rate is 1000 Hz. The following illustrative results are taken from the coupled damage scenario verifications. Uncoupled damage scenario results were omitted due to space.

The excitations and damages scenarios for the 2-DOF system subjected to the Mexico city case are shown in figure 10. In DOF 1, the system remains healthy until 86 s when it experiences a 10% stiffness loss. In DOF 2, the system experiences an 8% stiffness loss at 46 s. The error functions track the onset and existence of damage in each DOF accurately as evidenced by the error functions shown in figure 11.

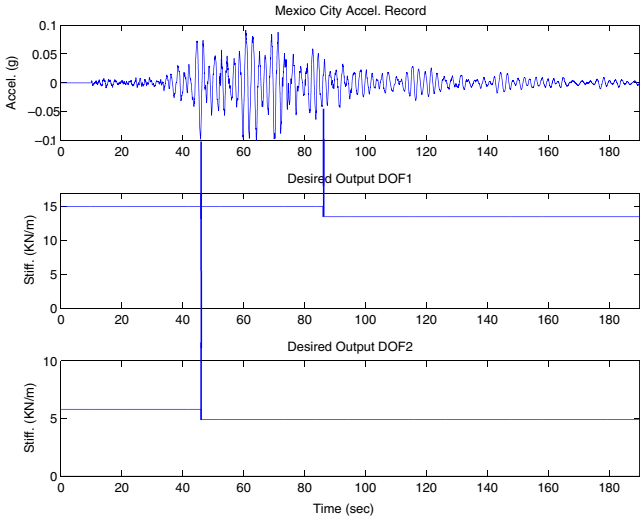

Figure 10. Mexico City earthquake record with coupled real time stiffness loss.

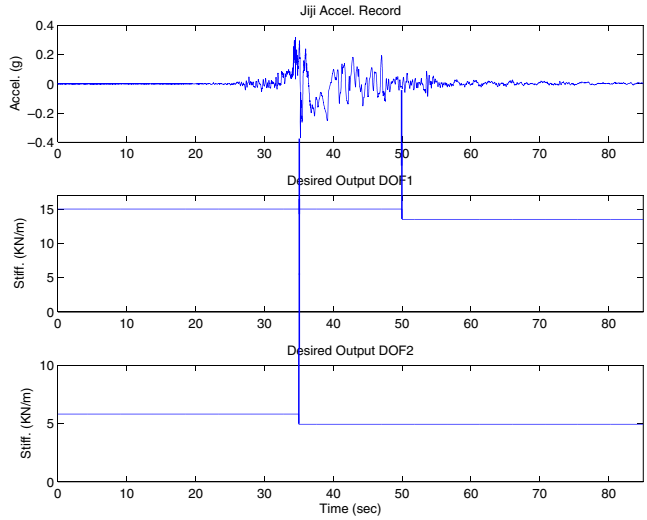

Figure 12. Jiji earthquake record with coupled real time stiffness loss.

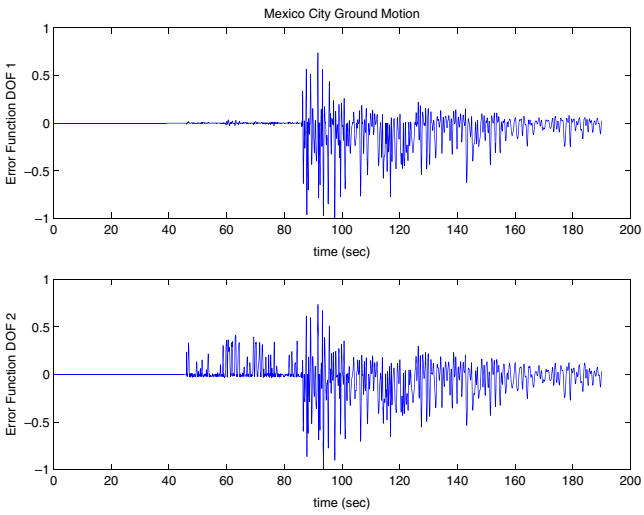

Figure 11. Augmented RBFN detection filter error function (equation (14)) for Mexico City case.

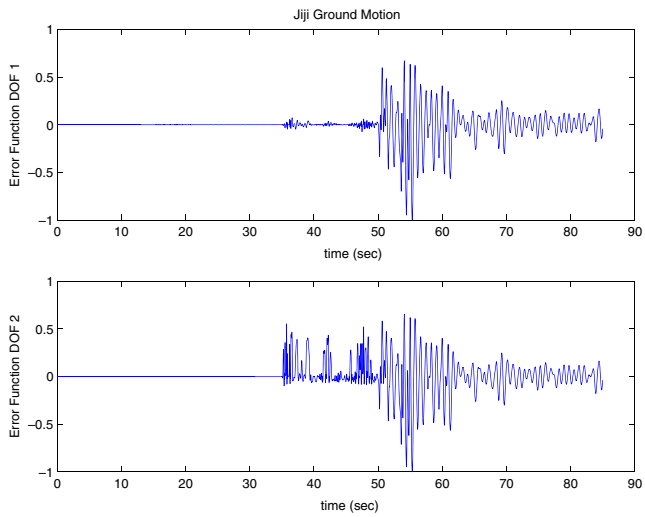

Figure 13. Augmented RBFN detection filter error function (equation (14)) for Jiji case.

The excitations and damages scenarios for the 2-DOF system subjected to the Jiji quake case are shown in figure 12. In DOF 1, the system remains healthy until 50 s when it experiences a 10% stiffness loss. In DOF 2, the system experiences a 15% damage at 35 s. The RBFN formulation is capable of identifying and isolating the previously described damage scenario as evidenced by the error functions shown in figure 13.

5.2. Results for 10-DOF mass–spring–damper system with non-collocated input and output

To validate the proposed formulation further, a 10-DOF system is examined. Much differently than the 2-DOF simulation, input was induced at only three of 10 DOFs (8, 9, and 10), and data is only being observed at another three of the 10 DOFs (1, 5, and 8). The input/output configuration and the damaged DOFs are depicted in figure 5 and the modal parameters of the healthy system are shown in table 3. Verifying the RBFN on a larger 10-DOF model with non-collocated input and output demonstrates that the proposed formulation is viable for systems without input actuation and sensing at the same locations. The 10-DOF model was subjected to coupled fault scenarios in the presence of the Mexico City (1985 EW) and Jiji (NS) quake cases and uncoupled fault scenarios in the presence of the Mexico City (1985 EW) and Jiji (NS) quake cases; however, due to space limitations only the coupled in time damage scenarios for the Mexico City quake are shown, figure 16 (left), and the uncoupled in time damage scenarios for the Mexico City quake case are shown, figure 17 (left).

Once again, the network is trained per section 4.3. The same training input excitations implemented in section 5.1 are applied to DOFs 8–10. During the forced vibration training each degree of freedom is damaged an incremental amount,

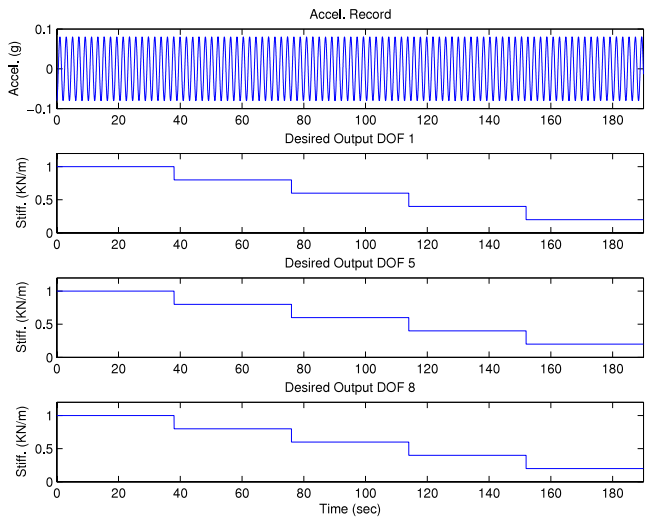

Figure 14. Training Acc. and stiffness change in 10-DOF structure for Mexico City case.

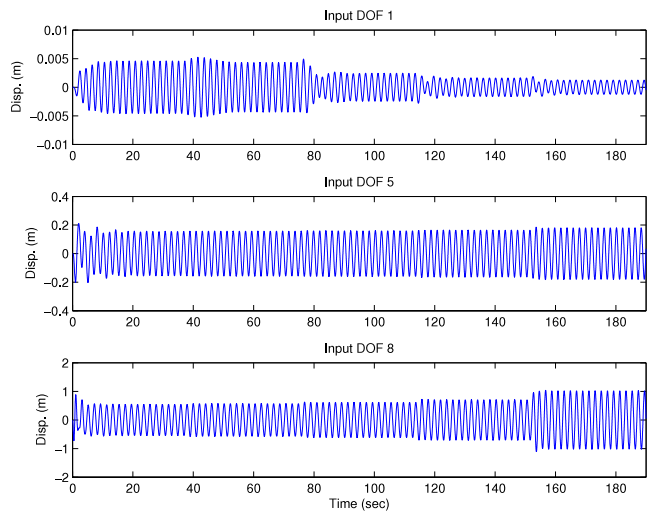

Figure 15. Network training input at selected DOFs produced by Acc. in figure 14 (top).

Table 3. Ten DOF system properties.

ω (rad s^{-1})	ζ (%)
$\omega_1 = 1.12$	$\zeta_1 = 0.15$
$\omega_2 = 3.19$	$\zeta_2 = 0.09$
$\omega_3 = 6.00$	$\zeta_3 = 0.10$
$\omega_4 = 21.30$	$\zeta_4 = 0.27$
$\omega_5 = 31.12$	$\zeta_5 = 0.39$
$\omega_6 = 41.86$	$\zeta_6 = 0.53$
$\omega_7 = 51.45$	$\zeta_7 = 0.65$
$\omega_8 = 54.44$	$\zeta_8 = 0.68$
$\omega_9 = 72.78$	$\zeta_9 = 0.91$
$\omega_{10} = 127.00$	$\zeta_{10} = 1.59$

and the desired stiffness output for the Mexico City training case is shown in figure 14 for only the damaged DOFs: DOFs 1, 5, and 8. The training input to the network for the Mexico City training case is also shown in figure 15 for damaged degrees of freedom. The dimensionality of the input vector given in section 5.1 is also maintained.

In figure 16 (left), the 10-DOF system is damaged during overlapping time intervals (the coupled damage scenario) 10% during 80–120 s, 40% during 60–100 s, and 20% during 40–80 s for DOFs 1, 5, and 8, respectively. The error function is capable of detecting and isolating the damage interval as illustrated by the nonzero error functions shown in figure 16 (right). In figure 17 (left), the 10-DOF system is damaged during time intervals which are not overlapping (the uncoupled damage scenario): 10% during 80–100 s, 30% during 40–60 s, and 20% during 100–120 s for DOFs 1, 5, and 8, respectively. Again, the RBFN formulation does produce relatively greater nonzero error functions, shown in figure 17 (right), that corresponds to the aforementioned damage intervals.

6. SAIVS experimental validation

The ability of the SAIVS (semi-active independent variable stiffness) device to vary stiffness in real time creates a unique opportunity for experimental validation of online SHM algorithms. The proposed method is validated using the experimental data involving coupled stiffness variation. Both stiffness degradation as well as stiffness recovery are evaluated. A brief overview of the SAIVS device and the testing procedure follows.

6.1. Overview of SAIVS device

The SAIVS device was developed by Nagarajaiah and Mate [21] and has been studied in numerous applications since 1998. A schematic diagram of the device is shown in figure 18. The device can smoothly vary stiffness in the x direction due to its rhombus configuration of springs. To prevent buckling of the springs their motion is guided by two cross bars, one connecting joint 3 and joint 4 to allow translation in the x-direction, and the other connecting joint 1 and joint 2 to allow translation in the y-direction. The device has two limit states controlled by the angle, θ, of the rhombus shown in the figure 18 schematic. If the x-direction of the device is placed parallel to the axis of motion, when $\theta = 90°$ the rhombus produces the weakest limit state of least combined spring stiffness. However, when $\theta = 0°$ the rhombus produces the strongest limit state of maximum combined stiffness. By varying θ between these two limit states, stiffness is varied in real time.

Recently, the SAIVS device has been employed in several applications involving seismic disturbance rejection. It has been used by Nagarajaiah and Varadarajan to develop a semi-active tuned mass damper [28]. Also, Narasimhan and Nagarajaiah have successfully coupled the device with a base isolation system in buildings and shown it successful in reducing earthquake induced vibrations [25]. Finally, Nagarajaiah and Saharabudhe have verified a new smart sliding structure that implements the SAIVS device both numerically and experimentally [22]. The SAIVS device application in this study extends the parameter varying capability to structural health monitoring.

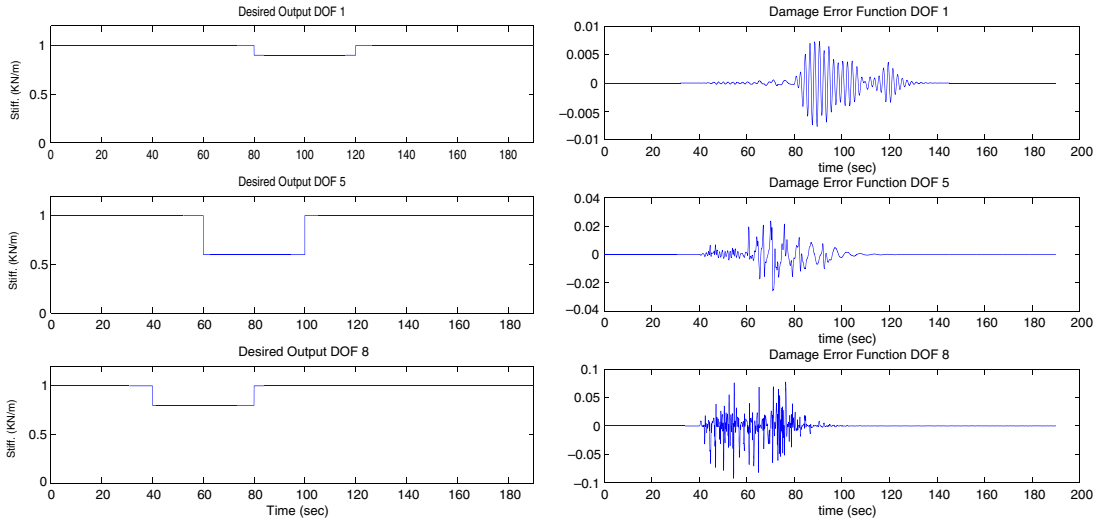

Figure 16. Coupled stiffness change (left) and error function (equation (14)) detection (right) for Mexico City case.

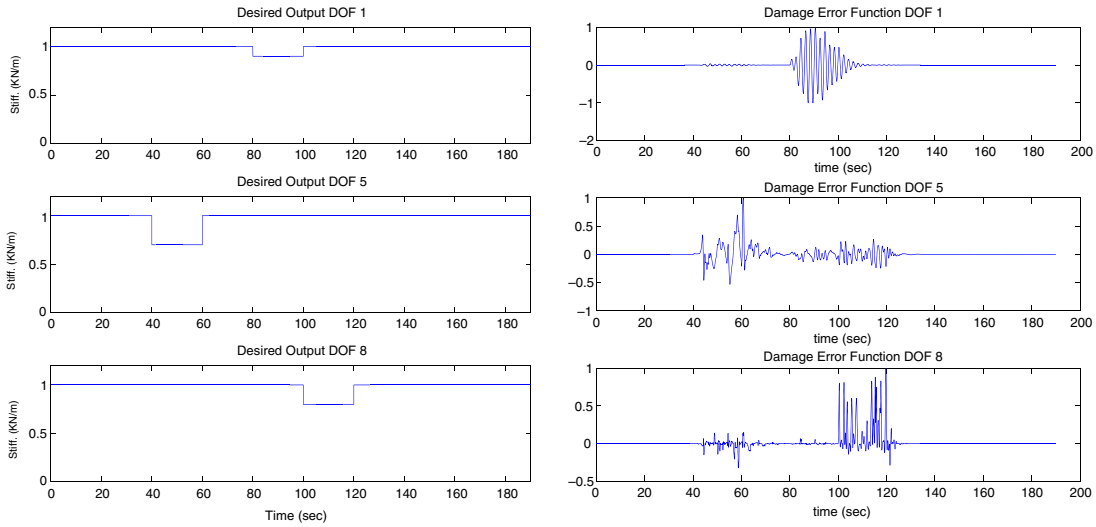

Figure 17. Uncoupled stiffness change (left) and error function (equation (14)) detection (right) for Mexico City case.

6.2. Setup and stiffness variation

One of the most attractive elements of the proposed formulation is its ability to detect coupled parameter variation. The 2-DOF experiment conducted at Rice University by Dharap [8] performs coupled parameter variation and records displacement (using a linear variable differential transformer, LVDT) in each degree of freedom. The values for mass, m_1 and m_2 are 242.13 kg and 9.77 kg, respectively. The initial values for stiffness, $k_1(t_0)$ and $k_2(t_0)$, are considered to be healthy values of 15 kN m^{-1} and 3.7 kN m^{-1}, respectively. A diagonal mass matrix was formed and the traditional tridiagonal stiffness matrix was created using the aforementioned values. Rayleigh damping was used to model the experimental damping and approximated by the relationship, $\mathbf{C}^{\text{damp}} = 0.3\mathbf{M} + 0.025\mathbf{K}$. By using the SAIVS device in each degree of freedom to switch between limit states, the stiffnesses are varied in real time according to table 4 and figure 23 (left).

Table 4. Real time stiffness variation for experimental setup.

	0–15 s	15–35 s	35–55 s	55–75 s
$k_1(t)$ (kN m^{-1})	15	7.8	15	15
$k_2(t)$ (kN m^{-1})	3.7	5.8	5.8	3.7

6.3. Results and discussion

The analytical model of the experimental setup is described by equations (34) for $n = 2$. The stiffness output is shown in figure 19 (second and third from top), and the desired input training data shown in figure 20 is used to train the network. Since the testing harmonic excitation is known *a priori*, site specific response spectra is not needed as in the case of earthquake input. The input used to create the training displacement data is a 1.5 Hz sinusoidal input and is also shown in figure 19 (top). The sinusoidal input experienced during testing, shown figure 22 (top), contained noise and was

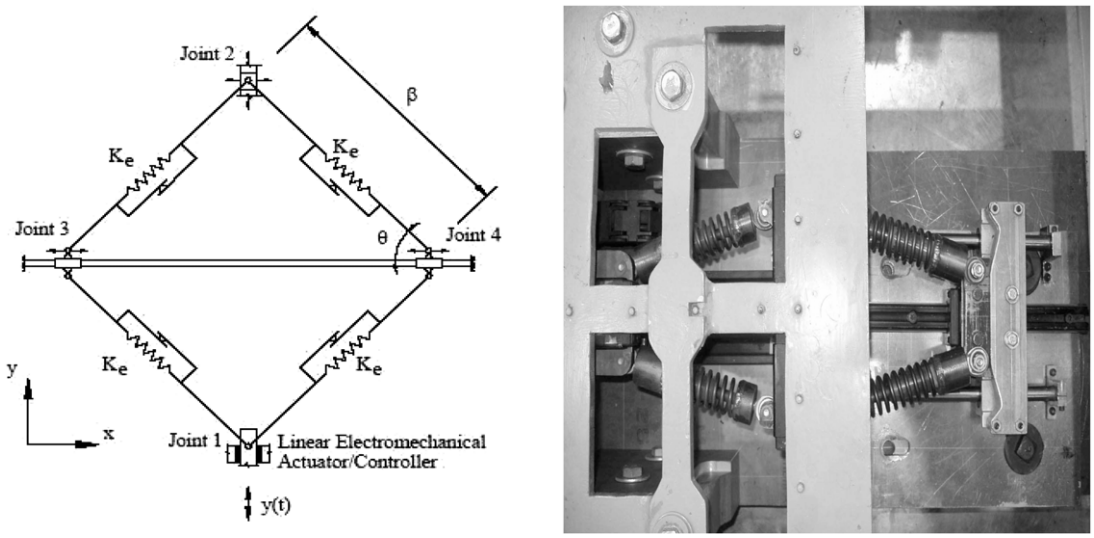

Figure 18. SAIVS device design and implementation.

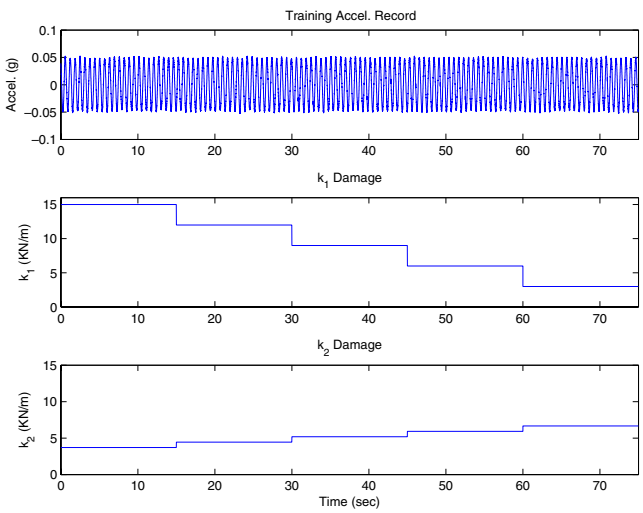

Figure 19. Training Acc. and desired stiffness output for experimental validation.

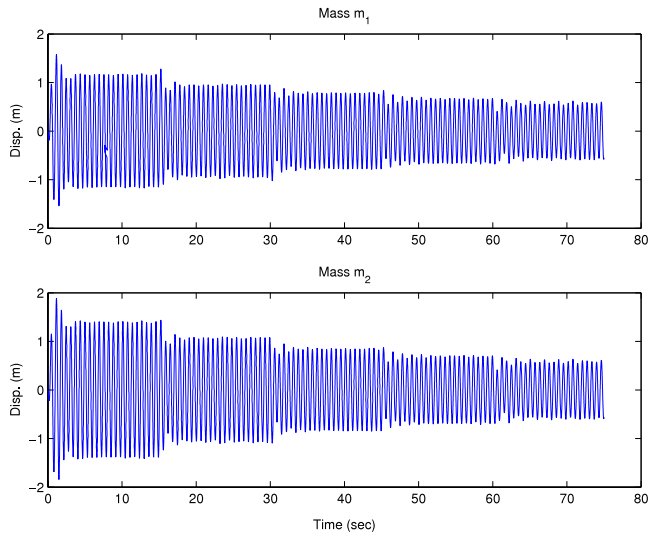

Figure 20. Network training input produced by Acc. in figure 19 (top).

composed of several frequencies with the most dominant being near 1.5 Hz. The dimension of the training input vector is chosen to be in $\Re^{101}$ so that it is of sufficient size for both training and testing.

The effects of the network adding and pruning neurons is shown in figure 21 during training. The number of hidden layer neurons oscillates in the initial learning iterations. As training continues, the number of hidden neurons ceases to fluctuate. For display purposes, the hidden neurons are plotted for every instant of observation. Training was minimized due to the computational efficiency of the network and the training input being monotonic. The experimental data from the coupled damage test is shown in figure 22.

Illustrated by a comparison of figures 23 (left) and (right), the damage error function accurately detects and isolates the stiffness variation. Also, there are some instances where a small magnitude error residual exists when it should strictly be zero. The inability to identify a perfect system model and measurement noise contribute to the magnitude of this error. As with most formulations, to ensure robust FDI it is important that this baseline deviation is small relative to the error function.

It should also be noted, that this error residual in figure 23 is unique with respect to the previous validations in this study because its magnitude is primarily negative. It has been shown in section 4.2 that the sign of the error function is somewhat arbitrary as long as $r_j \rightarrow 0$, for $\mathbf{F}_m \rightarrow \mathbf{0}$. It should also be noted that the magnitude of r_j is scaled by the weights, α_j. Proper scaling of α_j may pose a possible extension to future fault quantification applications.

In summary, assuming a system can be expressed in the form of equations (4), (29) and (33) can be fulfilled, the formulation proposed in this study is robust. This would include some systems with non-Rayleigh damping as well as systems with low levels of noise. These statements are

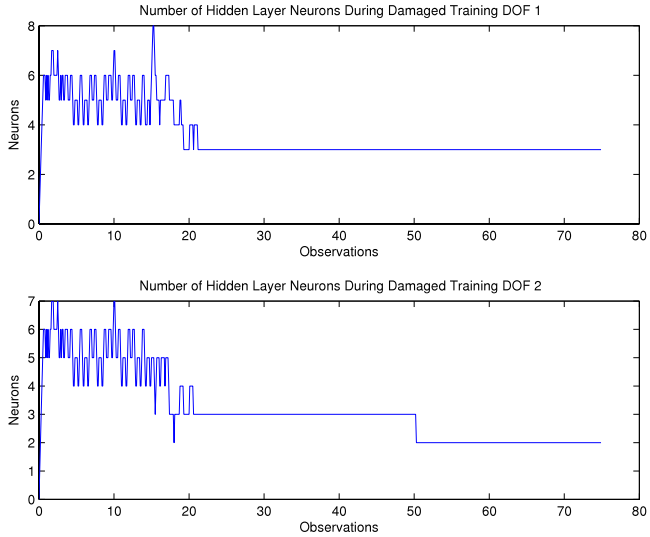

Figure 21. Neuron growth and pruning during training.

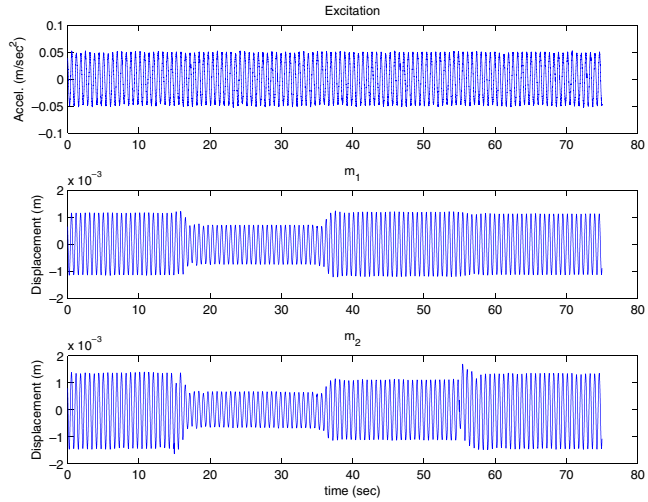

Figure 22. Testing excitation and response displacements of mass m_1 and m_2.

supported by the results for the experimental system in which the damping in actuality was not perfectly Rayleigh, and there was noise present in the data. Regarding modeling error in the healthy observer (i.e. equation (14) having a value near zero for a no damage case), the more accurate the representation of the healthy model, the more clearly the formulation displays the onset of time varying faults. For the simulations and experiment in this study, desirable results were achieved with a straightforward mathematical model, but this may not be the case for highly nonlinear structures with poor analytical representation.

7. Conclusion

A novel health monitoring formulation capable of isolating coupled and uncoupled stiffness variation, damage and recovery, has been proposed. It was also illustrated that for the generalized training procedure described in this study, coupled and uncoupled damage cases could be sufficiently detected and isolated. Some of the merits of the formulation include real time detection and generalized offline training. The algorithm has been validated for systems experiencing damage during earthquake excitations, larger systems with non-collocated input and output with damage in only a few members, and experimental data from stiffness variation tests in the presence of harmonic excitation performed using the SAIVS device at Rice University. The developed method has significant potential as evident from the numerical and experimental validations.

Acknowledgments

The first author gratefully acknowledges the National Science Foundation for its support through the Alliance for

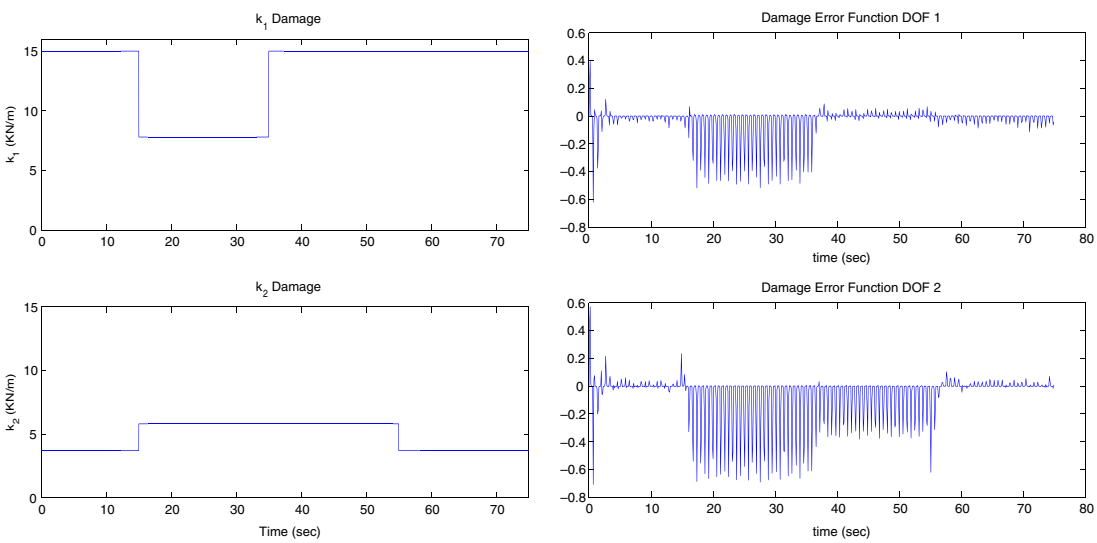

Figure 23. Coupled stiffness change (left) and error function (equation (14)) detection (right).

Graduate Education and the Professoriate fellowship program, cooperative agreement no. HRD-0450363.

References

[1] Beard R V 1971 Failure accommodation in linear system through self reorganization *PhD Thesis* Massachusetts Institute of Technology

[2] Chen B and Nagarajaiah S 2008 Structural damage detection using decentralized controller design method *Smart Struct. Syst.* **4** 779–94

[3] Chen B and Nagarajaiah S 2008 H_-/H_∞ structural damage detection filter design using iterative LMI approach *Smart Mater. Struct.* **17** 035019

[4] Chen B and Nagarajaiah S 2007 Linear matrix inequality based robust fault detection and isolation using the eigenstructure assignment method *J. Guid. Control Dyn.* **30** 1831–5

[5] Contreras M, Nagarajaiah S and Narasimhan S 2008 Fault tolerant neural aided controller for multi degree of freedom structures experiencing online sensor failure *Adv. Sci. Technol.* **56** 247–52

[6] Contreras M, Nagarajaiah S and Narasimhan S 2009 Real time damage detection in buildings using filter based radial basis function network mapping *Proc. ASCE Struct. Eng. Instit. (Austin, TX, May 2009)* pp 1–10 (doi:10.1061/41031(341)121)

[7] Cover T M 1965 Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition *IEEE Trans. Electron. Comput.* **14** 326–34

[8] Dharap P 2006 Real-time structural damage detection using interaction matrix formulation and observers *PhD Thesis* Department of Civil and Environmental Engineering, Rice University, Houston, TX

[9] Dharap P, Koh B H and Nagarajaiah S 2006 Structural health monitoring using ARMarkov observers *J. Intell. Mater. Syst. Struct.* **17** 469–81

[10] Douglas R K 1993 Robust fault detection filter design *PhD Dissertation* The University of Texas at Austin

[11] Frank P M 1990 Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—a survey and some new results *Automatica* **26** 459–74

[12] Holland H 1975 *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI: University of Michigan Press)

[13] Kadirkamanathan V and Niranjan M 1993 A function estimation approach to sequential learning with neural network *Neural Comput.* **5** 954–75

[14] Koh B H, Li Z, Dharap P, Nagarajaiah S and Phan M Q 2005 Actuator failure detection through interaction matrix formulation *J. Guid. Control Dyn.* **28** 895–901

[15] Koh B H, Dharap P, Nagarajaiah S and Phan M Q 2005 Real-time structural damage monitoring by input error function *AIAA* **43** 1808–14

[16] 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. Inst. Electr. Eng. Control Theory Appl.* **147** 476–84

[17] Li Z, Koh B H and Nagarajaiah S 2007 Detecting sensor failure via decoupled error function and inverse input–output model *J. Eng. Mech.* **133** 1222–8

[18] Liberatore S 2002 Analytical redundancy, fault detection and health monitoring for structures *PhD Thesis* University of California, Los Angeles

[19] Massoumnia M A 1986 A geometric approach to failure detection and identification in linear systems *PhD Dissertation* Department of Aeronautics and Astronautics, Massachusetts Institute of Technology

[20] Moody J and Darken C J 1989 Fast learning in networks of locally-tuned processing units *Neural Comput.* **1** 281–94

[21] Nagarajaiah S and Mate D 1998 Semi-active control of continuously variable stiffness system *Proc. Second World Conf. Structural Control* vol 1, pp 397–405

[22] Nagarajaiah S and Saharabudhe S 2006 Seismic response control of smart sliding isolated buildings using variable stiffness systems: an experimental and numerical study *Earthq. Eng. Struct. Dyn.* **35** 177–97

[23] Nagarajaiah S and Li Z 2004 Time segmented least squares identification of base isolated buildings *Soil Dyn. Earthq. Eng.* **24** 577–86

[24] Narasimhan S, Suresh S, Sundararajan S and Nagarajaiah S 2008 On-line learning failure-tolerant neural controller for earthquake excited structures *J. Eng. Mech.* **134** 258–68

[25] Narasimhan S and Nagarajaiah S 2005 A STFT semiactive controller for base isolated buildings with variable stiffness isolation systems *Eng. Struct.* **27** 514–23

[26] Narendra K S 1996 Neural networks for control: theory and practice *Proc. IEEE* **84** 1385–406

[27] Platt J 1991 A resource allocating network for function interpolation *Neural Comput.* **3** 213–25

[28] Nagarajaiah S and Nadathur V 2005 Semi-active control of wind excited building with variable stiffness TMD using short time Fourier transform *J. Eng. Struct.* **27** 431–41

[29] Warwick K 1995 A critique of neural networks for discrete-time linear control *Int. J. Control* **61** 1253–64

[30] White J E 1985 Detection filter design by eigensystem assignment *PhD Thesis* University of Texas, Austin