Journal of Sound and Vibration (2002) **256**(3), 515–532
doi:10.1006/jsvi.5007, available online at http://www.idealibrary.com on IDEAL®

PROPER ORTHOGONAL DECOMPOSITION AND ITS APPLICATIONS – PART II: MODEL REDUCTION FOR MEMS DYNAMICAL ANALYSIS

Y. C. Liang

Department of Computer Science, Jilin University, 10 *Qian Wei Road, Changchun*, 130012, *People's Republic of China. E-mail: ycliang@public.cc.jl.cn*

W. Z. Lin

Institute of High Performance Computing, 89*C Science Park Drive*, #02-11/12, *the Rutherford*, *Singapore* 118261, *Singapore*

H. P. Lee, S. P. Lim and K. H. Lee

Centre for Advanced Computations in Engineering Science (*ACES*), *c/o Department of Mechanical Engineering, National University of Singapore*, 10 *Kent Ridge Crescent*, *Singapore* 119260, *Singapore*

and

H. Sun

Department of Mathematics, Jilin University, 10 *Qian Wei Road, Changchun*, 130012, *People's Republic of China*

(*Received* 11 *July* 2001, *and in final form* 4 *December* 2002)

Proper orthogonal decomposition (POD) methods are popular tools for data analysis aimed at obtaining low-dimensional approximate descriptions of a high-dimensional process in many engineering fields. The applications of POD methods to model reduction for microelectromechanical systems (MEMS) are reviewed in this paper. In view of the fact that existing POD methods in the model reduction for dynamic simulation of MEMS dealt with only noise-free data, this paper proposes a neural-network-based method that combines robust principal component analysis (PCA) neural network model with Galerkin procedure for dynamic simulation and analysis of non-linear MEMS with noisy data. Simulations are given to show the performance of the proposed method in comparison with the existing method. Compared with the standard PCA neural network model, the robust PCA neural network model has a number of numerical advantages such as the stability and robustness to noise-injected data and the faster convergence of iterations in the training stages than the existing neural network technique. The macro-model generated by using the eigenvectors extracted from the proposed method as basis functions shows its flexibility and efficiency in the representation and simulation of the original non-linear partial differential equations.

1. INTRODUCTION

Proper orthogonal decomposition (POD) is an important and essential technique for data reduction, image compression, and feature extraction. It has been widely used in various disciplines including random variable, image processing, signal analysis, data compression, and process identification, etc. The widespread applications of POD methods enable the

0022-460X/02/$35.00

POD to be a popular tool in many fields. For example, POD methods have been successfully used in structural vibrations [1–7]. A summary of the equivalence of the three POD methods, the Karhunen–Loève decomposition (KLD), the principal component analysis (PCA), and the singular value decomposition (SVD), has been made and the connections among the three methods have been demonstrated in reference [8]. In this paper, applications of the three POD methods to the microelectromechanical systems (MEMS) model reduction are reviewed first, then a neural-network-based method for model reduction that combines robust generalized Hebbian algorithm (RGHA) with Galerkin procedure to perform the dynamic simulation and analysis of non-linear MEMS is proposed and examined. Comparative experiments are made and the results show that the proposed method improves the performances of the existing PCA algorithm based on neural networks when noises are present.

The development of increasingly complex MEMS demands sophisticated simulation techniques for design and optimization [9]. This simulation and modelling are usually presented with non-linear partial differential equations (PDE) because MEMS devices typically involve multiple coupled energy domains and media and there exist inherent non-linearities of electrostatic actuation forces and geometric non-linearities caused by large deformation. There are techniques such as finite element methods (FEM) or finite difference methods (FDM) to convert continuous dynamic non-linear systems with infinite number of degrees of freedom to discrete finite dimensional models. But the resulting number of degrees of freedom is usually too large so that it is extremely computationally intensive and time consuming for practical problems. Therefore, a major current goal of simulation and modelling research is to develop efficient methods of creating accurate low-order dynamic models that capture most of the accuracy and flexibility of the original PDE, or of the fully meshed dynamic FEM or FDM model [9, 10].

In recent years, several approaches of model reduction for dynamic simulations of MEMS have been presented [9–20], including lumped-parameter techniques, linear modal analysis techniques and Arnoldi-based model reduction approaches, etc., which are reviewed in references [10, 11]. More recently, three POD methods including the singular value decomposition (SVD), the Karhunen–Loève decomposition (KLD), and the principal component analysis (PCA) using neural networks have been proposed to handle the model reduction for MEMS respectively [10–13]. In these four papers, the authors demonstrate how efficiently the reduced-order dynamical models for micromechanical devices can be constructed using data from a few runs of fully meshed numerical models such as those created by the FEM or FDM, and how these low-order macro-models are generated by extracting global basis functions from the fully meshed model runs in order to parameterize solutions with far fewer degrees of freedom. Among the three approaches the first two methods need some matrix computations in advance, such as the computation on the input correlation matrix in the KLD. A matrix eigenvalue problem of large scale is required to be solved and all eigenfunctions of the correlation matrix or some equivalent matrix from an ensemble of signals have to be calculated in the SVD or KLD. Compared with these two POD model reduction methods for dynamic simulations of MEMS, the PCA method using neural networks does not need to compute the input correlation matrix in advance; it is necessary to find only very few of the required basis functions. The neural-network-based PCA method therefore possesses potential advantages when the measured data are large. However, the three POD approaches dealt with only noise-free data in the model reduction of MEMS. Since the POD methods have to process information from the real world, it should have the ability to cope with noisy data. In this paper, we extend the neural-network-based method of model reduction for MEMS to handle the noisy data by using robust PCA neural networks. The ability of processing noisy data enables the

proposed robust neural network method to be an ideal choice in the model reduction of MEMS in real applications.

2. THE SYSTEM AND GOVERNING EQUATIONS

In this section, a doubly clamped micro-beam pulled in by the electrostatic actuation force with squeezed gas-film damping effect is examined to demonstrate the model reduction technique and the generation of the macro-model based on the robust PCA neural network.

The cross-section of the micro-beam is shown in Figure 1 [21]. The device consists of a deformable elastic beam microstructure that is electrostatically pulled in by an applied voltage waveform. When a voltage V is applied on the top and bottom electrodes, the top deformable micro-beam is pulled downwards due to the electrostatic force. At the same time, the narrow air gap between the moving micro-beam and the substrate will generate back pressure force on the micro-beam due to squeezed gas-film damping effect [22]. The top micro-beam will reach an unstable point and pull-in onto the 0·5 μm dielectric layer coated on the bottom substrate when the applied voltage attains the pull-in voltage. The applied voltage is sensitive to the ambient pressure of the air thus structure can be used as accelerometer [22] and pressure sensor [21].

The device shown in Figure 1 is a coupled domain system. In general, the micro-beam can be modelled by the Euler beam equation with electrostatic actuation force, and the back pressure force and be modelled by the non-linear Reynold' squeezed gas-film damping equation [23] to become the following PDE:

$$EI\frac{\partial^4 z}{\partial x^4} - T\frac{\partial^2 z}{\partial x^2} = -\frac{\varepsilon_0 \mathbf{B}\mathbf{V}^2}{2z^2} + \int_0^B (p - p_a)\,\mathrm{d}y - \rho\frac{\partial^2 z}{\partial t^2}, \tag{1}$$

$$\nabla\cdot(z^3 p\,\nabla p) = \frac{12\eta}{1+6K}\frac{\partial(pz)}{\partial t}, \tag{2}$$

where E is the elastic modulus, $I = Bh^3/12$ is the moment of inertia where B is the width of the micro-beam and h is the thickness, T is the residual stress, $K(x,t) = \lambda/z$ is the Knudson number where λ is the mean-free path of the air and is equal to 0·064 μm, $z(x,t)$ is the height of the micro-beam above the substrate, $-\varepsilon_0 BV^2/(2z^2)$ is the electrostatic actuation force where V is the applied voltage, ε_0 is the permittivity of free space and is equal to $8{\cdot}854\times10^{-12}$ F/m, $p(x,y,t)$ is the back pressure force caused by the squeezed gas-film where an isothermal process is assumed, p_a is the ambient pressure and is equal to $1{\cdot}013\times10^5$ Pa, ρ is the density and η is the air viscosity and is equal to $1{\cdot}82\times10^{-5}$ kg/(m s).

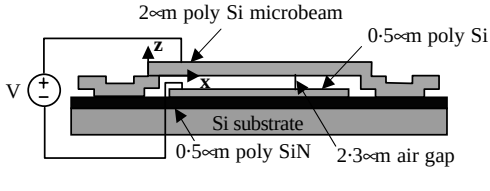

Figure 1. Doubly clamped micro-beam.

3. MODEL REDUCTION BASED ON THE ROBUST PCA NEURAL NETWORK

Traditional FEM or FDM can be used for explicit dynamical simulations of PDE. However, computational prototyping using full models to simulate non-linear PDE are usually computationally very intensive and time consuming, making them difficult to use when a large number of simulations are needed. It has been demonstrated that a Galerkin procedure employing the eigenvectors obtained from the GHA neural network can convert the dynamic non-linear system to a model with a small number of degrees of freedom, while capturing most of the accuracy and flexibility of the original system efficiently [11]. The principal components are the most important linear features of the random observation vectors. The purpose of the PCA is to identify the dependent structure behind a multivariate stochastic observation in order to obtain a compact description of it. Through the PCA many variables can be represented by a few principal components, so the PCA can be considered as a feature extraction technique. Performing the PCA on a set of multivariate random data means computing the eigenvectors of its correlation matrix corresponding to the largest eigenvalues, and the projection of the data over the eigenvectors to obtain a number of principal components.

Since the pioneering work of Oja [24] on extracting the first principal component by a linear neuron model, the issue of neural learning PCA has generated great interest recently. A number of unsupervised learning algorithms for extracting multiple principal components or their subspace have been proposed, usually developed from the variance maximization or the mean-square error minimization [25–31]. Unlike the traditional statistical eigenvector analysis algorithms, these neural-network-based approaches do not require the computation of the input data covariance, which may increase significantly with the dimensionality of the training data. Furthermore, there is no need to evaluate all the eigenvalues and eigenvectors if only the eigenvectors corresponding to the first several significant eigenvalues are required. In this paper we use the robust PCA algorithm proposed in reference [31] to extract the principal eigenvectors of the correlation matrix from an ensemble of signals. The algorithm is derived from the representation error minimization, which is given by

$$
\begin{aligned}
w_{k+1}(i) &= w_k(i) + \mu_k y_k(i) f(e_k(i)) \quad (i = 1, 2, \ldots, M), \\
e_k(i) &= x_k - \sum_{j=1}^{I(i)} y_k(j) w_k(j) \quad (i = 1, 2, \ldots, M), \\
y_k(i) &= x_k^{\mathrm{T}} w_k(i) \quad (i = 1, 2, \ldots, M),
\end{aligned}
\tag{3}
$$

where $w_k(i)$ is the weight vector of the ith neuron, $e_k(i)$ is the instantaneous representation error vector, $y_k(i)$ is the output of the ith neuron, μ_k is the gain parameter, M is the number of the neurons in the output layer of the network, x_k is the input data vector, and $f(\cdot)$ is a non-linear function. In general, the non-linear function $f(t)$ adding to the error term in equation (3) should subject to the following requirements: $f(t)$ is a monotonically growing function of t. For stability reasons, it is necessary to assume that $f(t) \leqslant 0$ for $t < 0$ and $f(t) \geqslant 0$ for $t > 0$, i.e., it is required that the growing of $f(t)$ should be less than the linear growing. The upper bound of the summation index $I(i)$ represents the two different cases of the network models. In the standard symmetric case, $I(i) = M$, for all $i = 1, 2, \ldots, M$. In the standard hierarchic case $I(i) = i$, then the optimal weight vector of the ith neuron defines the robust counterpart of the ith principal eigenvector. In the standard hierarchic case and

linear special case $f(t) = t$, equation (3) coincides exactly with the well-known generalized Hebbian algorithm (GHA) proposed originally by Sanger [32],

$$w_{k+1}(i) = w_k(i) + \mu_k \left[y_k(i)x_k - y_k(i) \sum_{j \leqslant i} y_k(j) w_k(j) \right] \quad (i = 1, 2, \ldots, M). \tag{4}$$

Equation (3) defines a generalization of the GHA algorithm. Considering that the algorithm using equation (3) possesses the robustness for noise-injected data through the numerical simulation, we call equation (3) the robust GHA (RGHA).

Robustness theory is concerned with solving problems subject to model perturbation or added noise. A robust algorithm could not only perform well under the assumed model, but also produce a satisfactory result under the deviation of the assumed model. That is the reason for choosing the RGHA to deal with the model reduction of MEMS with noisy data in this paper.

In the present model reduction algorithm we use equation (3) of the RGHA to obtain the eigenvectors (principal components) by iteratively training the neural network, where the input vector x_k is the snapshot described in the next section and the weight vectors $w_k(i)$ is the eigenvector which we seek for. It should be pointed out that in our experience the choice of the gain parameter μ_k in equation (3) has a profound impact in the convergence speed of the RGHA. In general μ_k should decrease with time such that

$$\lim_{k \to \infty} \mu_k = 0 \quad \text{and} \quad \sum_{k=0}^{\infty} \mu_k = \infty. \tag{5}$$

In order to compare the simulation results in this paper with those obtained from reference [11] using the GHA model we adopt the adaptive choice for μ_k employed in reference [11], where μ_k can be calculated iteratively by

$$\mu_k = \frac{\mu_{k-1}}{\gamma + y_k^2 \mu_{k-1}}, \tag{6}$$

where $0 < \gamma \leqslant 1$ is a factor chosen by the user. Simulation results show that good convergence can be obtained if γ is chosen to be closer to 1.

4. GENERATING SNAPSHOTS AND THE MACRO-MODEL

We now describe how to obtain the ensemble of signals or snapshots from the numerical solution of an original non-linear dynamic system. Firstly, the time-dependent deflection $z(x, t)$ and pressure $p(x, y, t)$ in equations (1) and (2) are simulated using the FDM technique. For the system shown in Figure 1, the pull-in dynamics of the micro-beam at a series of different time are simulated using FDM for an ensemble of applied step voltage to obtain the beam deflection $z_n(x_i, t_s)$ and the back air pressure $p_n(x_i, y_j, t_s)$ ensemble. These deflection and back pressure ensemble are then used as snapshots, i.e., the ensemble of signals for the GHA network to generate the eigenvectors. The ensemble of applied step voltage is taken to be that of the operating range of the systems.

In order to simulate the system shown in Figure 1 using FDM, we discretize the Euler beam equation (1) and Reynold equation (2) in space to generate an $(M + 1) \times (N + 1)$ mesh

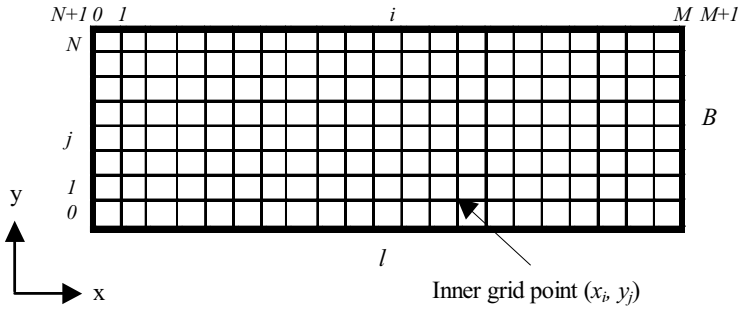

Figure 2. Finite difference mesh of the micro-beam.

with $M \times N$ inner grids and $2M + 2N + 4$ boundary grids as shown in Figure 2. The central difference is used to discretize the spatial partial derivative operators in equations (1) and (2) and the trapezoidal rule is adopted to discretize the integral operator. The state of three unknowns $z(x,t)$, $\partial z(x,t)/\partial t$ and $p(x,y,t)$ is projected onto each grid point. This discretization will transform equations (1) and (2) into a set of $M \times N + 2M$ non-linear ODE, and we can use the following state space to represent the unknowns on the grids:

$$\dot{\mathbf{x}} = \left(\frac{\partial z_1}{\partial t} \cdots \frac{\partial z_M}{\partial t} \; \frac{\partial^2 z_1}{\partial t^2} \cdots \frac{\partial^2 z_M}{\partial t^2} \; \frac{\partial p_{11}}{\partial t} \cdots \frac{\partial p_{MN}}{\partial t} \right)^{\mathrm{T}} \tag{7}$$

and they are integrated numerically by using the fifth-order Runge–Kutta method with the following boundary conditions:

$$z = z_0, \qquad \frac{\partial z}{\partial x} = 0, \qquad \frac{\partial p}{\partial n} = 0 \quad (\text{at } x = 0, l),$$

$$p = p_a \quad (\text{at } y = 0, B) \tag{8}$$

and the initial conditions:

$$z = z_0, \qquad \frac{\partial z}{\partial t} = 0, \qquad p = p_a \quad (\text{at } t = 0). \tag{9}$$

The snapshots can be taken at varied or fixed time intervals during pull-in dynamics. Since there is no distinct difference between transient and steady state for the system shown in Figure 1, we take the snapshots at fixed time intervals in this paper. Using the above-mentioned snapshots as inputs to the RGHA neural network, we can obtain the eigenvectors of the input correlation matrix. It is noted that the inputs are centralized to become mean-value free signals before the principal component is performed [8] and the fact that the true eigenvectors are distorted by inclusion of the mean value is also reported in reference [33].

Next, the Galerkin procedure which employs these eigenvectors as basis functions is applied to the original non-linear governing PDE (1) and (2) to convert it to a macro-model with a small number of ordinary differential equations (ODE).

Considering that the independent deflection and pressure basis functions make the Galerkin derivation simpler and also make sense the physics of the problem, we perform the

principal component extraction using the RGHA corresponding to the deflection and pressure respectively. Denoting the eigenvectors with respect to the deflection as $\phi_i^z(x)$ and those with respect to the pressure as $\phi_j^p(x, y)$, we can represent the deflection $z(x, t)$ and pressure $p(x, y, t)$ as a linear combination of the eigenvectors as follows:

$$z(x,t) = z_0 + \sum_{i=1}^{I} a_i^z(t)\,\phi_i^z(x), \tag{10}$$

$$p(x,y,t) = p_a + \sum_{j=1}^{J} a_j^p(t)\,\phi_j^p(x,y), \tag{11}$$

where z_0 is the initial gap between the deformable micro-beam and the substrate, p_a is the gap air ambient pressure, I and J are the numbers of basis vectors for the deflection and back pressure respectively. Substituting equations (10) and (11) into equations (1) and (2) and applying the Galerkin procedure, we have

$$M_j \frac{\mathrm{d}^2 a_j^z}{\mathrm{d}t^2} + \sum_{i=1}^{I} K_{ji} a_i^z + f_j = 0 \quad (j = 1,2,\ldots,I), \tag{12}$$

$$\sum_{i=1}^{J} H_{ji} \frac{\mathrm{d}a_i^p}{\mathrm{d}t} + \sum_{i=1}^{J} S_{ji} a_i^p + c_j = 0 \quad (j = 1,2,\ldots,J), \tag{13}$$

where

$$M_j = \int_L \rho(\phi_j^z)^2\,\mathrm{d}x \tag{14}$$

$$K_{ji} = K_{ij} = \int_L \left(EI \frac{\partial^2 \phi_j^z}{\partial x^2} \frac{\partial^2 \phi_i^z}{\partial x^2} + T \frac{\partial \phi_j^z}{\partial x} \frac{\partial \phi_i^z}{\partial x} \right) \mathrm{d}x, \tag{15}$$

$$f_j = \int_L \left(\frac{\varepsilon_0 B V^2}{2z^2} - \int_0^B (p - p_a)\,dy \right) \phi_j^z\,\mathrm{d}x \tag{16}$$

and

$$H_{ji} = H_{ij} = \int_A \frac{12\eta}{1 + 6K} z \phi_j^p \phi_i^p\,\mathrm{d}x\,\mathrm{d}y. \tag{17}$$

$$S_{ji} = S_{ij} = \int_A \left\{ z^3 p \left(\frac{\partial \phi_j^p}{\partial x} \frac{\partial \phi_i^p}{\partial x} + \frac{\partial \phi_j^p}{\partial y} \frac{\partial \phi_i^p}{\partial y} \right) + \frac{12\eta}{1 + 6K} \phi_j^p \phi_i^p \frac{\partial z}{\partial t} \right\} \mathrm{d}x\,\mathrm{d}y, \tag{18}$$

$$c_j = \int_A \frac{12\eta}{1 + 6K} p_a \phi_j^p \frac{\partial z}{\partial t}\,\mathrm{d}x\,\mathrm{d}y, \tag{19}$$

where $\int_L$ indicates the integration along the length of the micro-beam and $\int_A$ indicates the integration over the area of the micro-beam.

The above small set of coupled ODE (12) and (13) constitutes the macro-model with global basis functions, which is the low-order dynamic simulation of the original non-linear

PDE system, equations (1) and (2). Since this dynamic macro-model of ODE is generated by Galerkin procedure employing the eigenvectors extracted from the RGHA network, the resulting degrees of freedom is usually small. It is very efficient to simulate the system compared with the full model of FEM or FDM which contains large degrees of freedom. The set of ODE (12) and (13) is integrated numerically in time by a fifth-order Runge–Kutta method to simulate the dynamics of the system. The initial values for the system are as follows:

$$a_j^z|_{t=0} = \frac{\int_L (z - z_0)|_{t=0}\, \phi_j^z \, \mathrm{d}x}{\int_L (\phi_j^z)^2 \, \mathrm{d}x}, \tag{20}$$

$$a_j^p|_{t=0} = \frac{\int_A (p - p_a)|_{t=0}\, \phi_j^p \, \mathrm{d}x\, \mathrm{d}y}{\int_A (\phi_j^p)^2 \, \mathrm{d}x\, \mathrm{d}y}. \tag{21}$$

5. NUMERICAL RESULTS

We now present simulation results on the MEMS device shown in Figure 1 to demonstrate the efficiency and accuracy of the present model reduction technique using the RGHA neural network. The features and dimension of the micro-beam are given in Table 1. The snapshots generated for pressure and displacement based on runs of the FDM code for an ensemble of two different step voltages at $V_1 = 10$ V and $V_2 = 16$ V which are assumed to be the device operating range under consideration. Based on numerical experiments, a mesh size of 40×20 for the finite difference simulation of the original non-linear equations (1) and (2) is able to generate sufficient accuracy. The minimum step pull-in voltage for this device is calculated at 8·87 V by the FDM code, which is matched to the experimental data measured at 8·76 V [34]. Each 25 snapshots are taken at the fixed time interval from the moment when each step voltage is applied till the pull-in happened. Then, in general, these snapshots are used to generate the eigenvectors from the application of the POD methods. The validity and suitability of the eigenvectors obtained using the GHA model as proper shape functions have been demonstrated in reference [11]. The eigenvectors given by the GHA and those obtained by the KLD are examined and compared. The two sets of eigenvectors from these two different methods are quite similar. The first two order eigenvectors corresponding to the deflection are plotted in Figure 3. The other eigenvectors corresponding to the deflection and pressure also possess such similarities. The Galerkin

TABLE 1

Values of the features and geometric dimension for the micro-beam

Elastic modulus E	Residual stress $[T/(hB)]$	Beam's density $[\rho/(hB)]$	Knudsen's number $K = \lambda/z$	Length l	Width B	Thickness h	Initial gap z_0
149 GPa	− 3·7 GPa	2330 kg/m^3	≈ 0·028	610 μm	40 μm	2·2 μm	2·3 μm

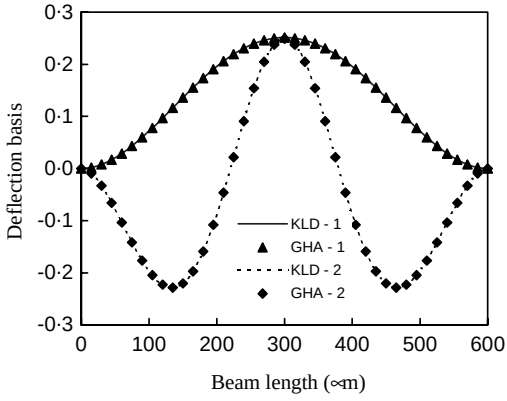

Figure 3. Comparison of the first two eigenvectors from KLD and GHA for noise-free snapshots.

procedure uses these eigenvectors as the basis functions to generate the macro-model to represent and simulate the pull-in dynamics.

In references [11, 13] it has been shown that the macro-model can be represented well when only one deflection basis vector and four back pressure basis vectors are chosen in the simulations. It has been demonstrated that the macro-model is very flexible and efficient to simulate the system without regeneration of the macro-model even when the input voltage wave spectrum is changed and the input voltages are far from the voltages used to create the basis functions [11, 13]. However, all these results are obtained based on the assumption that the data used to generate the eigenvectors have not been spoiled by noise. Currently, little attention has been paid to this problem in MEMS model reduction literatures, although the problem is essentially important for real applications. In practice, real data often contain some noise, and usually it is not easy to separate the noise from the data set. As will be shown by the experiments given in this paper, the noise significantly deteriorates the performances of the existing PCA algorithms.

One of the main purposes of this paper is to examine the influence of the measured noise upon the data processing techniques based on the PCA neural network methods by analyzing the effect of the noise on the eigenvectors obtained using the GHA and RGHA models. Figure 4 shows the comparisons of the first two order eigenvectors corresponding to the deflection, where GHA-1 and GHA-2 represent the first and second order eigenvectors obtained by using the GHA model based on the noise-free data, and GHA-N1 and GHA-N2 represent the first and second order eigenvectors obtained using the GHA model based on the noise-injected data with a noise level of 0·005. The deterioration caused by the noise can be observed from the figure. For example, it can be seen that the deviation from the true values and the destruction of the symmetry of the mode shape in the second eigenvector obtained based on the noise-injected data are obvious. Since the PCA algorithms have to process information from real world, they should have the ability to cope with the noisy data or have the robustness when the noise exists. How to decrease the influence of the noise upon the feature extraction by choosing a suitable data processing technique is an important problem in many cases.

This paper would examine the robustness of the RGHA approach to the noise. Comparative experiments are conducted between the two methods of GHA and RGHA. The noise is added to the snapshots obtained using the FDM. The noise array with uniform distribution is first scaled between the range of $[-1, 1]$. Considering the variation

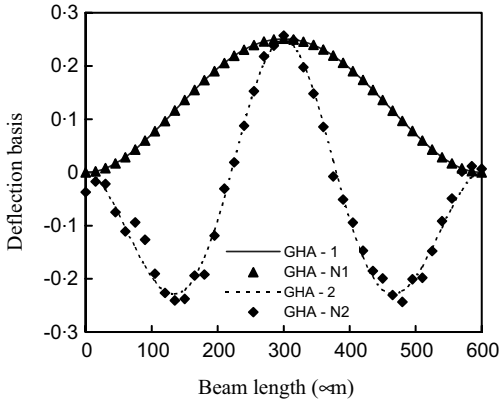

Figure 4. Comparison of the first two eigenvectors from GHA for noise-free and noise-injected snapshots.

magnitude of the displacement of the micro-beam is among 10^{-1}, 10^{-2} and 10^{-3}, the noise scale is controlled within 0 to the magnitude of 10^{-3} in this paper such that the noise-injected displacement data do not have serious distortion to the original data.

In order to compare the results obtained from the noise-free data and the noise-injected data, we define the square error function

$$E = \begin{cases} \sum_{i=1}^{q} (N_i - S_i)^2 \Big/ \sum_{k=1}^{q} S_k^2 & (N_j S_j > 0), \\ \sum_{i=1}^{q} (-N_i - S_i)^2 \Big/ \sum_{k=1}^{q} S_k^2 & (N_j S_j < 0), \end{cases} \tag{22}$$

as a criterion for the comparison, where q is the number of the components of the eigenvector, N_i is the component of the eigenvector obtained using the neural network method based on noise-injected snapshots, S_i is the component of the eigenvector obtained using the KLD based on noise-free snapshots, and N_j and S_j are two selected typical values from the above N_i and $S_i (i = 1, 2, \ldots, q)$ respectively. In the above definition it is considered that the eigenvectors obtained using the two different methods may have different signs.

The comparative experiments are implemented using the following two different ways:

1. *Comparisons using different non-linear functions.* In order to examine the effectiveness of using the non-linear function in equation (3) to noise-injected data, besides the sigmoid function which is used frequently in a number of neural network models, several other non-linear functions listed in reference [35] are also used in the comparative experiments. The non-linear functions employed in this paper are as follows:

(1) $f_1(t) = \dfrac{1 - \mathrm{e}^{-\alpha t}}{1 + \mathrm{e}^{-\alpha t}}$,

(2) $f_2(t) = \begin{cases} t, & |t| \leqslant \beta, \\ \beta \operatorname{sgn}(t), & |t| > \beta, \end{cases}$

(3) $f_3(t) = \begin{cases} t, & |t| \leqslant \beta, \\ 0, & |t| > \beta, \end{cases}$

$$(4)\quad f_4(t) = \begin{cases} \beta \sin\left(\dfrac{t}{\beta}\right), & |t| \leqslant \pi\beta, \\ 0, & |t| > \pi\beta, \end{cases}$$

$$(5)\quad f_5(t) = \operatorname{sgn}(t),$$

$$(6)\quad f_6(t) = \log(1 + 5t),$$

where $f_1(t)$ is the sigmoid function, α is a real coefficient, and β is a parameter its value taken as 1 in the simulations.

The square errors defined in equation (22) for the first eigenvectors obtained using the GHA and the RGHA employing different non-linear functions are calculated according to different noise levels of the noise-injected deflection snapshots. Considering the randomness of the noise-injected data, we calculate 10 times for each non-linear function and each noise level and then take the statistic mean value as the results for the comparison. The comparisons of the square errors obtained using the GHA with those obtained using the RGHA employing different non-linear functions versus noise are shown in Table 2, where RGHA-i represents that the non-linear function f_i ($i = 1, 2, \ldots, 6$) is used in the RGHA model. In the simulations 25 000 iteration steps are used in the training of the GHA and RGHA neural networks, respectively, and the coefficient α in the sigmoid function is taken as 1·5. The comparative experiments show that the same results can be obtained by using the GHA and RGHA models based on the noise-free data, which are not listed in the table. However, the simulations for the noise-injected data are different. From the table it can be seen the results using the RGHA employing the sigmoid function are better than those using the GHA, whereas the results using the RGHA employing other non-linear functions are not better than those using the GHA, sometimes the results are even worse than those using the GHA. From these simulation results some conclusions can be made. In general, it is not true that the RGHA approach possesses the robustness to the noise for all kinds of non-linear functions in equation (3). However, the RGHA algorithm using the sigmoid function has the ability of decreasing the influence of the injected noise.

2. *Comparisons using sigmoid function.* Because of the robustness of the RGHA using the sigmoid function to the injected noise, we use only the sigmoid function in the RGHA model

TABLE 2

Comparison of square errors using GHA with those using RGHA versus noise (%)

Noise	GHA	RGHA-1	RGHA-2	RGHA-3	RGHA-4	RGHA-5	RGHA-6
0·00	3·06E − 14	6·63E − 15	3·06E − 14	3·06E − 14	1·5E − 14	1·5E − 14	1·17E − 14
0·05	8·41E − 09	8·38E − 09	8·00E − 09	8·36E − 09	9·70E − 09	9·70E − 09	8·51E − 09
0·10	3·11E − 08	3·73E − 08	3·26E − 08	3·83E − 08	3·78E − 09	3·78E − 08	3·41E − 08
0·15	7·34E − 08	7·03E − 08	6·47E − 08	8·23E − 08	8·60E − 08	8·60E − 08	9·02E − 08
0.20	1·41E − 07	1·39E − 07	1·51E − 07	1·34E − 07	1·38E − 07	1·38E − 07	1·37E − 07
0·25	2·42E − 07	2·18E − 07	2·24E − 07	1·93E − 07	2·32E − 07	2·32E − 07	2·32E − 07
0·30	3·25E − 07	3·11E − 07	3·19E − 07	3·67E − 07	3·61E − 07	3·61E − 07	3·61E − 07
0·35	4·63E − 07	4·28E − 07	4·31E − 07	5·04E − 07	5·06E − 07	5·06E − 07	4·81E − 07
0·40	5·46E − 07	5·35E − 07	5·64E − 07	5·51E − 07	6·44E − 07	6·44E − 07	6·71E − 07
0·45	7·01E − 07	6·46E − 07	7·40E − 07	7·01E − 07	8·28E − 07	8·28E − 07	6·48E − 07
0·50	8·70E − 07	8·56E − 07	9·22E − 07	1·04E − 06	1·00E − 06	1·00E − 06	9·13E − 07

in the following comparative experiments. We first examine the effectiveness of using the sigmoid function to the noisy data. Numerical experiments for the MEMS model reduction using the GHA neural network method have shown that the steps of the training to the

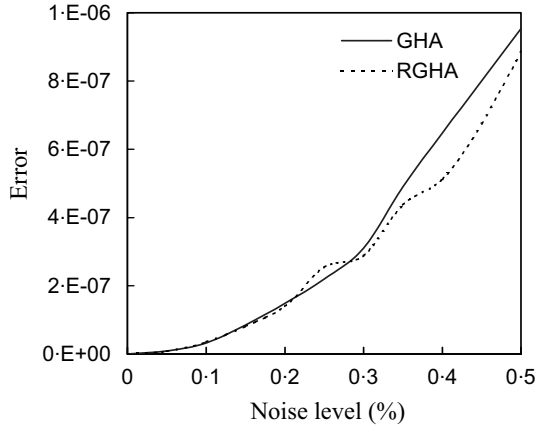

Figure 5. Comparison of errors using GHA and RGHA for sigmoid function.

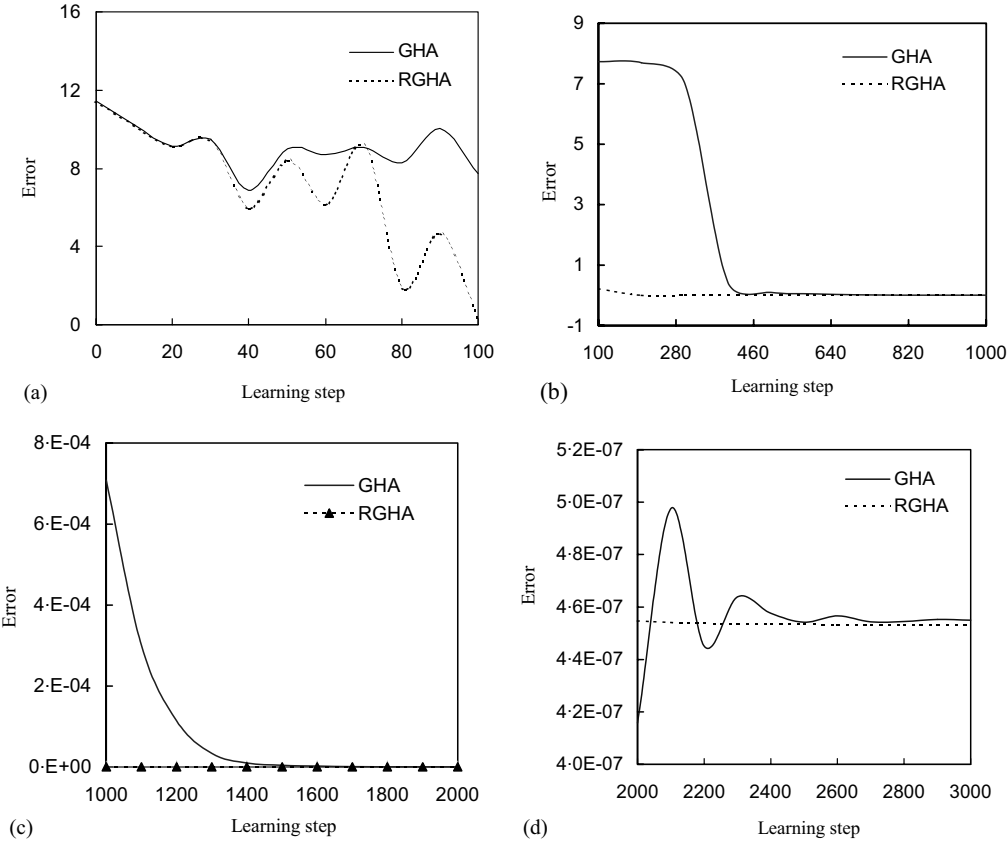

Figure 6. Comparison of errors using GHA and RGHA for learning steps: (a) 0–100; (b) 100–1000; (c) 1000–2000; (d) 2000–3000; (e) 3000–5000; (f) 5000–10 000; (g) 10 000–25 000.

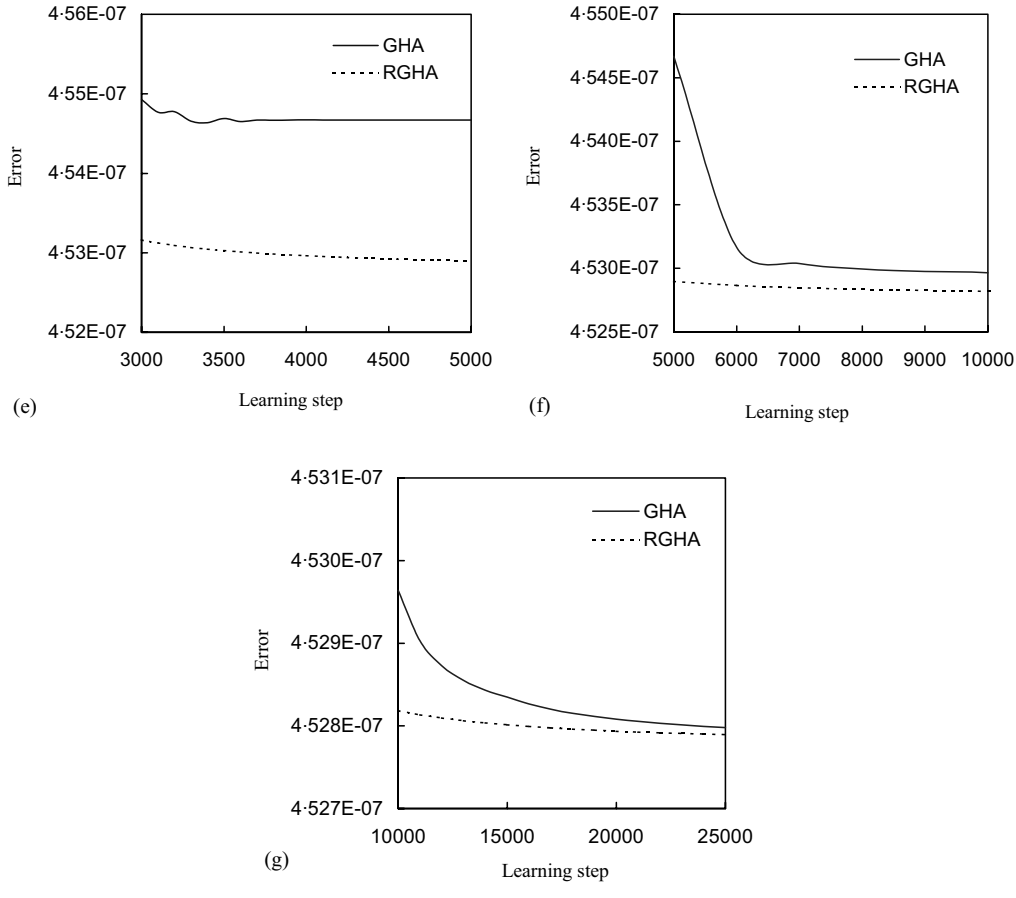

Figure 6. Continued

network have significant influence on the accuracy of the results. Simulations on the RGHA network using the sigmoid function show that the RGHA model is superior to the GHA model in dealing with the noisy data when the learning steps are changed. Figure 5 shows the comparison of the errors defined in equation (22) multiplied by $\sum_{k=1}^{q} S_k^2$, where the errors are the statistic mean values from the 10 times calculations for each noise level, the coefficient of the sigmoid function is 1·5 and the learning steps are 250 000. From a number of numerical simulations it can be seen that compared with the GHA model, the RGHA method has a relative stability of robustness to the noisy data when the noise level is larger than 0·0025. The training convergence is examined for the GHA and RGHA models to the different noise level, which shows that the RGHA approach is superior to the GHA obviously in both robustness and stability.

Figures 6(a)–6(g) show the errors defined in equation (22) multiplied by $\sum_{k=1}^{q} S_k^2$ versus the learning steps of the GHA and RGHA used to extract the first eigenvector of the deflection from the noise-injected snapshots with noise level of 0·003 respectively. The segmented plotting and the local enlargement of the figures enables one to observe clearly the change of the errors versus the learning steps. From Figures 6(a)–6(g) it can be seen that the speed of convergence of the RGHA possesses obvious superiority to that of the GHA. This feature is even more prominent in the initial stages of the training. For example, the

square error reaches the order of magnitude 10^{-6} when 300 iteration steps are completed using the RGHA, whereas it needs 1600 iteration steps for the RGHA to reach the same square error. It shows that for the same accuracy of convergence the RGHA needs fewer iteration steps than the GHA does. Therefore, the RGHA possesses the feature of fast convergence compared with the GHA. From the simulations it can also be seen that in the middle and later stages the error curves obtained using the GHA have considerable fluctuations, whereas the error curves of the RGHA are relatively smooth. It shows that the RGHA model is superior in the stability of convergence to the GHA model.

It has been shown that for the deflection simulation, the first eigenvector $\phi_1^z(x)$ captured 99·99% of the system characteristics while it takes at least four first eigenvectors for the back pressure $\phi_i^p(x)$ to capture the same level in the back pressure simulation [13]. For this reason we choose only one deflection basis vector and four back pressure basis vectors in the simulations as chosen in reference [11], which ensures that the macro-model can be represented well.

The numerical results are given and described below. We define the mean square error between the result using the macro-model and that using the FDM as follows:

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (z_{MM}(x_c, t_i) - z_{FDM}(x_c, t_i))^2, \tag{23}$$

where x_c denotes the centre point of the micro-beam, t_i the sampled time instant, z_{MM} the simulation result using the macro-model, z_{FDM} the finite difference solution of the original non-linear PDE (1) and (2), and m the number of the sampled time series. We first examine the simulation results obtained using the RGHA to the noise-free data. The same results as those by using the GHA model [11] can be obtained by using the RGHA model to the noise-free data, which shows the flexibility of the proposed approach. For example, the mean square error is very small when the system is applied with step voltages which are in the range to create the basis functions. Good accuracies can also be obtained when the voltages far from those used to create the basis functions are employed, and the input voltage wave spectra are changed, for example, when the sinusoidal and ramp input voltages are used.

In order to further examine the flexibility and efficiency of the proposed method in the representation and simulation of the original non-linear PDE, a number of simulations using the RGHA method to the noise-injected data are performed. Figure 7 shows

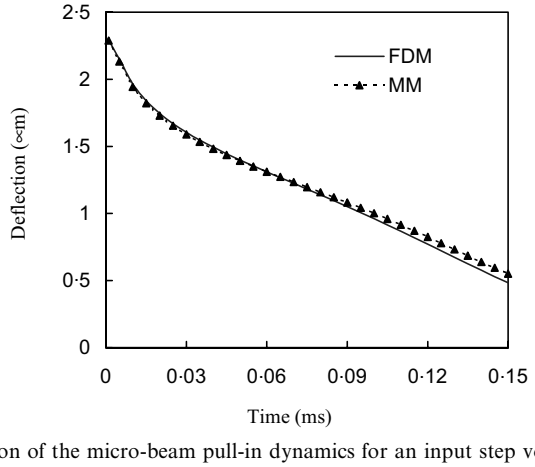

Figure 7. Comparison of the micro-beam pull-in dynamics for an input step voltage $V = 10{\cdot}25$ V.

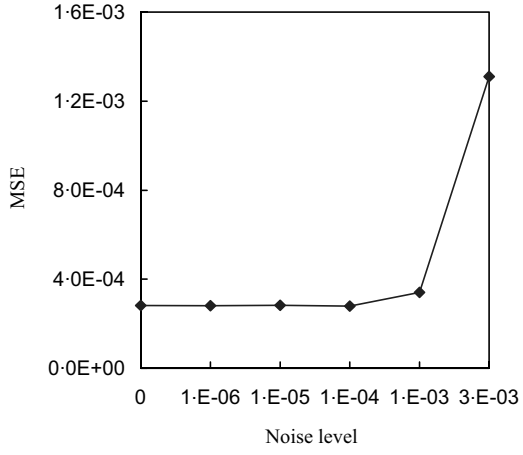

Figure 8. The mean square error of macro-model simulation compared with the FDM solution of the original non-linear equations for an input step voltage $V = 10{\cdot}25$ V.

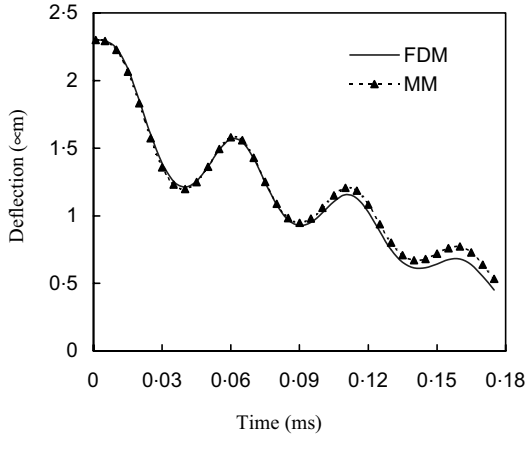

Figure 9. Comparison of the micro-beam pull-in dynamics for a sinusoidal input voltage $V = 14\sin(2 \times 10\,000\pi)$.

a comparison of the deflection of the centre point of the micro-beam between the FDM approximation of the original non-linear PDE and the macro-model (MM) representation when the system is applied with step voltage 10·25 V. The deflection using the FDM is obtained from the noise-free data and that using the RGHA is obtained from the noise-injected snapshots with noise level of 0·003. Figure 8 shows the mean square error of the macro-model simulation compared with the finite difference solution. Figure 9 shows the simulation for a sinusoidal input voltage with magnitude of 14 V at frequency of 10 kHz, where the deflection using the FDM is obtained from the noise-free data and that using the RGHA is obtained from the noise-injected snapshots with noise level of 0·003. Figure 10 shows the mean square error. From the figures it can be seen that for the situations of the noise-free data and the noise-injected data with small noise levels, we can obtain very good accuracy. When the noise level exceeds 0·001, the *MSE* has an obvious increase. However,

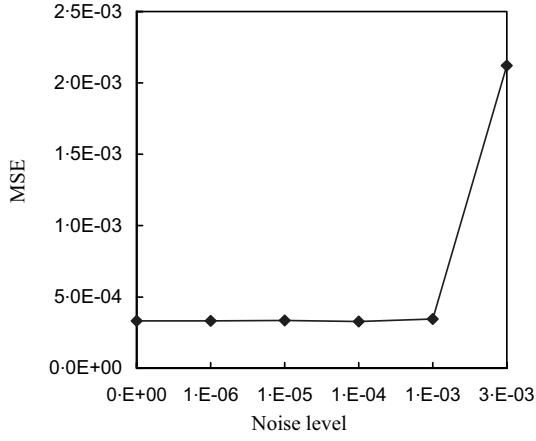

Figure 10. The mean square error of macro-model simulation compared with the FDM solution of the original non-linear equations for a sinusoidal input voltage $V = 14\sin(2 \times 10\,000\pi)$.

for the noise level of 0·003, good results can also be obtained. The satisfactory simulation results can also be observed in the situations that the input voltage wave spectrum is changed to the ramp type and the voltages far from those used to create the basis functions are employed.

6. DISCUSSIONS AND CONCLUSIONS

In this paper, we summarized the applications of the POD methods to model reduction for MEMS and proposed a model reduction approach for the simulation of the non-linear dynamics of MEMS based on a robust PCA neural network model. The macro-model generated by using the eigenvectors extracted from the RGHA neural network as basis functions in Galerkin procedure has shown its flexibility and efficiency in the representation and simulation of the original non-linear PDE. We demonstrated that the proposed method reduces the original non-linear PDE to a macro-model with small number of degrees of freedom, and the macro-model can represent and simulate the original systems almost exactly using the noise-free data. Besides these, this method does not need to compute the input correlation matrix in advance, it needs to find only very few of the required basis functions. This enables the method to possess potential advantages when the measured data are large. As for the computation time efficiency, when Silicon Graphics Origin 2000 is used, it takes more than 30 min to obtain the pull-in time by using FDM with 40×20 mesh when the input step voltage is 10·25 V. In comparison, it requires only few tens of seconds to simulate the pull-in dynamics by the macro-model with an acceptable precision when one basis function for deflection and four basis functions for back pressure are employed for the noise-free data. Simulations are given to show the performances of the proposed method in comparison with the existing GHA neural network method. Comparative experiments show that the proposed RGHA neural network model using the sigmoid function has a number of numerical advantages over the existing methods, such as stability and robustness to noise-injected data and the fast convergence of iterations in the training stages, etc. The simulation results show that the present model reduction technique provides another feasible way for system designers to design and optimize MEMS efficiently and effectively.

ACKNOWLEDGMENTS

The first author is grateful for the support of the NSFC (Grant No. 19872027) and the Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education of PRC.

REFERENCES

1. M. F. A. AZEEZ and A. F. VAKAKIS 2001 *Journal of Sound and vibration* **240**, 859–889. Proper orthogonal decomposition (POD) of a class of vibroimpact oscillations.
2. X. H. MA, A. F. VAKAKIS and L. A. BERGMAN 2001 *American Institute of Aeronautics and Astronautics Journal* **39**, 687–696. Karhunen-Loève modes of a truss: transient response reconstruction and experimental verification.
3. X. H. MA and A. F. VAKAKIS 2001 *Journal of Vibration and Acoustics-Transactions of the American Society of Mechanical Engineers* **123**, 36–44. Nonlinear transient localization and beat phenomena due to backlash in a coupled flexible system.
4. I. T. GEORGIOU, I. B. SCHWARTZ, E. EMACI and A. F. VAKAKIS 1999 *Journal of Applied Mechanics. Transactions of the American Society of Mechanical Engineers* **66**, 448–459. Interaction between slow and fast oscillations in an infinite degree-of-freedom linear system coupled to a nonlinear subsystem: theory and experiment.
5. I. T. GEORGIOU and I. B. SCHWARTZ 1999 *SIAM Journal on Applied Mathematics* **59**, 1178–1207. Dynamics of large scale coupled structural mechanical systems: a singular perturbation proper orthogonal decomposition approach.
6. J. P. CUSUMANO, M. T. SHARKADY and B. W. KIMBLE 1994 *Philosophical Transactions of the Royal Society of London. Physical Sciences and Engineering* **347**, 421–438. Experimental measurements of dimensionality and spatial coherence in the dynamics of a flexible-beam impact oscillator.
7. B. F. FEENY and R. KAPPAGANTU 1998 *Journal of Sound and Vibration* **211**, 607–616. On the physical interpretation of proper orthogonal modes in vibrations.
8. Y. C. LIANG, H. P. LEE, S. P. LIM, W. Z. LIN, K. H. LEE and C. G. WU *Journal of Sound and Vibration* **252**, 527–544. Proper orthogonal decomposition and its applications – Part I: Theory.
9. S. D. SENTURIA 1998 *Proceedings of the IEEE* **86**, 1611–1626. CAD challenges for microsensors, microactuators, and microsystems.
10. E. S. HUNG and S. D. SENTURIA 1999 *Journal of Microelectromechanical Systems* **8**, 280–289. Generating efficient dynamical models for microelectromechanical systems from a few finite-element simulation runs.
11. Y. C. LIANG, W. Z. LIN, H. P. LEE, S. P. LIM, K. H. LEE and D. P. FENG 2001 *Journal of Micromechanics and Microengineering* **11**, 226–233. A neural-network-based method of model reduction for dynamic simulation of MEMS.
12. E. S. HUNG, Y. J. YANG and S. D. SENTURIA 1997 *International Conference on Solid-State Sensors and Actuators, Chicago*, 1101–1104. Low-order models for fast dynamical simulation of MEMS microstructures.
13. W. Z. LIN, K. H. LEE, S. P. LIM and P. LU 2000 *Dynamoics, Acoustics and Simulations, American Society of Mechanical Engineers DSC* **68**, 155–162. A method of model reduction for dynamic simulation of microelectromechanical systems.
14. H. A. C. TILMANS 1996 *Journal of Micromechanics and Microengineering* **6**, 157–176. Equivalent circuit representation of electromechanical transducers. I: lumped-parameter systems.
15. G. K. ANATHASURESH, R. K. GUPTA and S. D. SENTURIA 1996 *Microelectromechanical Systems* (*MEMS*), *American Society of Mechanical Engineers, DSC* **59**, 401–407. An approach to macromodelling of MEMS for nonlinear dynamic simulation.
16. L. D. GABBAY. 1998 *Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology*. Computer aided macromodelling for MEMS.
17. L. D. GABBAY and S. D. SENTURIA 1998 *Proceedings of the Solid-State Sensor and Actuator Workshop, Hilton Head*, 197–220. Automatic generation of dynamic macromodels using quasistatic simulations in combination with modal analysis.
18. F. WANG and J. WHITE 1998 *Microelectromechanical Systems* (*MEMS*) *American Society of Mechanical Engineers DSC* **66**, 527–530. Automatic model order reduction of a microdevice using the Arnoldi approach

19. L. D. Gabbay, J. E. Mehner and S. D. Senturia 2000 *Journal of Microelectromechanical Systems* **9**, 262–269. Computer-aided generation of non-linear reduced-order dynamic macromodels. I: non-stress-stiffened case.
20. J. E. Mehner, L. D. Gabbay and S. D. Senturia 2000 *Journal of Microelectromechanical Systems* **9**, 270–277. Computer-aided generation of nonlinear reduced-order dynamic macromodels. II: stress-stiffened case.
21. R. K. Gupta and S. D. Senturia 1997 *Proceedings of the Tenth IEEE International Workshop on MEMS*. Nagoya, Japan, 290–294. Pull-in time dynamics as a measure of absolute pressure.
22. T, Veijola 1995 *Sensors and Actuators* **A48**, 239–248. Equivalent-circuit model of the squeezed gas film in a silicon accelerometer.
23. B. J. Hamrock 1994 *Fundamentals of Fluid Lubrication*. New York: McGraw-Hill.
24. E. Oja 1982 *Journal of Mathematics and Biology* **15**, 267–273. A simplified neuron model as a principal component analyzer.
25. T. N. Yang and S. D. Wang 2000 *IEEE Transactions on Neural Networks* **11**, 808–810. Fuzzy auto-associative neural networks for principal component extraction of noisy data.
26. B. L. Zhang, M. Y. Fu and H. Yan 2001 *Pattern Recognition* **34**, 203–214. A nonlinear neural network model of mixture of local principal component analysis: application to handwritten digits recognition.
27. T. N. Yang and S. D. Wang 1999 *Pattern Recognition Letters* **20**, 927–933. Robust algorithms for principal component analysis.
28. L. Xu and A. L. Yuille 1995 *IEEE Transactions on Neural Networks* **6**, 131–143. Robust principal component analysis by self-organizing rules based on statistical physics approach.
29. L. Xu 1993 *Neural Networks* **6**, 627–648. Least mean square error reconstruction principle for self-organizing neural-nets.
30. J. Karhunen and J. Joutsensalo 1994 *Neural Networks* **7**, 113–127. Representation and separation of signals using nonlinear PCA type learning.
31. J. Karhunen and J. Joutsensalo 1995 *Neural Networks* **8**, 549–562. Generalizations of principal component analysis, optimization problems, and neural networks.
32. T. D. Sanger 1989 *Neural Networks* **2**, 459–473. Optimal unsupervised learning in a single-layer linear feedforward neural network.
33. Y. Tamura, S. Suganuma, H. Kikuchi and K. Hibi 1999 *Journal of Fluids and Structures* **13**, 1069–1095. Proper orthogonal decomposition of random wind pressure field.
34. P. M. Osterberg and S. D. Senturia 1997 *Journal of Microelectromechanical Systems* **6**, 107–118. M-test: a test chip for MEMS material property measurement using electrostatically actuated test structures.
35. F. L. Luo and R. Unbehauen 1997 *Applied Neural Networks for Signal Processing*. Cambridge: Cambridge University Press.