A neural-network-based method of model reduction for the dynamic simulation of MEMS

Article *in* Journal of Micromechanics and Microengineering · April 2001

DOI: 10.1088/0960-1317/11/3/311

6 authors, including:

Y. C. Liang
Harbin Institute of Technology
51 PUBLICATIONS **427** CITATIONS
SEE PROFILE

WuZhong Lin
Singapore University of Technology and Design
52 PUBLICATIONS **693** CITATIONS
SEE PROFILE

Hp Lee
National University of Singapore
354 PUBLICATIONS **6,274** CITATIONS
SEE PROFILE

Kunwoo Lee
Seoul National University
181 PUBLICATIONS **3,013** CITATIONS
SEE PROFILE

INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF MICROMECHANICS AND MICROENGINEERING
J. Micromech. Microeng. **11** (2001) 226–233 www.iop.org/Journals/jm PII: S0960-1317(01)17441-3

A neural-network-based method of model reduction for the dynamic simulation of MEMS

Y C Liang[1,2], W Z Lin[3], H P Lee[2], S P Lim[2], K H Lee[2,3] and D P Feng[4]

[1] Department of Computer Science, Jilin University, 10 Qian Wei Road, Changchun 130012, Peoples' Republic of China
[2] Centre for Advanced Computations in Engineering Science (ACES), c/o Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260
[3] Institute of High Performance Computing, 89C Science Park Drive 02-11/12, The Rutherford, Singapore 118261
[4] Department of Mathematics, Jilin University, 10 Qian Wei Road, Changchun 130012, Peoples' Republic of China

Received 26 September 2000

Abstract

This paper proposes a neuro-network-based method for model reduction that combines the generalized Hebbian algorithm (GHA) with the Galerkin procedure to perform the dynamic simulation and analysis of nonlinear microelectromechanical systems (MEMS). An unsupervised neural network is adopted to find the principal eigenvectors of a correlation matrix of snapshots. It has been shown that the extensive computer results of the principal component analysis using the neural network of GHA can extract an empirical basis from numerical or experimental data, which can be used to convert the original system into a lumped low-order macromodel. The macromodel can be employed to carry out the dynamic simulation of the original system resulting in a dramatic reduction of computation time while not losing flexibility and accuracy. Compared with other existing model reduction methods for the dynamic simulation of MEMS, the present method does not need to compute the input correlation matrix in advance. It needs only to find very few required basis functions, which can be learned directly from the input data, and this means that the method possesses potential advantages when the measured data are large. The method is evaluated to simulate the pull-in dynamics of a doubly-clamped microbeam subjected to different input voltage spectra of electrostatic actuation. The efficiency and the flexibility of the proposed method are examined by comparing the results with those of the fully meshed finite-difference method.

1. Introduction

The development of increasingly complex microelectromechanical systems (MEMS) demands some sophisticated simulation techniques for design and optimization. The present CAD tools enable the simulation and modelling of the MEMS devices that may not have been constructed. This simulation and modelling are usually presented with nonlinear partial differential equations (PDE) because MEMS devices typically involve multiple coupled energy domains and media and there exist inherent nonlinearities of electrostatic actuation forces and geometric nonlinearities caused by large deformation. Traditional finite-element methods (FEM) or finite-difference methods (FDM) can be used for explicit dynamical simulations of PDE. However, computational prototypes using full models to simulate nonlinear PDE are usually computationally very intensive and time-consuming, making them difficult to use when a large number of simulations are needed. 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 [1, 2].

There have been several approaches to model reduction for the dynamic simulation of MEMS [1–10]. Previous MEMS macromodelling efforts have investigated lumped-parameter techniques [3]. However, it is difficult to construct lumped-element models from continuous systems. Another model reduction approach uses linear modal analysis to generate normal modes which are used as basis functions for the model [4–6]. However, linear modes may not adequately capture all the features of nonlinear behaviour. Wang and White [7] demonstrated that an Arnoldi-based model reduction approach could be used to generate accurate reduced-order models from linear systems in coupled domain simulation of MEMS devices. Again, when the device behaviour is fundamentally nonlinear, the approach loses accuracy and a nonlinear extension needs to be investigated.

In this paper, we propose a neuro-network-based method in which the generalized Hebbian algorithm (GHA) [11] and the Galerkin procedure are used to perform the dynamic simulation and analysis of nonlinear MEMS. The GHA model is a principal component analysis (PCA) neural network, which has been investigated and used in many applications [11–14]. Previous studies have demonstrated that the GHA can be used to solve image coding and texture segmentation problems, to find principal eigenvectors of a correlation matrix in different kinds of seismograms, and to handle sensor array signal processing in the complex domain. This work applies the GHA network to a coupled domain, time-dependent double-clamped microbeam to extract a set of eigenvectors that capture the behaviour of the system efficiently. The eigenvectors are then employed as basis functions in the Galerkin procedure with original governing equations to generate the low-order macromodel for simulating the pull-in dynamics of the microbeam system with electrostatic actuation and squeezed film damping effect. The efficiency and flexibility of the proposed method are examined by simulated experiments.

It should be pointed out that similar approaches to generate the macromodel based on singular value decomposition (SVD) and Karhunen–Loève decomposition (KLD) have been proposed in [2] and [8], respectively. In these two papers, the authors demonstrated how efficient 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 macromodels are generated by extracting global basis functions from the fully meshed model runs in order to parameterize solutions with far fewer degrees of freedom. However, both approaches need some matrix computations in advance, such as the computation on the input correlation matrix in KLD. A large scale matrix eigenvalue problem must be solved and all eigenfunctions of the correlation matrix, or some equivalent matrix from an ensemble of signals, have to be calculated in SVD or KLD. Compared with the existing traditional model reduction methods for the dynamic simulation of MEMS, the present method does not need to compute the input correlation matrix in advance; it needs only to find very few required basis functions. The present method therefore possesses potential advantages when the measured data are large.

2. The system and governing equations

In order to demonstrate the model reduction technique and the generation of the macromodel based on the GHA neural network, we examine a doubly-clamped microbeam pulled in by the electrostatic actuation force with the squeezed gas-film damping effect. Figure 1 shows a cross section of this device [15].

When a voltage V is applied on the top and bottom electrodes, the top deformable microbeam is pulled downwards due to the electrostatic force. At the same time, the narrow air gap between the moving microbeam and the substrate will generate back pressure force on the microbeam due to the squeezed gas-film damping effect [16]. The top microbeam will reach an unstable point and pull in onto 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 this structure can be used as an accelerometer [16] and a pressure sensor [15].

The device shown in figure 1 is a coupled domain system. In general, the microbeam can be modelled by the Euler beam equation with electrostatic actuation force, and the back pressure force can be modelled by the nonlinear Reynold's squeezed gas-film damping equation [17] to become the following PDE:

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

$$\nabla\cdot\left(w^3 p\nabla p\right) = \frac{12\mu}{1+6K}\frac{\partial(pw)}{\partial t} \quad (2)$$

where E is the Young's modulus; $I = Bh^3/12$ is the moment of inertia where B is the width of the microbeam and h is the thickness; T is the residual stress; ρ is the density; μ is the air viscosity and is equal to 1.82×10^{-5} kg (m·s)$^{-1}$; $K(x,t) = \lambda/w$ is the Knudsen number where λ is the mean-free path of the air and is equal to 0.064 μm; $w(x,t)$ is the height of the microbeam above the substrate; $-\varepsilon_0 BV^2/(2w^2)$ is the electrostatic actuation force where V is the applied voltage; ε_0 is the permittivity of free space and is equal to 8.854×10^{-12} Farad m^{-1}; $p(x, y, t)$ is the back pressure force caused by the squeezed gas-film where an isothermal process is assumed; and p_a is the ambient pressure and is equal to 1.013×10^5 Pa.

3. Model reduction based on GHA neural network

There are techniques, such as the FDM or FEM, to convert continuous dynamic nonlinear systems with an 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 purposes. It will be demonstrated that a Galerkin procedure employing the eigenvectors obtained from the GHA neural network can convert the dynamic nonlinear system to a model with a small number of degrees of freedom, while capturing all the essential behaviours of the original system efficiently and accurately.

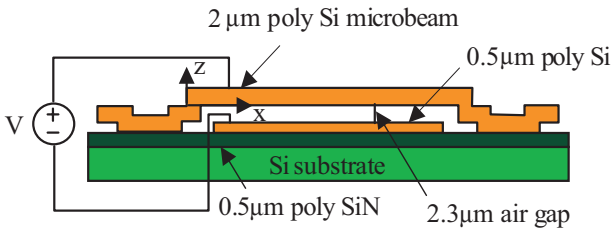

Figure 1. Doubly-clamped microbeam.

The principal components are the most important linear features of the random observation vectors. The PCA is a least-mean-squares technique. 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 [18], the PCA by neural networks and its extensions have become an important research field, both for the interesting implications on unsupervised learning theory and fruitful applications to neural information processing [12].

Over the recent years, several neural network architectures and learning rules for performing the PCA have been proposed in the scientific literature. In this paper, we use the well-known GHA presented by Sanger [11] to extract the principal eigenvectors of the correlation matrix from an ensemble of signals.

The GHA is closely related to classical Hebbian learning algorithms. Hebbian learning rules modify the connection between two units by an amount proportional to the product of the activation of those units. If x is the activation of the input nodes and C is the weight matrix, then $y = Cx$ is the activation at the outputs. Hebbian algorithms modify C by using

$$C(t+1) = C(t) + \beta(t)\left(y(t)x^{\mathrm{T}}(t)\right) \tag{3}$$

where $\beta(t)$ is a sequence of small step-size parameters, which determines the rate of change of the weights.

Oja [18] has shown that if the diagonal elements of CC^{T} are maintained equal to 1 (so that the norm of each row is 1), then a Hebbian learning rule will cause the rows of C to converge to the principal eigenvector of the correlation matrix $R_x = E\left[xx^{\mathrm{T}}\right]$. Oja [18] proposed a network learning algorithm

$$\Delta c_i = \beta(t)\left(y_i x - y_i^2 c_i\right) \tag{4}$$

where c_i^{T} is a row of C, and $y_i = c_i^{\mathrm{T}} x$. Oja showed that equation (4) can be approximated under conditions on x and $\beta(t)$ by a differential equation

$$\dot{c}(t) = R_x c_i - \left(c_i^{\mathrm{T}}(t) R_x c_i(t)\right) c_i(t). \tag{5}$$

Oja [18] then proved that for an arbitrary choice of initial weights, c_i would converge to the principal eigenvector e_1 (or its negative) so long as $c_i(0)^{\mathrm{T}} e_1 \neq 0$ at time zero.

The Oja algorithm only finds the first eigenvector, whereas the GHA presented by Sanger [11] allows us to find the other eigenvectors, and was designed by combining the Oja learning rule (4) and a Gram–Schmidt orthogonalization process.

Let the inputs to a single-layer neural network be an n-dimensional column vector x, the weights an $m \times n$ matrix C, and the outputs an m-dimensional column vector $y = Cx$ with $m < n$. Assume that the values of x are generated by a stationary white random vector stochastic process with a correlation matrix R_x. Therefore, x and y are both time-varying, and C will be time-varying as a result of adaptation through the training algorithm.

The GHA is given by:

$$c_{ij}(t+1) = c_{ij}(t) + \beta(t)\left(y_i(t)x_j(t) - y_i(t)\sum_{k\leqslant i} c_{kj}(t)y_k(t)\right) \tag{6}$$

where c_{ij} is the element of the weight matrix C, which is the connection strength between the jth input neuron and the ith output neuron (c_{ij} is initially assigned random weights), x_j is the jth component of the input vector x, y_i is the ith component of the output vector y, and $\beta(t)$ is a learning parameter that decreases with time such that

$$\lim_{t\to\infty} \beta(t) = 0 \qquad \text{and} \qquad \sum_{t=0}^{\infty} \beta(t) = \infty. \tag{7}$$

We can write equation (6) in matrix form as:

$$\Delta C(t) = \beta(t)\left(y(t)x^{\mathrm{T}}(t) - \mathrm{LT}\left[y(t)y^{\mathrm{T}}(t)\right]C(t)\right) \tag{6'}$$

where LT[·] sets all elements above the diagonal of its matrix argument to zero, thereby making it 'lower triangular'. The second term of the GHA in equation (6) is the Hebbian term, and the third term ensures that the algorithm learns successive eigenvectors (which are the principal components) of the correlation matrix of the input vectors ordered by decreasing eigenvalues. Under conditions (7), Sanger [11] proved the following:

Theorem 1. *If C is assigned random weights at time zero, then with probability 1, equation (6) will converge, and C will approach the matrix whose rows are the first m eigenvectors of the input correlation matrix R_x, ordered by decreasing eigenvalues.*

The significance of the theorem is that we have a procedure which is guaranteed to find the eigenvectors which we seek. The implementation network for the GHA possesses the following features:

(1) There is no need to compute the correlation matrix R_x in advance. This is because the eigenvectors are derived (learned) directly from the input vector. This is an important feature, particularly if the number of inputs is large so that computation and manipulation of R_x are not feasible. For instance, if a network has 4000 inputs, then R_x has 16 million elements, and it may be difficult to find the eigenvectors by traditional methods. However, the GHA requires only the computation of the outer products yx^{T} and yy^{T}, so that if the number of outputs is small the computational and storage requirements can be

correspondingly decreased. If there are five outputs, for example, yx^{T} will have only 20 000 elements, and yy^{T} will have only 25 elements. The GHA takes advantage of this network structure. In MEMS model reduction, the input vector is one snapshot of the deflection or pressure data at one temporal sampling. In general, the number of inputs is large and the number of required outputs is small. Therefore, the GHA provides a practical and useful procedure for finding the required few eigenvectors.

(2) Implementation with local operation. This feature is favourable for parallel computations and parallel hardware.

(3) Good expandability. The updating of the jth neuron is affected only by those neurons with a number less than j. Hence, if the first k principal eigenvectors have been obtained, then the learning of the $(k+1)$th neuron will leave intact the preceding k neuron weight vectors.

In the present model reduction method we use equation (6) of the GHA to obtain the eigenvectors (principal components) by iteratively training the neural network, where the input vector x is the snapshot described in the next section and the rows of the weight matrix C are the eigenvectors which we seek. It should be pointed out that, in our experience, the choice of learning parameter $\beta(t)$ in equation (6) has a profound impact on the convergence speed of the GHA. For the examples in [11], Sanger chose $\beta(t)$ empirically at a fixed value between 0.1 and 0.01, which provided good convergence for his examples. However, the fixed value of $\beta(t)$ in our examples cannot ensure good convergence. Here we adopt an adaptive choice for $\beta(t)$ described in [19], where $\beta(t)$ can be calculated iteratively by

$$\beta(t) = \frac{\beta(t-1)}{\gamma + y^2(t)\beta(t-1)} \tag{8}$$

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. One problem to be studied in the learning algorithm is how to balance the convergence speed and the convergence effectiveness.

4. Generating snapshots

In order to obtain the ensemble of signals or snapshots, the time-dependent deflection $w(x,t)$ and pressure $p(x,y,t)$ in equations (1) and (2) are simulated using the FDM technique. The snapshots of the ensemble must be representative of the dynamic characteristics of the system or device under consideration. We now describe how to obtain the snapshots from the numerical solution of an original nonlinear dynamic system.

For the system shown in figure 1, the pull-in dynamics of the microbeam at a series of different times are simulated using the FDM for an ensemble of applied step voltage to obtain the beam deflection $w_n(x_i, t_s)$ and the back air pressure $p_n(x_i, y_j, t_s)$ ensemble. These deflection and back pressure ensembles are then used as snapshots, i.e. the ensemble of signals for the GHA network to generate the eigenvectors. The ensemble of the applied step voltage is taken to be that of the operating range of the systems.

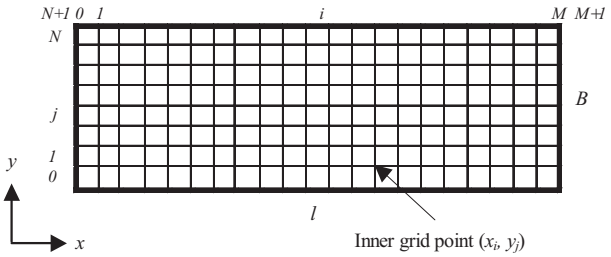

Figure 2. Finite-difference mesh of the microbeam.

To simulate the system shown in figure 1 using the FDM, we discretize the Euler beam equation (1) and the Reynold equation (2) in space to generate an $(M+1)\times(N+1)$ mesh 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 states of the three unknowns $w(x,t)$, $\partial w(x,t)/\partial t$ and $p(x,y,t)$ are projected onto each grid point. This discretization will transform equations (1) and (2) into a set of $M \times N + 2M$ nonlinear ordinary differential equations (ODE). We can use the following state space to represent the unknowns on the grids

$$\dot{x} = \left(\frac{\partial w_1}{\partial t} \cdots \frac{\partial w_M}{\partial t} \frac{\partial^2 w_1}{\partial t^2} \cdots \frac{\partial^2 w_M}{\partial t^2} \frac{\partial p_{11}}{\partial t} \cdots \frac{\partial p_{MN}}{\partial t} \right)^{\mathrm{T}}. \tag{9}$$

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

$$\begin{cases} w = w_0 \quad \dfrac{\partial w}{\partial x} = 0 \quad \dfrac{\partial p}{\partial n} = 0 & (\text{at } x = 0, l) \\ p = p_{\mathrm{a}} & (\text{at } y = 0, B) \end{cases} \tag{10}$$

and initial conditions

$$w = w_0 \qquad \frac{\partial w}{\partial t} = 0 \qquad p = p_{\mathrm{a}} \qquad (\text{at } t = 0). \tag{11}$$

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.

5. Macromodel generation

Using the above-mentioned snapshots as inputs to the GHA neural network, we can obtain the eigenvectors of the input correlation matrix. Then the Galerkin procedure which employs these eigenvectors as basis functions is applied to the original nonlinear governing PDE (1) and (2) to convert it to a macromodel with a small number of ODE.

Because independent deflection and pressure basis functions make the Galerkin derivation simpler and also make sense of the physics of the problem, we perform the principal component extraction using the GHA corresponding to the deflection and pressure, respectively. We denote the eigenvectors (the rows of matrix C) with respect to the deflection as $\phi_i^w(x)$ and those with respect to the pressure as

$\phi_j^p(x, y)$. Then we can represent the deflection $w(x, t)$ and pressure $p(x, y, t)$ as a linear combination of the eigenvectors as follows

$$w(x,t) = w_0 + \sum_{i=1}^{I} a_i^w(t)\phi_i^w(x) \tag{12}$$

$$p(x,y,t) = p_a + \sum_{j=1}^{J} a_j^p(t)\phi_j^p(x,y) \tag{13}$$

where w_0 is the initial gap between the deformable microbeam and the substrate, p_a is the gap air ambient pressure, and I and J are the numbers of basis vectors for the deflection and back pressure, respectively.

Substituting equations (12) and (13) into equations (1) and (2) and applying the Galerkin procedure, we have

$$M_j \frac{d^2 a_j^w}{dt^2} + \sum_{i=1}^{I} K_{ji} a_i^w + f_j = 0 \qquad (j = 1, 2, \cdots, I) \tag{14}$$

$$\sum_{i=1}^{J} H_{ji} \frac{da_i^p}{dt} + \sum_{i=1}^{J} S_{ji} a_i^p + c_j = 0 \qquad (j = 1, 2, \cdots, J) \tag{15}$$

where

$$M_j = \int_L \rho \left(\phi_j^w\right)^2 dx \tag{16}$$

$$K_{ji} = K_{ij} = \int_L \left(EI \frac{\partial^2 \phi_j^w}{\partial x^2} \frac{\partial^2 \phi_i^w}{\partial x^2} + T \frac{\partial \phi_j^w}{\partial x} \frac{\partial \phi_i^w}{\partial x} \right) dx \tag{17}$$

$$f_j = \int_L \left(\frac{\varepsilon_0 B V^2}{2w^2} - \int_0^B (p - p_a)\, dy \right) \phi_j^w dx \tag{18}$$

and

$$H_{ji} = H_{ij} = \int_A \frac{12\mu}{1 + 6K} w \phi_j^p \phi_i^p dx dy \tag{19}$$

$$S_{ji} = S_{ij} = \int_A \left\{ w^3 p \left(\frac{\partial \phi_j^p}{\partial x} \frac{\partial \phi_i^p}{\partial x} + \frac{\partial \phi_j^p}{\partial y} \frac{\partial \phi_i^p}{\partial y} \right) + \frac{12\mu}{1 + 6K} \phi_j^p \phi_i^p \frac{\partial w}{\partial t} \right\} dx dy \tag{20}$$

$$c_j = \int_A \frac{12\mu}{1 + 6K} p_a \phi_j^p \frac{\partial w}{\partial t} dx dy \tag{21}$$

where $\int_L$ indicates the integration along the length of the microbeam and $\int_A$ indicates the integration over the area of the microbeam.

The small set of coupled ODE equations (14) and (15) constitutes the macromodel with global basis functions, which is the low-order dynamic simulation of the original nonlinear PDE system, equations (1) and (2). Since this dynamic macromodel of ODE is generated by the Galerkin procedure employing the eigenvectors extracted from the GHA network, the resulting degree of freedom is usually small. It is very efficient to simulate the system compared with the full model of FEM or FDM which contains a large degree of freedom.

The set of ODE (14) and (15) 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:

$$a_j^w\big|_{t=0} = \frac{\int_L (w - w_0)\big|_{t=0} \phi_j^w dx}{\int_L \left(\phi_j^w\right)^2 dx} \tag{22}$$

$$a_j^p\big|_{t=0} = \frac{\int_A (p - p_a)\big|_{t=0} \phi_j^p dx dy}{\int_A \left(\phi_j^p\right)^2 dx dy}. \tag{23}$$

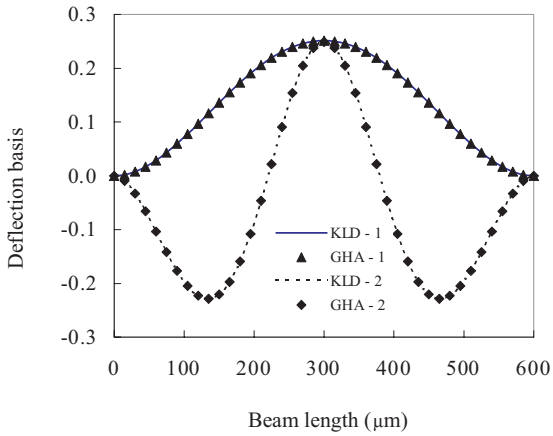

Figure 3. Basis functions for deflection.

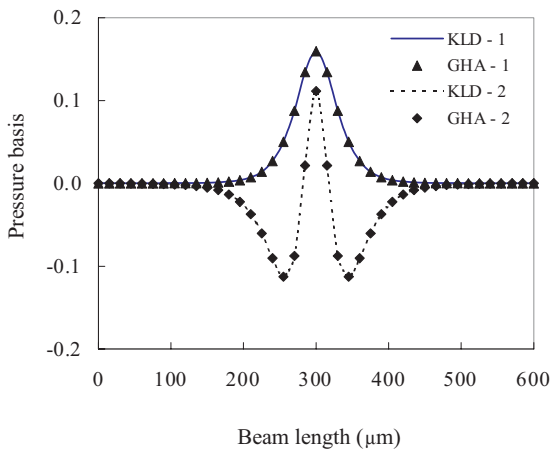

Figure 4. Basis functions for back pressure (along the centre of the beam).

6. Numerical results

In order to demonstrate the efficiency and accuracy of the present model reduction technique using the GHA network, simulated experiments on the MEMS device shown in figure 1 are implemented. The features and dimensions of the microbeam are given in table 1.

The snapshots are obtained from the solutions of equations (1) and (2) by using the FDM mentioned above 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. Each of the 25 snapshots is taken at a fixed time interval from the moment when each step voltage

Table 1. Values of the features and geometric dimensions for the microbeam.

Young's modulus E	Residual stress $T/(hB)$	Beam density $\rho/(hB)$	Knudsen number $K = \lambda/w$	Length l	Width B	Thickness h	Initial gap w_0
149 Gpa	−3.7 Gpa	2330 kg m^{-3}	≈0.028	610 μm	40 μm	2.2 μm	2.3 μm

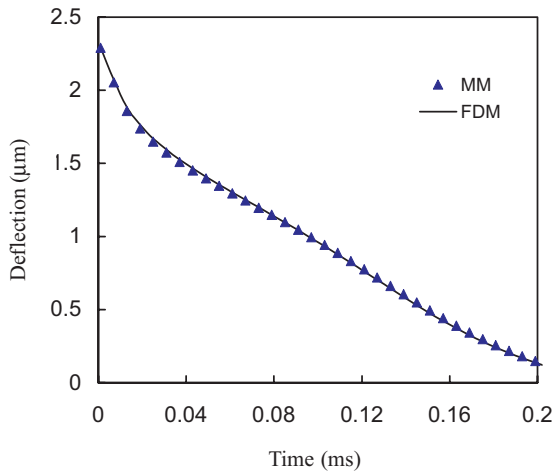

Figure 5. Comparison of the microbeam pull-in dynamics for an input step voltage $V = 10.25$ V.

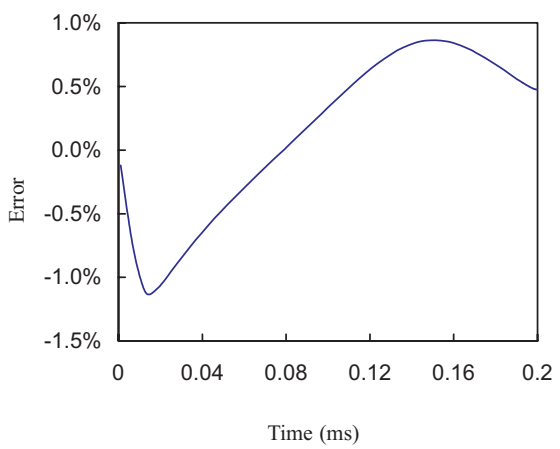

Figure 6. The error of macromodel simulation compared with the FDM solution of the original nonlinear equations for an input step voltage $V = 10.25$ V.

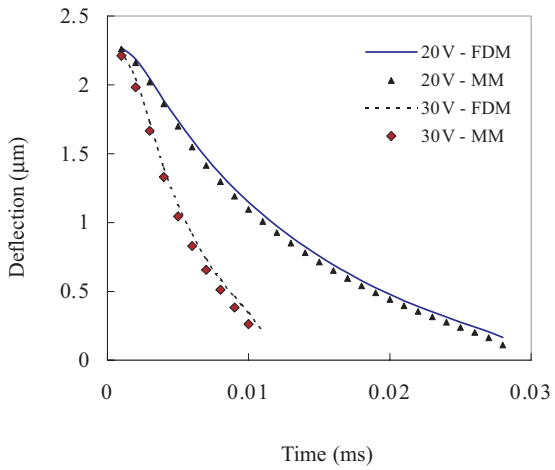

Figure 7. Comparison of the microbeam pull-in dynamics for input step voltages of $V = 20$ V and $V = 30$ V.

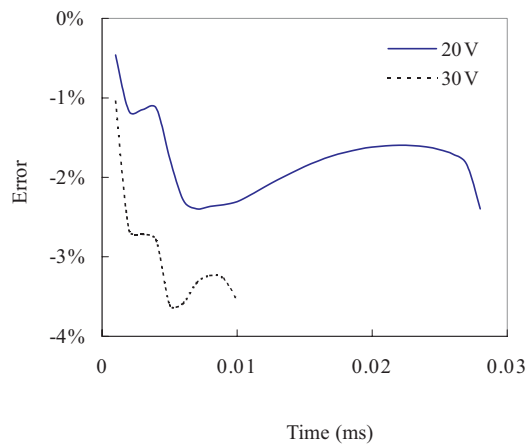

Figure 8. The errors of macromodel simulation for input step voltages of $V = 20$ V and $V = 30$ V.

is applied until the pull-in happens. These snapshots are then used to generate the eigenvectors from the application of the GHA neural network. In order to demonstrate the validity and suitability of the eigenvectors, obtained using the GHA, as proper shape functions, 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 and pressure are plotted in figures 3 and 4, respectively. The higher-order eigenvectors also possess such similarities. The Galerkin procedure uses these eigenvectors given by the GHA as the basis functions to generate the macromodel to represent and simulate the pull-in dynamics. Based on numerical experiments, a mesh size of 40 × 20 for the finite-difference simulation of the original nonlinear equations (1) and (2) is able to generate sufficient accuracy. The minimum step pull-in voltage for this device is calculated to be 8.87 V by the FDM code, which is matched to the experimental data measured at 8.76 V [20].

It has been shown that, for the deflection simulation, the first eigenvector $\phi_1^w(x)$ captures 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 [8]. For this reason, we choose only one deflection basis vector and four back pressure basis vectors in the simulation, which ensures that the macromodel can be represented well.

The numerical results are given and described below. Figure 5 shows a comparison of the deflection of the centre point of the microbeam between the FDM approximation of the original nonlinear PDE (1) and (2) and the macromodel

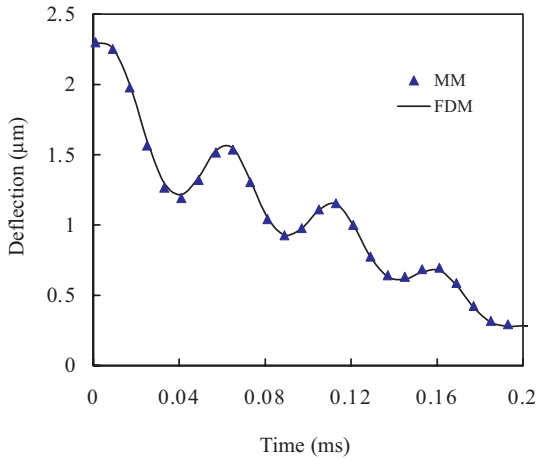

Figure 9. Comparison of the microbeam pull-in dynamics for a sinusoidal input voltage $V = 14\sin(2 \times 10\,000\pi)$.

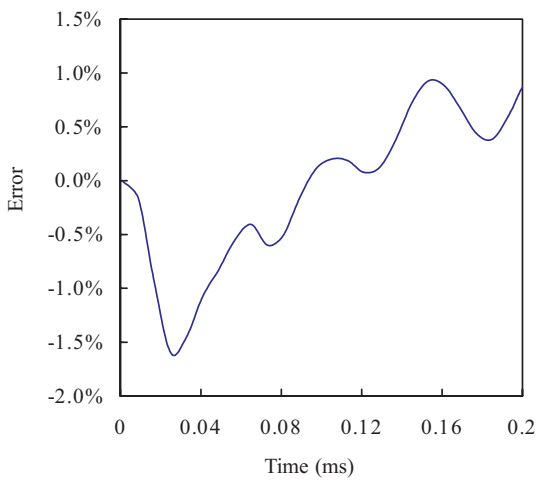

Figure 10. The error of macromodel simulation compared with the FDM solution of the original nonlinear equations for a sinusoidal input voltage $V = 14\sin(2 \times 10\,000\pi)$.

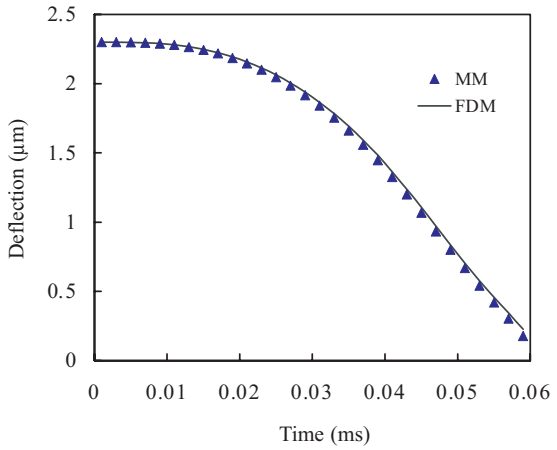

Figure 11. Comparison of the microbeam pull-in dynamics for a ramp input voltage $V = Rt$, $R = 0.4$ V μs^{-1}.

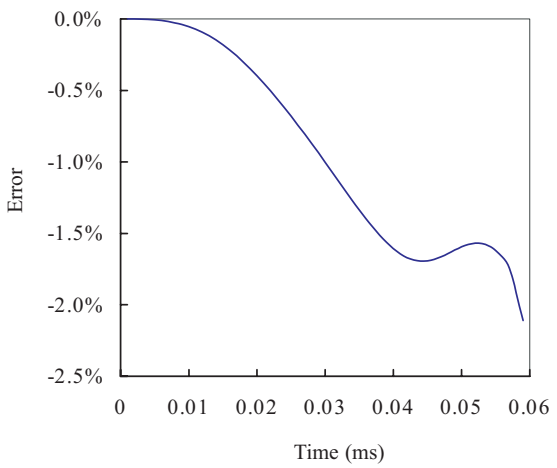

Figure 12. The error of macromodel simulation compared with the FDM solution of the original nonlinear equations for a ramp input voltage $V = Rt$, $R = 0.4$ V μs^{-1}.

(MM) representation when the system is applied with a step voltage of 10.25 V.

We define the error as follows:

$$\text{error} = \frac{w_{\text{MM}} - w_{\text{FDM}}}{w_0} \times 100\% \qquad (24)$$

where w_{MM} denotes the simulation result using the macromodel, w_{FDM} the finite-difference solution of the original nonlinear PDE (1) and (2), and w_0 the initial undeflected gap between the microbeam and substrate. Figure 6 shows that the error is very small ($\leqslant 1.2\%$) when we choose $I = 1$ and $J = 4$.

In order to examine the flexibility of the proposed approach, the simulations with voltages that are far from the voltages used to create the basis functions are performed. The input voltages of 20 and 30 V are used to simulate the pull-in dynamics by the macromodel. It shows that good accuracies can also be obtained when we choose the number of basis vectors $I = 1$ for deflection and $J = 4$ for back pressure without re-generating the macromodel. Figures 7 and 8 show that the errors are still very small, less than 2.5% and 3.7%, respectively.

It is noted that the macromodel generated by the above ensemble of the two different step voltages could also be used to simulate the system when the input voltage wave spectrum is changed. Figure 9 shows the simulation for a sinusoidal input voltage with a magnitude of 14 V at a frequency of 10 kHz, and the error of the macromodel simulation compared with the finite-difference solution is plotted in figure 10. It shows that the macromodel simulation can capture the system dynamics accurately with less than 1.7% when we choose the number of basis vectors $I = 1$ for deflection and $J = 4$ for back pressure without re-generating the macromodel.

To further demonstrate the flexibility of the macromodel generated by the above ensemble of the step voltage to simulate different input voltage spectra, figure 11 shows the simulation of a ramp input voltage of $V = Rt$ with the ramp rate at $R = 0.4$ V μs^{-1}. It is noted that the simulation has a very good result with an error less than 2.1% with $I = 1$ and $J = 4$ as shown in figure 12.

The results demonstrate again that the macromodel can simulate the system well with different input voltage spectra, without re-generating the macromodel. The reason for

obtaining good flexibility is that the step voltage consists of most of the input voltage spectra.

7. Discussion and conclusions

In this paper, we have proposed a model reduction approach for the simulation of the nonlinear dynamics of MEMS based on the neural networks. The macromodel generated by employing the eigenvectors extracted from the GHA network as basis functions in the Galerkin procedure has shown its flexibility and efficiency in the representation and simulation of the original nonlinear PDE. Although it needs an initial process by using the FDM to simulate the original nonlinear PDE to obtain the snapshots, it is demonstrated that the macromodel is very flexible and efficient for simulating the system without re-generation of the macromodel, even when the input voltage spectrum is changed. As for the computation time efficiency, when Silicon Graphics Origin 2000 is used, it takes 32 min 32 s to obtain the pull-in time by using the FDM with a 40×20 mesh when the input step voltage is 10.25 V. In comparison, it requires only 3 min 2 s to simulate the pull-in dynamics by the macromodel with an error of less than 1.2% when using one basis function for deflection and four basis functions for back pressure.

The potential applicability of the proposed neural network method for other MEMS structures is worth mentioning. In general, this methodology could be used to simulate the dynamic behaviours of MEMS, however, for different MEMS devices or systems, the shape functions would be different. The method would be extremely useful for designing and simulating MEMS devices and systems, especially if different types of coupled devices are involved in the system.

In conclusion, we demonstrate that the proposed method reduces the original nonlinear PDE to a macromodel with a small number of degrees of freedom, and the macromodel can represent and simulate the original systems almost exactly. Besides these, this method does not need to compute the input correlation matrix in advance, it needs only to find very few required basis functions compared with other existing model reduction methods for the dynamic simulation of MEMS. This enables the method to possess potential advantages when the measured data are large. For the input vector with dimensions n, the existing methods use memory space order of n^2 in computation because of the correlation matrix, but the neural network based on the GHA learning rule uses a memory space order of n to find the eigenvectors. If the measured data are large, the required memory space may be over the restriction of the PC by using the traditional methods. However, using the GHA learning rule, feeding data iteratively, we may find the eigenvectors in the PC without the memory space problem. Therefore, the computation of eigenvectors in MEMS data may become feasible by using the neural network based on the GHA learning rule in the PC. Successful simulation results show that the present model reduction technique provides another feasible way for system designers to design and optimize MEMS efficiently and effectively.

References

[1] Senturia S D 1998 CAD challenges for microsensors, microactuators, and microsystems *Proc. IEEE* **86** 1611–26

[2] Hung E S, Yang Y J and Senturia S D 1997 Low-order models for fast dynamical simulation of MEMS microstructures *Int. Conf. on Solid-State Sensors and Actuators (Chicago)* pp 1101–4

[3] Tilmans H A C 1996 Equivalent circuit representation of electromechanical transducers: I. Lumped-parameter systems *J. Micromech. Microeng.* **6** 157–76

[4] Anathasuresh G K, Gupta R K and Senturia S D 1996 An approach to macromodelling of MEMS for nonlinear dynamic simulation *Microelectromechanical Systems (MEMS), ASME, DSC* **59** 401–7

[5] Gabbay L D 1998 Computer aided macromodelling for MEMS *PhD Thesis* Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

[6] Gabbay L D and Senturia S D 1998 Automatic generation of dynamic macromodels using quasistatic simulations in combination with modal analysis *Proc. Solid-State Sensor and Actuator Workshop (Hilton Head)* pp 197–220

[7] Wang F and White J 1998 Automatic model order reduction of a microdevice using the Arnoldi approach *Microelectromechanical Systems (MEMS), ASME, DSC* **66** 527–30

[8] Lin W Z, Lee K H, Lim S P and Lu P 2000 A macromodel for dynamic simulation of microelectromechanical systems *Dynamics, Acoustics and Simulations ASME, DSC* **68** 155–62

[9] Gabbay L D, Mehner J E and Senturia S D 2000 Computer-aided generation of nonlinear reduced-order dynamic macromodels: I. A non-stress-stiffened case *J. Microelectromech. Syst.* **9** 262–9

[10] Mehner J E, Gabbay L D and Senturia S D 2000 Computer-aided generation of nonlinear reduced-order dynamic macromodels: II. A stress-stiffened case *J. Microelectromech. Syst.* **9** 270–7

[11] Sanger T D 1989 Optimal unsupervised learning in a single-layer linear feedforward neural network *Neural Netw.* **2** 459–73

[12] Fiori S 2000 An experimental comparison of three PCA neural networks *Neural Process.* **11** 209–18

[13] Zhang Y W and Ma Y L 1997 CGHA for principal component extraction in the complex domain *IEEE Trans. Neural Netw.* **8** 1031–6

[14] Huang K Y 1999 Neural networks for seismic principal components analysis *IEEE Trans. Geosci. Remote Sens.* **37** 297–311

[15] Gupta R K and Senturia S D 1997 Pull-in time dynamics as a measure of absolute pressure *Proc. MEMS 1997* pp 290–4

[16] Veijola T 1995 Equivalent-circuit model of the squeezed gas film in a silicon accelerometer *Sensors Actuators* A **48** 239–48

[17] Hamrock B J 1994 *Fundamentals of Fluid Lubrication* (New York: McGraw-Hill)

[18] Oja E 1982 A simplified neuron model as a principal component analyzer *J. Math. Biol.* **15** 267–73

[19] Diamantaras K I and Kung S Y 1996 *Principal Component Neural Networks: Theory and Applications* (New York: Wiley)

[20] Osterberg P M and Senturia S D 1997 M-test: a test chip for MEMS material property measurement using electrostatically actuated test structures *J. Microelectromech. Syst.* **6** 107–18