Emerging Simulation Approaches for Micromachined Devices

Tamal Mukherjee, *Member, IEEE*, Gary K. Fedder, *Member, IEEE*, D. Ramaswamy, and J.acob White, *Member, IEEE*

***Abstract*—In this survey paper, we describe and contrast three different approaches for extending circuit simulation to include micromachined devices. The most commonly used method, that of using physical insight to develop parameterized macromodels, is presented first. The issues associated with fitting the parameters to simulation data while incorporating design attribute dependencies are considered. The numerical model order reduction approach to macromodeling is presented second, and some of the issues associated with fast solvers and model reduction are summarized. Lastly, we describe the recently developed circuit-based approach for simulating micromachined devices, and describe the design hierarchy and the use of a catalog of parts.**

***Index Terms*—Extraction, macromodeling, MEMS, micromachining, microsystems, model-order reduction, simulation.**

I. Introduction

DECADES of enormous research and capital investment in very large scale intgration (VLSI) technology have made it possible to put more than a million transistors on a square centimeter of silicon, and that investment is now also making it possible to fabricate devices with micron-scale moving parts. The specific techniques used to fabricate such vanishingly small moving parts are often referred to as micromachining, and the potential impact of micromachining is hard to overstate. Micromachined devices will play a key role in making the now pervasive computer technology interact more directly with the physical world. Micromachined devices are already providing such physical-computer interfaces: micromachined accelerometers are used in automobile automatic airbag deployment systems [1], micromachined million mirror arrays are used in computer projection displays [2], and centimeter-sized pressure sensors are used in a range of industrial control applications [3].

Researchers in almost every engineering and scientific discipline are examining ways to harness the ability to fabricate, at low cost, centimeter-sized systems with hundreds of thousands of mechanical parts and transistors. Microresonators, which can replace bulky passive components in communication circuits, may usher in wristwatch-sized cell phones (for better or worse) [4]; active research on microfluidic valves, pumps and mixers may lead to single-chip chemical analysis systems which could be used to make "*in vitro*" medical diagnostic equipment or pocket-sized chemical agent detectors [5]; research on microfabricated turbines and generators [6] may lead to an alternative to batteries for portable energy; and microfabricated parts small enough to capture and hold individual biological cells will accelerate progress in both medical and scientific research [7].

Over the last decade there has been extensive, and successful, research focussed on developing and exploiting micromachining, though there are very few high-volume micromachined products. In addition, almost all the research in applying micromachining technology has been carried out by specialists with many years of focussed training. In contrast, integrated circuit designers do not need such a high level of specialization. Instead, they rely on a coordinated suite of synthesis and verification tools that makes it possible to design an application-specific circuit with high confidence of first-pass success, even without becoming an expert in semiconductor fabrication. The current situation for micromachined device designers is very different. These designers must know the fabrication process intimately, and may even have to design their own process. In addition, the design tools available are often limited and provide only domain-specific simulation or rudimentary layout editing. The combination of inadequate computer-aided design (CAD) tools and rapidly evolving process technology has created an expertise barrier that excludes nonspecialists who would bring important application expertise. Unless this expertise barrier is lowered, primarily through vastly improved CAD tools, it seems unlikely that the potential of micromachining to impact so many different disciplines will be achieved.

Although the need for design tools for micromachining has been recognized for well over a decade, progress has been stymied by a problem whose difficulty has been persistently underestimated. To introduce this problem, consider that for nearly 30 years, integrated circuit designers have relied on circuit simulation. This one tool has nearly eliminated the need to build prototype circuits in order to find major design flaws. One reason for the success of circuit simulation is that its input is the same schematic diagram that designers use to reason about the circuit, and the simulator's output is roughly the same as would be produced by prototyping the circuit and then measuring all the voltages and currents. The problem for micromachined designers is that there is no equivalent of a circuit simulator, and no equivalent of a schematic language

Manuscript received June 1, 2000. This paper was supported in part by the National Science Foundation NSF) under CAREER Award MIP-9 625 471, by the NSF awards MIP-9 901 171 and CCR-9 901 195, and the Defense Advanced Research Projects Agency (DARPA) and Rome Air Force Research Laboratory, Air Force Materiel Command, USAF, under Agreements Number F30602-96-2-0304 and F30602-97-2-0333. This paper was recommended by Associate Editor M. Pedram.

T. Mukherjee is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA.

G. K. Fedder is with the Department of Electrical and Computer Engineering and the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA.

D. Ramaswamy and J. White are with the Department of Electrical Engineering and Computer Science, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0278-0070(00)10457-9.

to describe the device to a simulator, if such a simulator existed. Simulator extension languages like VHDL-AMS [8] can greatly simplify the mechanics of incorporating models for micromachined devices into circuit simulators, but they do not address a more fundamental problem. In a traditional circuit schematic, elements interact only at nodes, and the physical position of elements has limited impact on performance. Neither of these circuit-oriented concepts translate directly to micromachined device design.

The problem of how best to extend circuit simulation to include micromachined devices is fundamental, and as yet, unsolved. For this reason, in this paper we will focus on the emerging approaches to simulation. In order to make some of the issues clearer, we will start in Section II with a brief description of a filter example which uses a micromachined device. Then in Section III, we will describe the currently most widely used approach to extending circuit simulation, that of generating semi-analytical macromodels for each type of micromachined device. Then, in Section IV, we will discuss the desirability and difficulties of replacing the semi-empirical macromodeling approach with a purely numerical approach based on computer simulation and model-order reduction. In Section V, we will approach the simulation problem from the specification side, and discuss a hierarchy of elements and a schematic description for certain classes of micromachined devices. Finally, in our conclusions, we try to tie together these separate approaches and loosely conjecture about where the field is going.

II. Example and Background

In this section, we describe a design example in order to help illustrate the difficulties in developing extensions to a circuit simulator for micromachined devices. The example is a bandpass filter which uses a series of comb-drive micromachined resonators [9], shown in a high-level form in Fig. 1. The high-level diagram is best described by tracing from input to output. The input v_{in} in Fig. 1 is connected to a triangle which represents a transistor amplifier. The parallel plates adjacent to F_e indicates an electrical to mechanical conversion. The force F_e accelerates the first mass in a spring-coupled cascade of spring-mass-dashpot resonators. Finally, the parallel plates adjacent to C indicates a mechanical to electrical conversion which feeds a transistor amplifier which generates v_{out}.

In order to better understand the filter example, consider a single comb-drive lateral microresonator, a layout is shown in Fig. 2. An SEM of the fabricated device is shown in Fig. 3. The polysilicon resonator structure, which appears white in the SEM picture and gray in the top view diagram, has been released from the substrate underneath except at certain attachment points. The thick black lines in Fig. 2 are used to show where the polysilicon structures are attached to the substrate, or anchored. As is clear from Fig. 2, the structure has three separately anchored parts: a left comb, a right comb, and a dual-comb central shuttle which is anchored to the substrate only through thin polysilicon beams. The thin beams serve two purposes. They act as springs and allow the central shuttle to oscillate from left to right, and they provide a conductive path between the

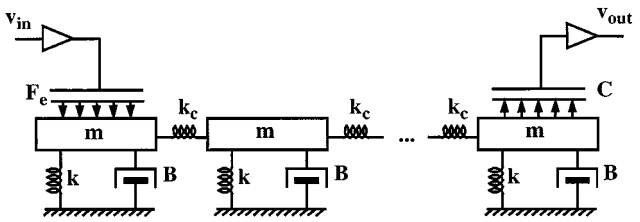

Fig. 1. System-level behavioral model of a multiresonator filter.

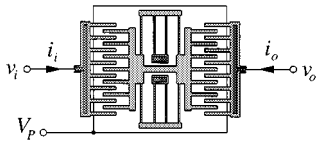

Fig. 2. Overhead view of the lateral microresonator. (Figure courtesy of C. Nguyen and R. Howe.)

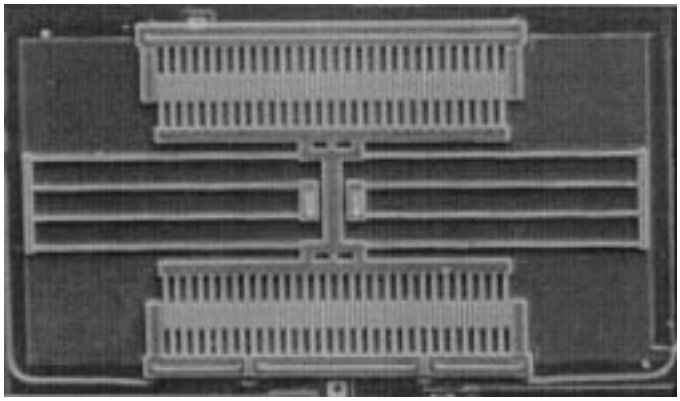

Fig. 3. SEM of an integrated CMOS resonator. (Photo courtesy of C. Nguyen and R. Howe.)

central shuttle and a fixed conducting plate held at a bias potential V_p. The interdigited combs generate electrostatic forces which pull the shuttle to the left when $v_i > v_0$ and to the right when $v_0 > v_i$, assuming both v_i and v_0 are larger than V_p. If out-of-phase sinusoidally varying voltages are applied to v_i and v_0, along with a dc offset, then the amplitude of the central shuttle's steady-state oscillation will be strongly frequency dependent.

As the diagram in Fig. 2 suggests, all that is needed to include the resonator in a circuit simulator is to determine a relationship between the currents i_i and i_0 and the voltages V_p, v_i, and v_0. And at least formally, the needed current-voltage relationship can be derived by determining the mechanical material properties and then solving a coupled system of time-dependent partial differential equations on a moving boundary. In particular, the shuttle accelerates and the tethers bend elastically in response to forces generated by exterior electric fields and viscous drag. The drag will not be exactly zero if the resonator is packaged in a vacuum, because there are still mechanical energy loss mechanisms which create an effective drag force.

Though the statement in the preceding paragraph is true, it is hides many of the important difficulties. Determining a device's three-dimensional (3-D) structure and associated material properties requires a detailed understanding of the fabrication processes as well as a set of carefully designed experiments [10].

Solving coupled systems of 3-D time-dependent partial differential equations with a complicated moving boundary requires sophisticated numerical techniques and a great deal of computer time [11], [12]. Finally, developing a current-voltage relation for a single resonator as in Fig. 2 presumes how the resonator will be used in a system, and the presumption may be incorrect. As an example, reconsider the original filter with the three stage resonator shown in Fig. 1. In the multistage resonator, the single-stage resonators are coupled together by springs which are implemented using thin polysilicon tethers. For the multistage resonator, the most important aspect to model well is the transfer function from the input of the first resonator to the output of the last resonator. However, there will be no way to arrive at that transfer function by "composing" the previously mentioned single resonator current-voltage models. Instead, an entirely new model will be needed for a two-stage resonator, and then another model for a three-stage resonator, and yet another model for a four stage resonator, etcetera. And if these models are going to be derived by solving time-dependent partial differential equations for structures as geometrically complicated as a multistage resonator, the computer time required may cast a more positive light on building prototypes.

In order to assess the importance of issues like deriving structure and material properties from layout and process information, the computational cost of partial differential equation solution, or model composibility, it is worth recalling that for integrated circuit design, simulator use can be divided into two broad classes. Early in the circuit design process, during a *synthesis or optimization phase*, many alternatives are being considered, and designs are typically represented only with a schematic. That is, circuit element interconnection is specified, but no layout information exists. The simplicity of the schematic representation both builds intuition and accelerates examining alternatives, though layout parasitics are either ignored or crudely estimated. As the design matures, when the circuit layout has been determined, a *verification phase* begins. Circuit simulators are then combined with layout extraction techniques to check that the final layout results in a circuit with the desired performance. Such a two stage approach also seems to be a natural fit to designing the microresonator filter. It would be very efficient if most of the layout details could be avoided while examining alternatives such as: fewer or more resonator stages; fewer or more comb fingers; heavier or lighter shuttles; and longer or shorter or more serpentine tethers. Then only during the verification phase would it be necessary to work with the layout and combine extraction techniques with simulation.

III. Semi-Analytical Macromodeling

By far the most common approach to including a micromachined device in circuit simulation is to analyze the device approximately, so as to generate a macromodel in the form of either a circuit or a low-order system of differential equations [13]. Generating the form of these models requires the device designer's physical insight, and can be as much art as science, though issues such as energy conservation can provide guidelines [14]. Once the form of the macromodel has been discerned, then values for the parameters must be determined. These parameters can be determined analytically, or by experiment, or by using numerical simulation. The decomposition between macromodel form and parameterization is not a precise one, and is best described by example. Below, a simple macromodel form for the single resonator example of Fig. 2 is derived, and then several alternatives for determining model parameters are examined. The merits and deficits of semi-analytic macromodels are then described in general.

A. Example Model Form

In order to develop a model for the resonator which can be incorporated in a circuit simulator, the resonator voltages must be related to the resonator currents. Resonators are usually modeled as RLC circuits [9] as such models help develop intuition. A differential equation model is developed below because the setting is more generally applicable. To begin, from Fig. 2, the currents i_i and i_o can be related to the voltages v_i, v_o and V_p by first noting that

$$i_i(t) = \frac{d}{dt} Q_L(v_i - V_p, x) \quad i_o(t) = \frac{d}{dt} Q_R(v_o - V_p, x) \quad (1)$$

where x is the displacement from center of the dual-comb shuttle, and Q_R and Q_L are the net charges on the left and right anchored combs, respectively. A simple parallel plate analysis suggests that the comb capacitance is an affine function of the displacement x, in which case the comb charges will satisfy an equation of the form

$$\begin{aligned} Q_L(v_i - V_p, x) &\approx (C_0 - C_1 x)(v_i - V_p) \\ Q_R(v_i - V_p, x) &\approx (C_0 + C_1 x)(v_o - V_p) \end{aligned} \quad (2)$$

where C_0 is the comb-pair capacitance at $x = 0$ and C_1 is the x derivative of that capacitance. Note that there is only one C_0 and one C_1 so we have exploited the left-right symmetry in the problem. Finally, a very simple spring-mass mechanical model for the comb suggests that the shuttle displacement, x, satisfies an equation of the form

$$M \frac{d^2}{dt^2} x + K_d \frac{d}{dt} x + K_s x = K_e((v_o - V_p)^2 - (v_i - V_p)^2) \quad (3)$$

where

- M mass of the shuttle;
- K_d drag force on the comb generated by the surrounding fluid (typically air at room pressure);
- K_s spring constant associated with the thin teathers;
- K_e constant which relates the electrostatic force generated by the comb to the square of the applied voltage.

B. Determining Model Parameters

A very simple analysis of the microresonator was used above to develop a differential equation system macromodel. The model is given by the combination of (1), (2), and (3). It is worth noting that the model is nonlinear and has quite a few parameters. Until the parameters are set, there is only a "form"

for the macromodel. In the above example, and for macromodels in general, specifying the macromodel form usually implies: assigning a set of state variables, determining which time derivatives appear, representing which state variables interact, and specifying where the parameters appear. It is also quite common to include certain expected nonlinearities, as was done with the squared potential in (3).

The above macromodel has many parameters, C_0, C_1, M, K_d, K_s, and K_e, so it is tempting to suggest that the model could fit anything. Since the macromodel is intended only as an example, we will consider the issue of how to determine the parameters rather than focusing on how to improve the model. There are two main issues associated with macromodel parameter selection.

1) *Will the parameters be determined by physical analysis or through fitting to measured or simulated data?*
2) *Will a new set of parameters be determined every time a change is made in the device geometry, or will the parameters be given as an explicit function of design attributes?*

Most macromodels use some combination of analysis and data fitting to determine the parameters. For example, the shuttle mass of the resonator, M, is easily determined from the geometry and material properties, but would be difficult to measure directly. There are techniques for estimating shuttle drag [15], K_d, though recent studies suggest that numerically solving the Stokes equation yields higher accuracy [16]. Finite-element analysis or measurements might also be superior to trying to use linear beam theory when attempting to determine the spring coefficient K_s [10], [17]. In general, as software for solving partial differential equations improves, parameter estimation will be more heavily based on results from simulation rather than analytical techniques.

There are many aspects of a microresonator that a designer can alter to try to improve performance including: the number and length of comb fingers, the tether lengths and widths, and the shuttle proof mass. One advantage of using physical analysis to determine macromodel parameters is that the analysis usually reveals an explicit form for the dependence of the parameters on design attributes. Macromodels whose parameters are given as explicit functions of design attributes are of obvious value during the synthesis and optimization phase of design. If the electrostatic force constant, K_e, is estimated analytically using a parallel-plate formula, then the resulting formula for the parameter K_e will include a term which grows linearly with the number of comb fingers.

Deriving macromodel parameters by fitting to measured data or simulation results does not preclude generating macromodels whose parameters are given as explicit functions of the design attributes. It is possible to use a multivariate polynomial fitting procedure to generate these explicit functions, but the procedure is not completely automatic and requires expert input [17]. To understand the difficulty, consider a micromachined device whose macromodel has a parameter P that is dependent on the value of d design attributes. We will denote the values of the d design attributes as a d-length vector u. Then, our problem becomes one of determining an explicit representation of $P(u)$.

A seemingly straight-forward approach to finding an explicit representation of $P(u)$ is to use a multivariate polynomial. To see the difficulty generated by such an approach, consider all the terms of a second-order polynomial in three design attributes

$$P(u) = \beta_0 + \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 + \beta_4 u_1 u_1 + \beta_5 u_1 u_2 + \beta_6 u_1 u_3 + \beta_7 u_2 u_2 + \beta_8 u_2 u_3 + \beta_9 u_3 u_3 \quad (4)$$

where $u = [u_1 u_2 u_3]^T$ are the design attributes and the β_is are the unknown coefficients of the multivariate polynomial. As should be clear from the above example, the number of terms in a qth order d-dimensional polynomial is proportional to d^q. This implies that it will be computationally hopeless to use multivariate polynomials directly to represent parameter variation when the number of design attributes exceeds a half dozen. Instead, the polynomials will have to be modified by "pruning" unnecessary terms. Determining which terms can be safely discarded requires significant mathematical and physical insight.

There is a second issue associated with fitting $P(u)$ with a polynomial, and this issue is sometimes referred to as the "design-of-experiments" problem. Consider again the problem of fitting $P(u)$ with a second-order polynomial in three design attributes. In order to compute the ten unknown polynomial coefficients $\beta_0, \ldots, \beta_9$, values for $P(u)$ must be computed for at least ten different values of the three-length vector u. There are several approaches to determining the "best" test values for u [17] based on statistical arguments, but the key difficulty and its cure can be seen by examining the matrix equation associated with the fitting. Again for our second-order example, assume there are $l+1$ measurements, $l \geq 9$, and let the measurements (either real or from simulation) be at points u^i for $i = 0$ to l. The matrix equation for the βs is then given by

$$\begin{bmatrix} P(u^0) \\ P(u^1) \\ \vdots \\ P(u^l) \end{bmatrix} = \begin{bmatrix} 1 & u_1^0 & u_2^0 & \cdots & u_3^0 u_3^0 \\ 1 & u_1^1 & u_2^1 & \cdots & u_3^1 u_3^1 \\ & \vdots & & & \vdots \\ 1 & u_1^l & u_2^l & \cdots & u_3^l u_3^l \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_9. \end{bmatrix}. \quad (5)$$

Note that the system will be square when the number of measurements equals the number coefficients, though typically the number of measurements far exceeds the number of coefficients.

As is clear from examining (5), the points u_0 through u_l should be chosen to make the rows (or the columns) of the matrix in (5) as close to mutually orthogonal as possible. Finding values of u^is which generate a nearly orthogonal matrix can be accomplished using a one test point at a time algorithm.

C. Merits and Deficits

The semi-analytic approach to macromodeling is in far wider use than the methods to be described below. And, since this method is "free-form", there are no restrictions as to what kind of micromachined devices can be modeled. In addition, if such macromodels are carefully parameterized, they can be used to excellent effect during the synthesis and optimization phase of design.

There are two difficulties with the semi-analytic macromodeling approach. The most obvious problem is that there is no standard method for generating these macromodels, and the only way to determine when the models are sufficiently accurate is by comparing the macromodel's results to those of experiments or very detailed simulation. The second problem

is simply that the macromodeling approach provides very little verification. One can not "extract" the macromodel from layout, or add in parasitics. In addition, since the device designer usually generates the macromodel, there is no independent check on whether an important interaction is being ignored.

IV. Numerical Macromodeling

When the first- and second-order behaviors of a micromachined device are well-understood, the most efficient strategy for including the device in a circuit simulator is to develop the kind of semi-analytic macromodel discussed in Section III. Given how rapidly the micromachining technology is changing, it is rarely possible to wait for such device expertise to develop. And since it is almost impossible to design systems which use micromachined devices without access to reliably accurate macromodels, slow macromodel development translates into slow technology deployment. For this reason, there has been a steady effort over the last decade to develop nearly automatic approaches for generating accurate macromodels of micromachined devices starting from only layout and process descriptions. Most efforts is this area are following a three step approach [18], [11], [19].

- Use modified extrusion to generate an approximate 3-D structure from a layout and process description [10], [20].
- Use fast coupled-domain 3-D simulation techniques to analyze the entire micromachined device [12], [21].
- Use a projection-based model-order reduction strategy to generate macromodels from 3-D simulation [22].

Below we describe some of the recent developments and persistent challenges in fast coupled-domain simulation and model-order reduction.

A. Fast Coupled-Domain 3-D Solvers

The above strategy for generating macromodels by performing projection-based model-order reduction relies on the ability to simulate an entire micromachined device in a reasonable period of time. In order to simulate the microresonator in Fig. 2, for example, it is necessary to solve a complicated 3-D moving boundary problem which couples elastic, fluidic and electrostatic forces. Simulating such problems with standard finite-element methods is nearly intractable, because it is necessary to discretize the volume in both the interior and exterior of the resonator. In order to simulate entire micromachined devices, it was first necessary to develop faster techniques for analyzing the exterior field problems. Then, it became necessary to develop robust approaches for coupling those fast techiques to standard finite-element algorithms for computing elastic deformation.

1) Fast Field Solvers: For most micromachined devices, the electrostatic and fluidic forces in the exterior of the device satisfy linear partial differential equations. Specifically, the surface electrostatic forces can be determined by solving an exterior Laplace's equation and the fluid drag forces can be determined by solving an exterior Stokes equation. For both equations, it is possible to derive integral formulations which avoid the exterior volume entirely and instead relate potentials to forces in the electrostatic case, and velocities to forces in the fluid case.

More specifically, the electrostatic potential and the fluid velocity, assuming Stokes flow, both satisfy an integral equation over the device surface given by Green's theorem

$$u(x) = \int_{\text{surfaces}} G(x,x') \frac{\partial u(x')}{\partial n} + \frac{\partial G(x,x')}{\partial n} u(x')\, da' \tag{6}$$

where

- u either the electrostatic potential or the fluid velocity;
- x a point on the surface;
- $\partial/\partial n$ derivative in the direction normal to the polysilicon surface.

Discretization of the above integral equation leads to a dense system of equations which becomes prohibitively expensive to form and solve for complicated problems. To see this, consider the electrostatics problem of determining the surface charge given the potential on conductors. A simple discretization for the electrostatics problem is to divide the polysilicon surfaces into n flat panels over which the charge density is assumed constant. A system of equations for the panel charges is then derived by insisting that the correct potential be generated at a set of n test, or collocation, points. The discretized system is then

$$Pq = \Psi \tag{7}$$

where q is the n-length vector of panel charges, Ψ is the n-length vector of known centroid potentials. Since the Green's function for electrostatics is the reciprocal of the separation distance between x and x'

$$P_{i,j} = \int_{\text{panel}_j} \frac{1}{4\pi\epsilon ||x_i - x'||}\, da' \tag{8}$$

and, therefore, every entry in P is nonzero.

If direct factorization is used to solve (7), then the memory required to store the matrix will grow like n^2 and the matrix solve time will increase like n^3. If instead, a preconditioned Krylov-subspace method like GMRES [23] is used to solve (7), then it is possible to reduce the solve time to order n^2 but the memory requirement will not decrease.

In order to develop algorithms that use memory and time that grows more slowly with problem size, it is essential *not* to form the matrix explicitly. Instead, one can exploit the fact that Krylov-subspace methods for solving systems of equations only require matrix-vector products and not an explicit representation of the matrix. For example, note that for P in (7), computing Pq is equivalent to computing n potentials due to n charged panels and this can be accomplished approximately in nearly order n operations [24], [25]. To see how to perform such a reduction in cost, consider Fig. 4. The short-range interaction between close-by panels must be computed directly, but the interaction between the cluster of panels and distant panels can be approximated. In particular, as Fig. 4 shows, the distant interaction can be computed by summing the clustered panel charges into a single multipole expansion (denoted by M in Fig. 4), and then the multipole expansion can be used to evaluate distant potentials.

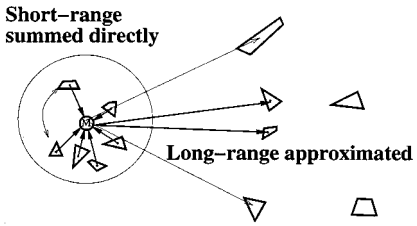

Fig. 4. A Cluster of collocation points separated from a cluster of panels.

Several researchers simultaneously observed the powerful combination of integral equation approaches, Krylov-subspace matrix solution algorithms, and fast matrix-vector products [26], [32]. Perhaps the first practical use of such methods combined the fast multipole algorithms for charged particle computations with the above simple discretization scheme to compute 3-D capacitance and electrostatic forces [27]. Higher order elements and improved efficiency for higher accuracy have been the recent developments [21], [31]. The many different physical domains involved in micromachined devices have focussed attention on fast techniques which are Green's function independent, such as the precorrected-fast Fourier transform (FFT) schemes [25], [29].

2) Coupled-Domain Simulation: Self-consistent electromechanical analysis of micromachined polysilicon devices typically involves determining mechanical displacements which balance elastic forces in the polysilicon with electrostatic pressure forces on the polysilicon surface. The technique of choice for determining elastic forces in the polysilicon is to use finite-element methods to generate a nonlinear system equations of the form

$$F(u) - P(u,q) = 0. \tag{9}$$

where

- u vector of finite-element node displacements;
- F relates node displacements to stresses;
- P force produced by the vector representing the discretized surface charge q.

Note that as the structure deforms, the pressure changes direction, so P is also a function of u. One can view this mechanical analysis as a "black box" which takes an input, q, and produces an output u as in

$$u = H_M(q). \tag{10}$$

In order to determine the charge density on the polysilicon surface due to a set of applied voltages, one can use a fast solver, as described above. One can view the electrostatic analysis as a "black box" which takes, as input, geometric displacements, u, and produces, as output, a vector of discretized surface charges, q, as in

$$q = H_E(u). \tag{11}$$

Self-consistent analysis is then to find a u and q which satisfies both (10) and (11).

A simple relaxation approach to determining a self-consistent solution to (10) and (11) is to successively use (10) to update displacements and then to use (11) to update charge. Applying (10) implies solving the nonlinear equation, (9), typically using Newton's method.

Although the relaxation method is simple it often does not converge. Instead one can apply Newton's method to the system of equations

$$\begin{bmatrix} q \\ u \end{bmatrix} - \begin{bmatrix} H_E(u) \\ H_M(q) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{12}$$

in which case the updates to charge and displacement are given by solving

$$\begin{bmatrix} I & -\frac{\partial H_E}{\partial u} \\ -\frac{\partial H_M}{\partial q} & I \end{bmatrix} \begin{bmatrix} \Delta q \\ \Delta u \end{bmatrix} = - \begin{bmatrix} q - H_E \\ u - H_M \end{bmatrix}. \tag{13}$$

The above method is referred to as a multilevel Newton method because forming the right-hand side in (13) involves using Newton's method to apply H_M.

In order to solve (13), one can apply a Krylov-subspace iterative method such as GMRES. The important aspect of GMRES is that an explicit representation of the matrix is *not* required, only the ability to perform matrix-vector products. As is clear from examining (13), to compute these products one need only compute $(\partial H_M/\partial q)\Delta q$ and $(\partial H_E/\partial u)\Delta u$. These products can be approximated by finite differences as in

$$\frac{\partial H_M}{\partial q}\Delta q \approx \frac{H_M(q+\alpha\Delta q) - H_M(q)}{\alpha} \tag{14}$$

where α is a very small number. Therefore, this matrix-free multilevel-Newton method [34] can treat the individual solvers as black boxes. The black box solvers are called once in the outer Newton loop to compute the right hand side in (13) and then called once per each GMRES iteration. Computing $H_M(q + \alpha d_1)$ means using an inner loop Newton method to solve (9), which is expensive, though improvements can be made [12]. An important advantage of matrix-free multilevel-Newton methods is that it is not necessary to modify either the mechanical or electrostatic analysis programs.

In order to improve the efficiency of the multilevel-Newton method, one would like to avoid solving (9) on every GMRES iteration. Instead, it is possible to modify the finite-element mechanical solver so that perturbations in displacements due to perturbations in charge can be directly computed. To see this, note that by definition

$$\left(\frac{\partial u}{\partial q}\right) = \frac{\partial H_M}{\partial q} = \left(\frac{\partial (F-P)}{\partial u}\right)^{-1} \frac{\partial P}{\partial q} \tag{15}$$

provided $(\partial(F-P)/\partial u)$, the pressure modified stiffness matrix, is nonsingular. If the charge is perturbed by δq, then the perturbation to the displacements, δu is given by solving

$$\frac{\partial (F-P)}{\partial u}\,\delta u = \frac{\partial P}{\partial q}\,\delta q. \tag{16}$$

Using (16) to compute $(\partial H_M/\partial q)d_1$ is quite efficient because $\partial(F-P)/\partial u$ will have already been constructed and factored when using Newton's method to compute the right side in (13). This reduces the cost in each GMRES iteration from a nonlinear solve to a single backsolve and a multiplication by $\partial P/\partial q$. The multiplication by $\partial P/\partial q$ is inexpensive

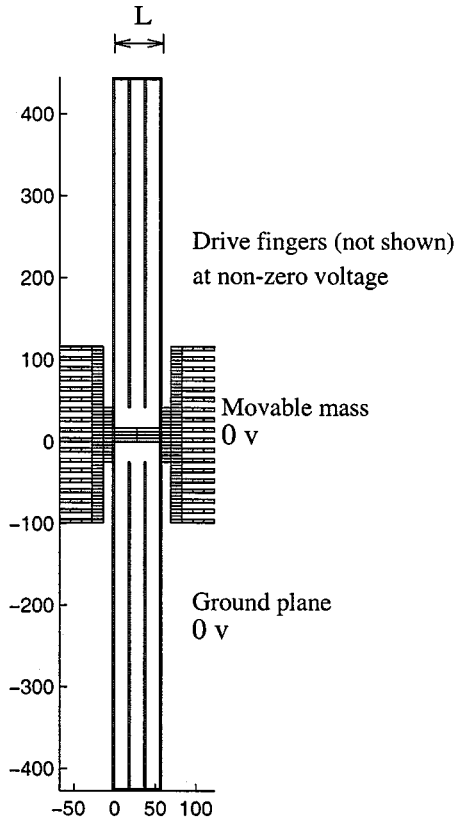

Fig. 5. Comb drive resonator.

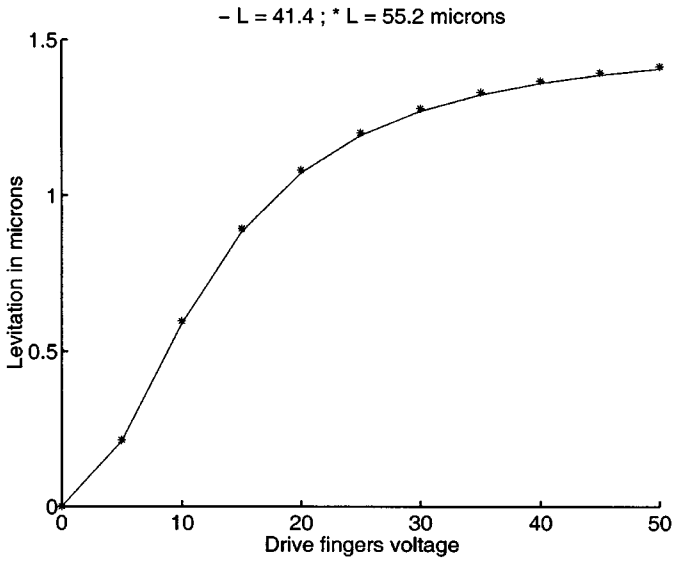

Fig. 6. Levitation.

because $\partial P/\partial q$ is sparse, only local pressures are effected by local charges.

The above detailed examination of the coupled electromechanical problem unearths a general paradigm for solving coupled domain problems with "black-box" domain-specific solvers. If the single domain solvers are developed with the added functionality of efficiently computing results from input perturbations, then such solvers can easily be used to perform coupled domain analysis following the multilevel-Newton approach outlined above. In the finite-element mechanics case, the perturbations were computed using an implicit representation of the geometric sensitivity to pressure and, therefore, such methods are given the lengthy title *matrix-implicit Multilevel-Newton Methods*.

3) Analyzing a Micromachined Resonator: In order to demonstrate that the above techniques make it computationally feasible to simulate an entire resonator, we now present results from a matrix-implicit multilevel-Newton coupled electromechanical solver [30]. The program uses the precorrected-FFT accelerated integral equation solver [29] with planar triangular panels to compute the electrostatic forces. A finite-element, mixed rigid/elastic mechanical analysis program using 20 noded isoparametric brick elements [33] is used to compute displacments.

An 18-finger polysilicon resonator is shown in Fig. 5. In this resonator, the central shuttle is suspended by 400-μm-long folded beams with a uniform thickness of 1.94 μm and finger dimensions of 13.8 $\times$ 4.6 μm. The central shuttle and an underside fixed plate (not shown) were set to 0 V, and a drive voltage was applied to the right- and left-hand side combs (also not shown). For this example, the Young's modulus of the polysilicon was determined to be 150 MPa, and the poisson ratio was 0.3.

The effect of varying the separation of the suspension beams, shown as L in Fig. 5, on levitation was investigated using the coupled-domain solver. The results are plotted in Fig. 6, which shows levitation (motion normal to the substrate) as a function of applied comb drive voltage. Levitation in resonators is to be avoided, because raising the central shuttle causes a misalignment of the interdigitated fingers. This misalignment reduces the resulting electrostatic forces, and may also allow the central shuttle to twist and collide with the side combs. The simulation results plotted in Fig. 6 show that levitation in the resonator can be nearly as large as the resonator thickness, and that changing separation L of the suspension beam inperceptibly effects levitation. The simulation was run on a Sun Ultra 30, and each load step required 70 minutes of CPU time.

B. Model-Order Reduction

Many micromachined devices are nonlinear, and extracting dynamically accurate nonlinear macromodels from simulation is a relatively open problem. For this reason, there has been much current interest in developing nonlinear model-order reduction strategies [35]–[37]. To better describe the challenges in nonlinear model-reduction, consider simulating the dynamics of a fixed-fixed beam in a fluid (air). Fig. 7 [38] shows the front view of the structure. When a voltage is applied, the flexible top plate deforms downward due to the generated electrostatic force, and the squeezed air in the gap damps the plate motion through a back pressure force. The exact deformation of the top plate due to the applied voltage is sensitive to the ambient pressure of the air, so this structure can be used as a pressure sensor. This pressure sensor also exhibits a very nonlinear response characteristic known as "pull-in." If the top plate deforms downward sufficiently, the electrostatic force will increase rapidly enough with additional downward motion to create a positive-feedback that forces the plate all the way down to the substrate.

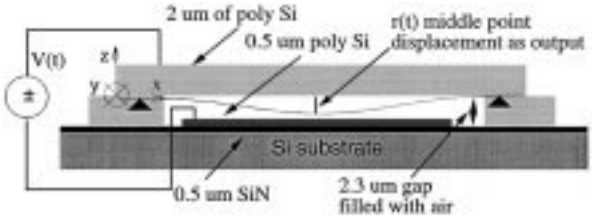

Fig. 7. The fixed-fixed beam structure. The x and z axes are parallel to the length and width of the beam, respectively, and the y axis points into the page.

Following Hung et al. [38], the dynamic behavior of this coupled electro-mechanical-fluid system can be modeled with the 1D Euler beam equation (17) and the two-dimensional (2-D) Reynold's squeeze film damping equation (18)

$$EI\frac{\partial^4 u}{\partial x^4} - S\frac{\partial^2 u}{\partial x^2} = F_{\text{elec}} + \int_0^w (p - p_a)\,dz - \rho\frac{\partial^2 u}{\partial t^2} \quad (17)$$

$$\nabla\cdot(u^3 p\nabla p) = \frac{12\mu}{1+6K}\frac{\partial(pu)}{\partial t} \quad (18)$$

where

x, y and z	as shown in Fig. 7;
E	Young's modulus;
I	moment of inertia;
S	stress coefficient;
ρ	density;
p_a	ambient pressure;
μ	viscosity;
K	Knudsen number;
w	width of the beam in the y direction;
$u = u(x,t)$	height of the beam above the substrate;
$p(x,y,t)$	pressure distribution in the fluid.

Finally, the electrostatic force is approximated assuming nearly parallel plates and is given by $F_{\text{elec}} \cong -(\epsilon_0 w V^2/2u^2)$ where V is the applied voltage.

Spatial discretization of (17) and (18) leads to a large nonlinear system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{b}^{\mathbf{T}}\mathbf{u}(\mathbf{t}) \quad y = \mathbf{c}^{\mathbf{T}}\mathbf{x} \quad (19)$$

where $\mathbf{x}$ is an n-length state vector, in this case the vector of displacements and their time derivatives. The function $\mathbf{f}$, which maps an n-length vector to an n-length vector, represents the spatially discretized partial differential equation. The above system with a nonlinear state equation is referred to as the "original" system which will be reduced to a much smaller system. The applied voltage generates $u(t)$, the input of the system. The output of the system is $y(t)$, and is chosen to be the beam's center point displacement.

1) Numerical Model Reduction: The goal of numerical model-order reduction is to generate a model with many fewer than n states which still preserves the input/output behavior of the original system. Almost all the numerical model-order reduction strategies are based on a change of variables

$$V_q x_q = x \quad (20)$$

where x_q is a q-length vector (q is assumed much much less than n), and V_q is $n \times q$ orthonormal matrix whose columns represent important "modes". Then, the matrix V_q represents a transformation from the original to the reduced coordinate system. Substituting the change of variables in (19) and multiplying the resulting equation by V_q^T yields

$$\dot{x_q} = V_q^T f(V_q x_q) + V_q^T b u(t) \quad y = c^T V_q x_q. \quad (21)$$

It should be noted that the dynamical system could have been multiplied by a second transformation matrix, U_q, leading to a wider range of algorithms.

Buried in (21) are the two key model-order reduction issues. First, one must select a good change of variables so that the input/output behavior is captured by the q states in the reduced system. Second, and perhaps less obvious, one must have a representation of $V_q^T f(V_q\cdot)$ that can be efficiently stored and evaluated. For example, suppose $n = 100\,000$ and $q = 10$. Then computing $V_q^T f(V_q x_q)$ explicitly would require on the order of 100 000 operations, and that hardly satisfies the efficiency goal of model-order reduction. If f were linear, so that $f(x) = Ax$ where A is an $n \times n$ matrix, then the representation problem is easily solved. To see this, consider that for the linear case

$$f_q(x_q) \equiv V_q^T f(V_q x_q) = V_q^T A V_q x_q = A_q x_q \quad (22)$$

where $f_q(\cdot)$ is a function which maps a q-length vector to a q-length vector and denotes the general nonlinear representation of the reduced model in the reduced variables. The matrix A_q is a representation of the reduced model which can be used when the problem is linear. As is clear from the equations, A_q is an easily computed $q \times q$ matrix. For the example numbers above, using the A_q matrix representation to compute $V_q^T f(V_q x_q)$ costs only 100 operations instead of order 100 000.

Returning to the first issue, selecting the change of variables or equivalently chosing V_q, there are a number of methods. If the problem is linear, the methods for determining V_q include: examining Krylov-subspaces [39], [22], [40], selecting from orthogonalized time-series data [35], or computing singular vectors of the underlying differential equation Hankel operator [41]. The approach based on using time-series data extends directly to the nonlinear cases, and the Krylov-subspace and Hankel operator approaches can be extended to the nonlinear case by linearizing the system only for the purpose of computing V_q, and then applying the change of variables to the original nonlinear system. As shown in [37], linearization approaches can be ineffective and better strategies may exist.

Regardless of how the V_qs are computed, for nonlinear problems there is still the difficulty of finding an efficient representation for $V_q^T f(V_q\cdot)$. One approach is to assume the reduced model is a multidimensional quadratic [43], [42], in which case

$$V_q^T f(V_q x_q) = f_q(x_q) \approx J_q^{(1)} x_q + x_q^T J_q^{(2)} x_q \quad (23)$$

where $J_q^{(1)}$ is the $q \times q$ Jacobian, or first derivative, of f_q and $J_q^{(2)}$ is a $q \times q \times q$ second derivative of f_q. Both $J_q^{(1)}$ and $J_q^{(2)}$ are easily computed from f by finite-differences, though q^2 function evaluations are needed to evaluate $J_q^{(2)}$ [42]. If higher order nonlinearities are required, such as cubic or quartic terms, the above strategy becomes computationally ineffective. The difficulty arises from the fact that there are q^{k+1} entries in the kth derivative of f_q, so generating a tenth-order reduced-order

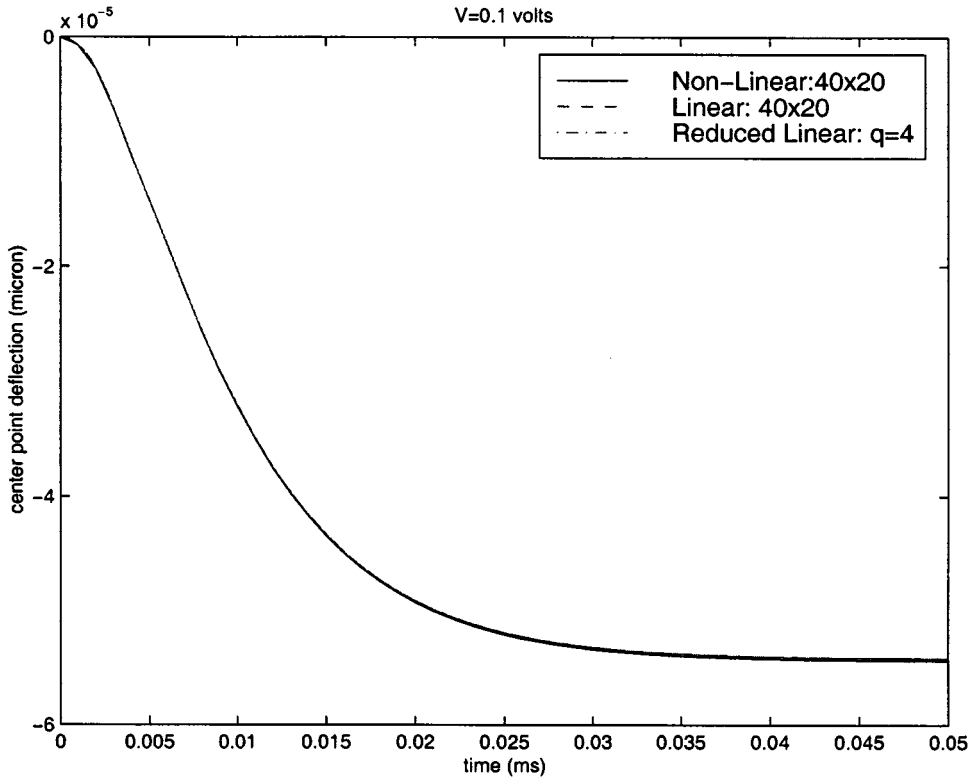

Fig. 8. A comparison of the time responses of the nonlinear model, linear model and fourth-order reduced model at small voltage input $V = 0.1$ V. The center point displacement is plotted against time.

model and including all the quartic terms requires a representation with over 100 000 entries. It is possible to use heuristics to prune the higher order nonlinearities, so that only a small fraction of the q^{k+1} terms are retained. Equivalently, the problem is one of determining a sparse representation for $J_q^{(k)}$.

An alternative view of the nonlinear model reduction problem can be developed for the case where the original nonlinear function, f, can be represented as the gradient of a scalar function [45]. That is

$$f(x) = \nabla_x \phi(x) \tag{24}$$

where ϕ maps an n-length vector to a scalar. Such representations occur naturally for second-order energy-conserving systems

$$\ddot{x} = \nabla_x \phi(x) + b^T u(t) \tag{25}$$

where $\phi(x)$ is derived by constructing the system's associated Hamiltonian. For such systems, the representation problem can be reduced to a single fitting problem by noting that

$$f_q(x_q) = V_q^T f(V_q x_q) = V_q^T \nabla_x \phi(V_q x_q) = \nabla_{x_q} \phi_q(x_q) \tag{26}$$

where ϕ_q maps an q-length vector to a scalar. Then, the scalar function of q variables, ϕ_q, can be approximately represented using a q-dimensional kth-order polynomial

$$\phi_q(x_q) \approx \sum_{i_1=0}^{R_1} \cdots \sum_{i_k=0}^{R_k} a_{i_1,\ldots i_k} x_{q_1}^{i_1} \cdots x_{q_q}^{i_k}. \tag{27}$$

If one represents f directly using derivatives, as in the previous paragraph, there are order q^{k+1} terms in the reduced model. Using a polynomial to represent ϕ up to order k requires only order q^k terms, so it would seem that exploiting $f = \nabla\phi$ results in a saving of only a factor of q. However, one can relatively easily fit the scalar function ϕ with a ratio of polynomials as in

$$\phi_q(x_q) \approx \frac{\sum_{i_1=0}^{R_1} \cdots \sum_{i_k=0}^{R_k} a_{i_1,\ldots i_k} x_{q_1}^{i_1} \cdots x_{q_q}^{i_k}}{\sum_{i_1=0}^{R_1} \cdots \sum_{i_k=0}^{R_k} b_{i_1,\ldots i_k} x_{q_1}^{i_1} \cdots x_{q_q}^{i_k}} \tag{28}$$

and such rational function representations can be effectively much higher order than k without the commersurate increase in cost [45].

2) Clamped Beam Example: We now present the results of comparing the reduced-order models generated using the Arnoldi method and a finite difference solution of the original nonlinear governing equations which is provided by Hung [38]. Hung verified this nonlinear solution with experimental data.

The air-gap in a 2-D representation of the pressure sensor was spatially discretized using a 40 $\times$ 20 mesh. Linearization was used to generate an eight-hundred-eightieth-order system of ordinary time-dependent equations. In the results below, we compare the time history of the displacement of the center point of the beam at different step input voltages: 0.1, 2, and 9 V. Three models are compared, namely, the full finite difference solution of the original nonlinear model of equations, the linearized system of equations, and a fourth-order reduced model generated using a Krylov-subspace method for selecting V_q. Fig. 8 shows that for a small input voltage $V = 0.1$ V the three curves representing solutions of each of the three models overlap with

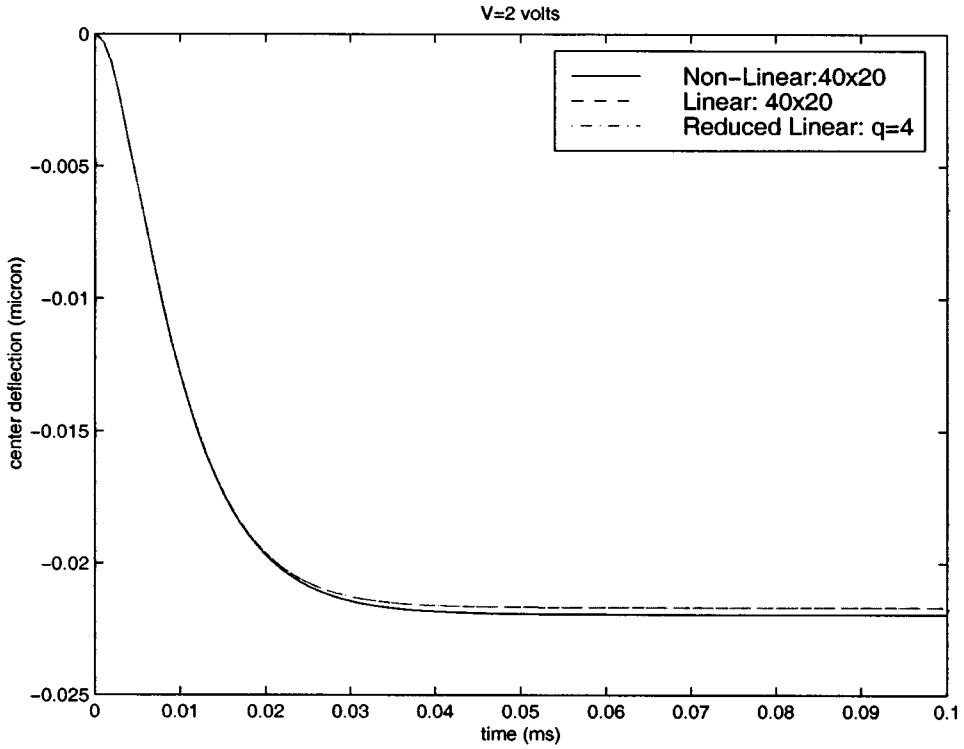

Fig. 9. At $V = 2$ V, the linear model starts to deviate from the nonlinear model. The reduced-order model still follows the linear model.

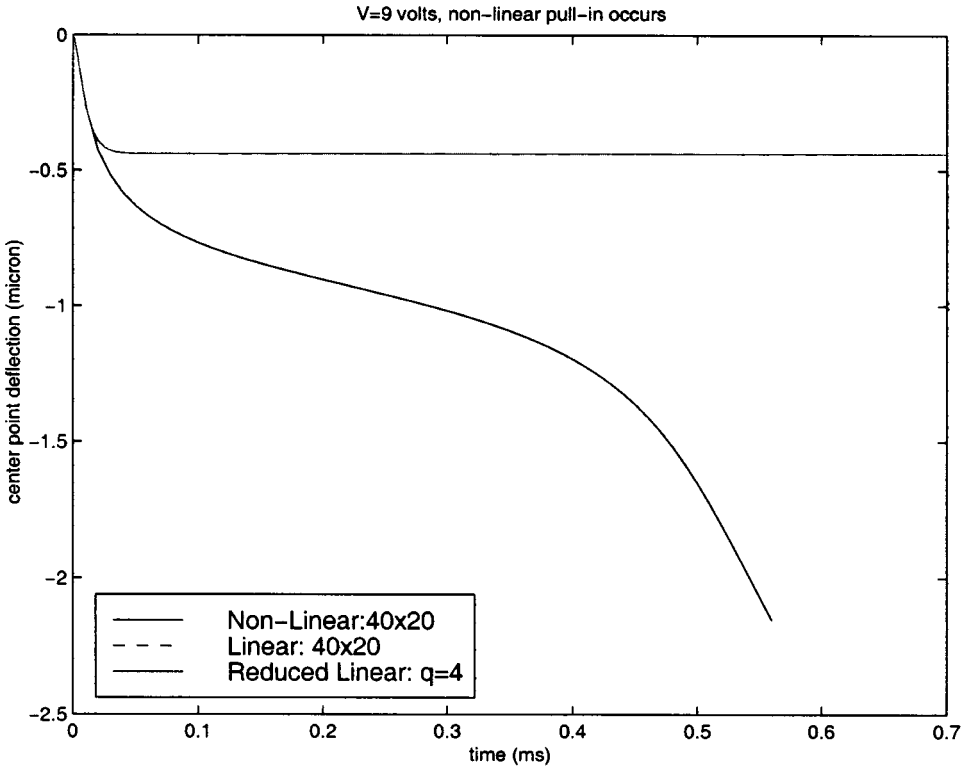

Fig. 10. A comparison of the time responses of the nonlinear model, linear model and fourth-order reduced model at small voltage input $V = 0.1$ V. The center point deflection is plotted against time.

one another. The Fig. 8 shows that with such a small input voltage the original system behaves almost perfectly linearly, and that the fourth-order reduced model faithfully reproduces the behavior of the eight-hundred-eightieth-order linear system.

In Fig. 9, we see that the linearized model starts to deviate from the nonlinear model, but the fourth-order reduced model still follows the linear model nearly exactly. Fig. 10 demonstrates that at the pull-in voltage, the time response of the struc-

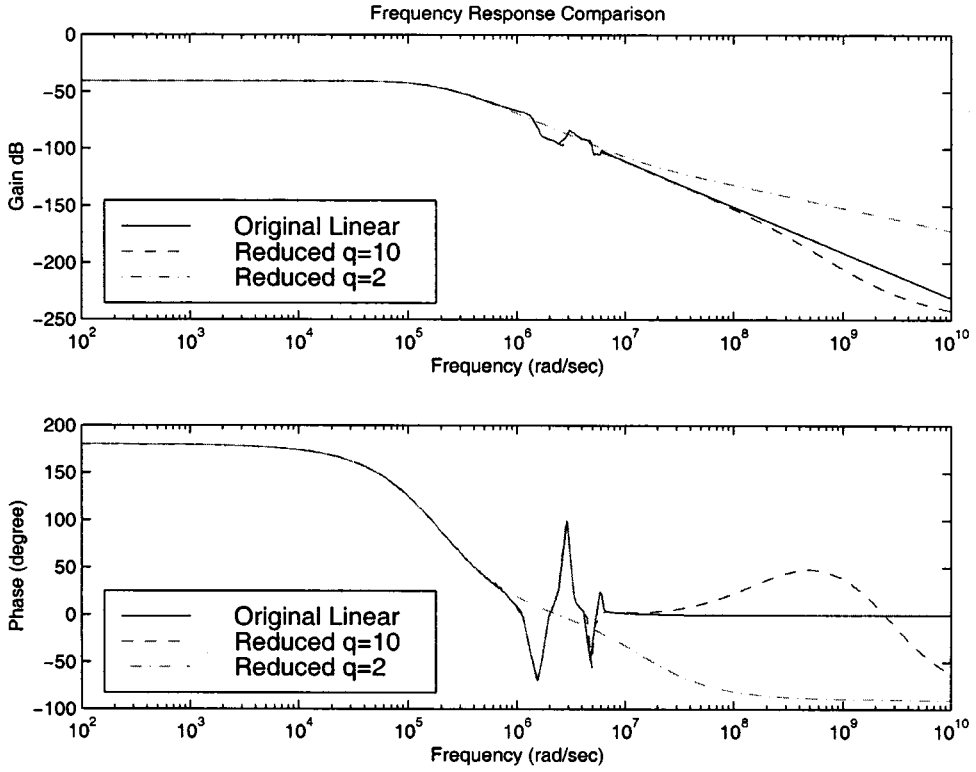

Fig. 11. A comparison of the frequency responses of the full linear system, second-order reduced model and tenth-order reduced model.

ture is extremely nonlinear. The linear model and the fourth-order macro-model are only accurate during the initial part of the transient.

For a discretization size of 10×5, which is in turn a seventieth-order linear system, we compared the frequency responses of the linear model and various orders of Arnoldi-based macro-models. Fig. 11 shows the comparison of the frequency responses of the large linear system and two macro-models that are of the order of two and ten, respectively. We see from Fig. 11 that the original linear system is a well damped system. The original linear system has a bandwidth frequency of 1.8×10^5.

Fig. 11 also shows that the second-order macro-model perfectly matches the linear model in a low frequency range up to 10^6 Hz. And the tenth-order model is able to follow all the oscillatory behavior both in the gain plot and the phase plot. The frequency-domain accuracy of the tenth-order model would be important if the device where part of a feedback system.

V. A Circuit Representation for Micromachined Devices

The semi-analytical macromodeling approach can be quite effective for design synthesis and optimization, but given the macromodel has built-in assumptions about device behavior, the approach is not a very effective verification strategy. The numerical model-order reduction approach, even if the difficulties with nonlinearities were overcome, seems poorly suited to the synthesis and optimization phase of a design because the approach requires a complete layout of the device, and it yields no information about sensitivities to changes in design attributes. In this section, we will describe a third alternative to extending circuit simulation to include micromachined devices. An approach will be developed which comes much closer to providing a schematic-like description for micromachined devices, but at the cost of narrowing the range of micromachined devices which can be so treated. So in that sense, this circuit-like description intended to make simulation easier is also a step toward top-down or structured design methodologies [46]–[54].

Developing a circuit representation for micromachined devices involves determining the list of elements for the circuit representation, the model for each element, and the definition of the nature or discipline for the terminals of the elements. The goals in selecting the list, model and nature include design reuse and simulation accuracy. Element parameterization provides both, while supporting a wide class of micromachined device designs. Parameterization with both design attributes and process parameters (captured in the model technology file) allows process independent models that can be used to simulate devices in a variety of fabrication lines. A conservative Kirchhoffian network representation is used both for simulation accuracy, and for compatibility with electronics design. Signal-flow representations, commonly used for behavioral or system-level modeling, are more cumbersome to use for this application because they are based on unidirectional elements while mechanical and electrical components interact bidirectionally.

A. Element Hierarchy

Micromachining technology combines sacrificial etching with VLSI-style deposit, pattern and etch sequences to produce miniaturized mechanical components that are suspended, cavitied, hinged, or otherwise mounted. The circuit approach

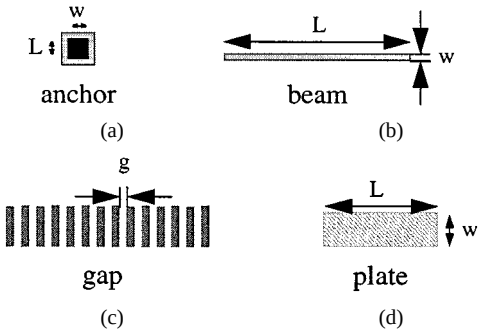

Fig. 12. Atomic elements for design of suspended micromachined systems include: (a) anchor, (b) beam, (c) gap, and (d) plate.

described below focuses primarily on suspended microstructural devices as that is the most mature micromachining design space. In principle, the circuit approach can be extended to include hinged structures for optical applications or cavitied structures for fluidic applications.

Suspended micromachined devices involve plates tethered by beams to anchors. Air gaps between conductive micromachined elements act as a variable sensing capacitance and a source for electrostatic actuation force. For example, plates can be actuated electrostatically to tilt micromirrors in digital computer projection displays [2] or in optical switches. Micromachined inertial sensors employ one or more plates as proof masses, which move when accelerated and whose motion is sensed capacitively [55]. Such suspended micromachined structures decompose into anchors, beams, plates and gaps, as shown parametrized in Fig. 12. This set of elements is chosen for three reasons: they all occur commonly (albeit sized by appropriate geometric parameters); they are modular (in the sense that they are decoupled from neighboring elements); and their behavior can be accurately approximated with a simple lumped parameter model. For the class of Manhattan-geometry suspended polysilicon devices, this set of four elements completely covers the possible design space, and forms a set of atomic basis elements in the circuit representation.

A circuit simulation environment for micromachined devices based on this element library with parametrized behavioral models, called NODAS (Nodal Design of Actuators and Sensors), has been developed [56], [57], [60]. Schematic examples of a crab-leg resonator and an O-shaped spring using the basis elements are shown in Fig. 13. The use of circuit element libraries for nodal analysis of micromachined systems [64]–[66] and for microgyroscope simulation [58], [67] are also being simultaneously pursued.

Circuit representation of additional elements at higher levels of the hierarchy may also be desirable, primarily because such elements aid in the capture of complex designs. In particular, the parameterized functional elements such as the linear comb-drive sensor or actuator, or the crab-leg spring, O-spring, or folded-flexure spring are easily re-used because they capture a single function (generate electrostatic force, provide mechanical stiffness, etc.) and hence can be accurately represented by behavioral models. While parametrized models at high levels of component abstraction are still possible, the fixed topology of these components limits their re-usability. Moreover, the large number of design variables in such abstractions significantly increases the complexity of generating a parametrized lumped-parameter model as detailed in Section III-B.

In addition to carefully choosing the elements in the design library to ensure richness of coverage of the design space, the terminals of each element needs to be carefully chosen to balance the need for interoperability between the elements and the accuracy and speed of design simulation. By using the same terminal natures at all levels of the design hierarchy, a composable design representation for mixed-level simulation is possible. This is particularly important as simulation of entire systems at the atomic level, though possible, may require unnecessary long simulation times. A library consisting of the most common atomic and functional elements, therefore, supports both rapid simulation as well as the capability to represent a wide class of designs.

As the underlying simulation representation is a Kirchhoffian network, the nature of the quantity defined *across* and *through* each branch in the network is very important. In the electrical domain, voltage across and current through a branch is the accepted standard. For mechanical domains, no standard nature exists. Two possible translational mechanical across-through relations are velocity-force and displacement-force. The latter representation is preferred in micromechanical design, as displacement is generally the most common observable state. Similarly, the rotational mechanical nature is angular displacement across and torque through a branch.

The associated reference directions of the mechanical terminals correspond directly to the physical directions of displacement and force. As with electrical circuit simulation, a consistent and systematic set of associated reference directions for mechanical terminals is essential. A simple convention specifies translational displacement across variables as positive in the positive-axis directions and the rotational displacement variables as positive in a counterclockwise rotation (right-hand rule) around the positive-axis directions. Through variables going into a branch are interpreted as positive-valued force or torque in the positive direction [60].

Once the terminal natures are defined, the element can be modeled by relating the flow through the terminals to the potential across the terminals. This model is often called a constitutive relationship in network theory. The models need to capture all the physics of the given element, hence a beam element needs to include mass, spring and damping physics, all parametrized by the beam design geometry and the process model parameters. Parametrized models that are within a few percent of continuum simulations have been derived using techniques described in Section III.

Mechanical parasitics need to be considered for accurate circuit simulation. For example, due to the lumped parameter modeling of the atomic elements, the joint between two beams in a flexure becomes a parasitic. The compliance of the joint is a fringing effect that can be modeled by extending the length of the beams incident at the joint. If one of the beams incident at the joint is significantly wider than the other beam, then the moment relations at the joint need to be considered. Extension factors and the use of plate joints have been verified by comparing the circuit simulation with continuum finite element simulation for

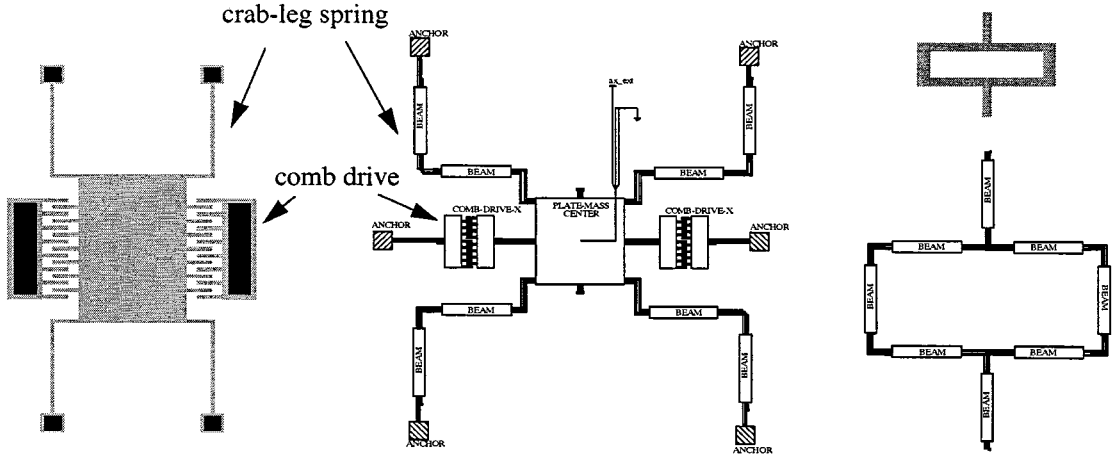

Fig. 13. (a) Layout and (b) schematic representation of an individual crab-leg microresonator. (c) Layout and (d) schematic representation of "O" coupling spring.

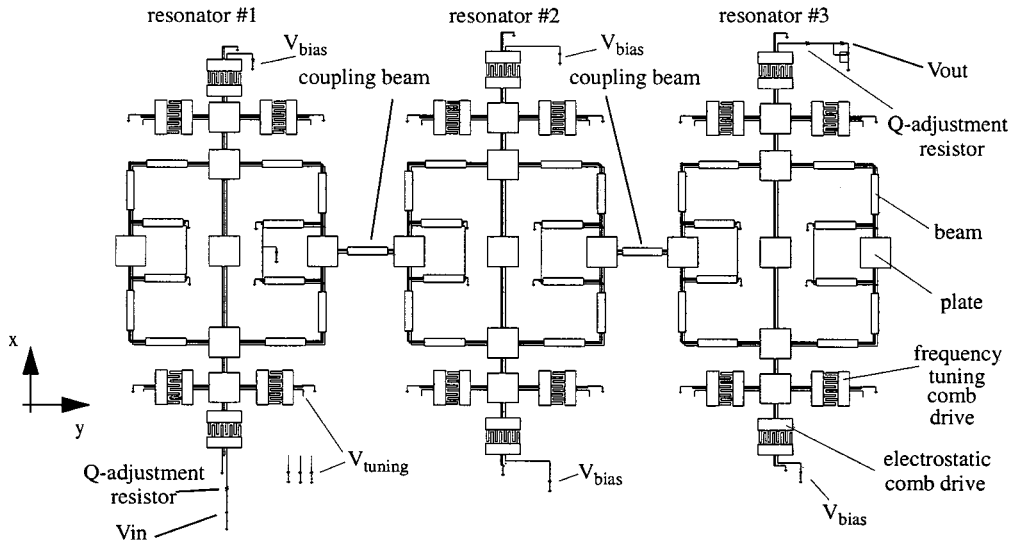

Fig. 14. Circuit Schematic of three resonator filter.

all the common flexure topologies and a range of beam sizes. In all cases, the error in flexure compliance and resonant frequency was less than 2%. Additional sources of parasitics include the capacitance and resistances in the interface between the microstructures and electronics.

B. Micromechanical Bandpass Filter Circuit

The flexibility of the microelelectromechanical circuit representation is best demonstrated by returning to the bandpass filter example. The filter is composed of three identical resonators, each with a center frequency of 300 kHz, coupled by beam springs [9]. The topology of the filter (Fig. 14) with both mechanical structures and interface circuitry is captured in the schematic using the symbols from the NODAS element library [68], [61]. The interface circuitry includes Q-adjustment, frequency tuning, and a trans-resistance sense amplifier.

When an ac input voltage, V_{in}, is applied across the electrostatic comb drive, the suspended shuttle masses and flexural beams will be driven by the electrostatic force and move in the x direction. This mechanical vibration is coupled to the other two resonators via the coupling beams, resulting in three resonant peaks, thus forming a passband. The location and spacing of the three peaks are determined by the stiffness of coupling beams, leading to different center frequency and bandwidth. An equivalent SPICE representation is derived in [9] and [68] using the methods described in Section III-B. The equivalent SPICE models represent the mechanical resonators as second-order systems of lumped parameters for mass, spring constant and damping, and represent the coupling beams as massless ideal springs with a coupling spring constant. Fig. 15 shows the result of simulating the filter in NODAS with massless beam models compared to the equivalent SPICE model in vacuum. The natural frequency of the resonators is 299.43 kHz, and the quality factor is 495 000. The coupling beams are 88.2 μm long and 1.12 μm wide. There are three peaks around the natural frequency, ranging from 299.43 kHz to 299.95 kHz. NODAS and SPICE results match to within 4%.

The peaks can be flattened to form a flat passband by applying Q-adjustment series resistors, shown by the simulated filter frequency response in Fig. 16. The three sharp peaks of the initial

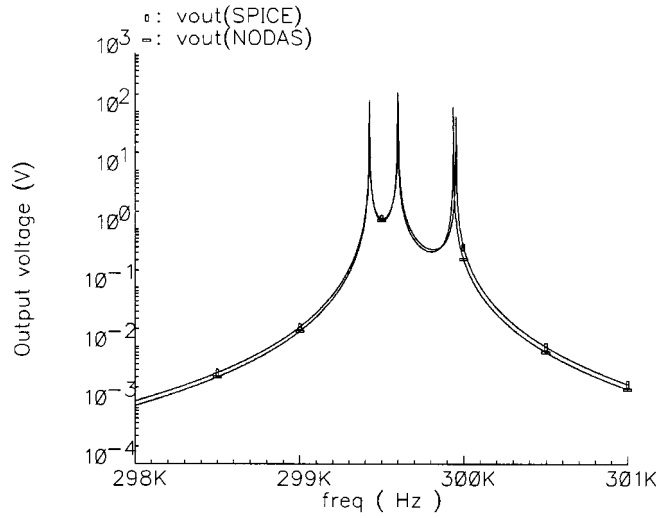

Fig. 15. Comparison of NODAS and equivalent SPICE frequency response.

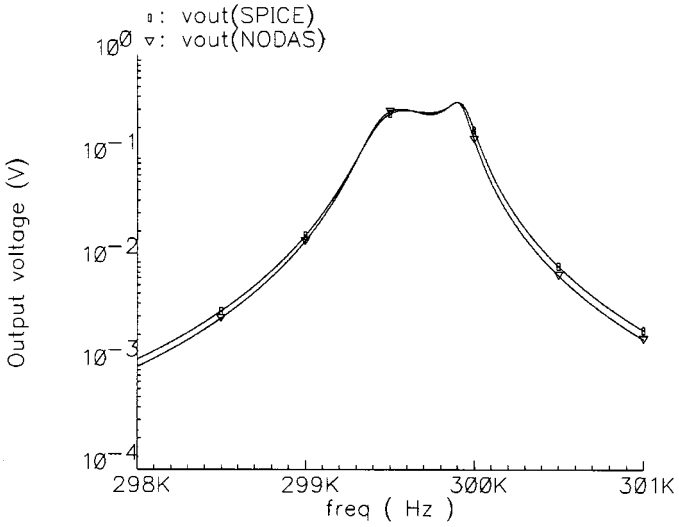

Fig. 16. Filter frequency response after Q-adjustment.

high-Q filter are now compressed down to a nearly flat passband with a ripple of -21 dB (Q of 587). NODAS and SPICE simulation results match to within 4%.

The actual flexure and coupling beams have finite mass, and can not be treated as ideal springs, a simplification needed for the analytical derivation of the equivalent SPICE model. Flexure beam mass shifts the center frequency of the filter, while coupling beam mass contributes to the lumped parameter equivalent masses of adjacent resonators also shifts resonant frequencies as well as causes passband distortion. This combined effect can be quantified by comparing the frequency response using the default NODAS beam model (with finite mass) with the response using a massless beam model, as shown in Fig. 17.

The combination of ease of schematic entry, simulation accuracy, applicability in iterative design of the wide class of suspended microsystems, compatibility with existing VLSI design flows, and support for co-simulation of electronic and micromachined devices make this micromecahnical circuit simulation approach very attractive. To expand this circuit approach, continued research is needed in identifying the basis elements for enlarged design spaces that include cavitied and hinged structures. Terminal natures for additional physics such as fluidic pressure and flow rate or optical beam intensity are also needed to model new classes of devices. Additionally, methodologies and tools for automated extraction of geometric and material parameters for accurate simulation are crucial for the wide applicability of this simulation-based design approach.

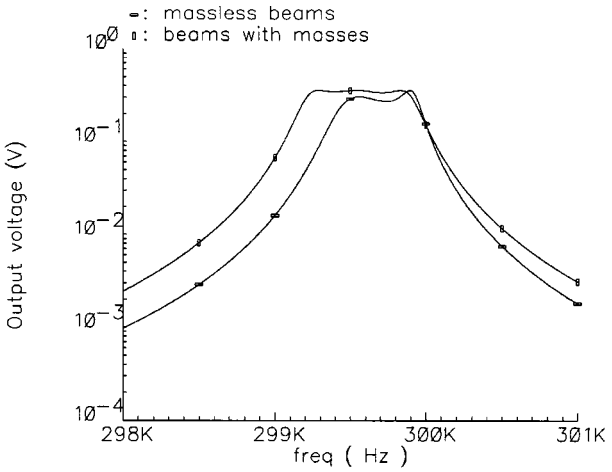

Fig. 17. Finite mass effect in frequency response.

C. Extraction from Layout

The circuit-like representation for micromachined devices fits perfectly with the synthesize and optimize phase of device design, as device performance can be simulated but layout details can be avoided. For this representation to be useful during the verification phase, it must be possible to extract the circuit from the layout. Just like for mainstream integrated circuits, layout extraction involves recognizing patterns in the layout and then inferring a one-to-one correspondence between the layout patterns and the circuit elements.

Layout extraction involves recognition of the layout patterns that correspond to the circuit schematic elements based on their features (shape, size, location). Once the schematic elements are recognized, the extraction creates a connected schematic to capture the shape and location, and annotates the element sizes, thus creating a complete schematic representation of the microstructure layout. The elements can be extracted as fixed values (e.g., plate has 1-μg mass), or as geometrical parameters (e.g., square plate has length of 100 μm). The abstractions used for mixed-domain circuit simulation are based on geometrically parametrized models of the atomic elements, requiring extraction of geometrical parameters from the layout. This approach is similar to device extraction in VLSI, where geometrical parameters for the MOS model are extracted from the layout. Unlike VLSI layout extraction, however, the features (shape, size and position) of each layout rectangle are of utmost importance in recognizing the constitutive micromachined atomic elements (VLSI extractors would consider a sequence of beams forming a suspention to be a single wire).

Once the constitutive atomic elements are recognized, element-specific extraction can be used as necessary. This procedure involves purely geometrical reasoning to identify the con-

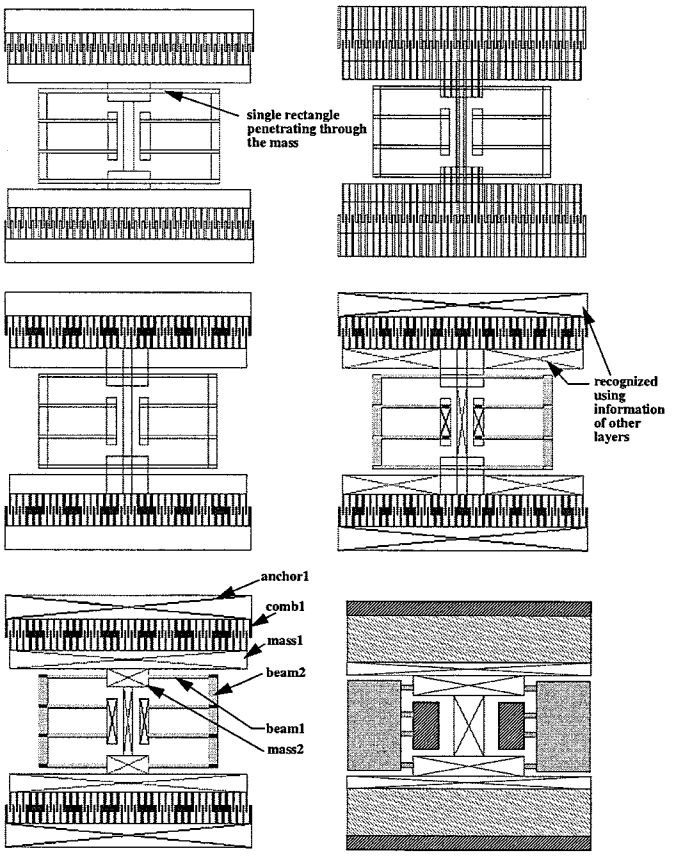

Fig. 18. Folded flexure resonator (a) layout, (b) canonical representation, (c) canonical representation after separating the fingers, (d) intermediate state, (e) detected state, and (f) functional elements detected.

stitutive schematic elements, followed by determining the appropriate parameters for each instance of an atomic or functional element [63] found in the layout. In addition to extracting a schematic representation from the layout, parasitics can also be identified and extracted [53], [54].

Rectangles that comprise the layout are generated by algorithms specific to the layout editing tools. The first step in feature recognition for Manhattan-geometry structures, therefore, involves creating a unique representation of the layout. Starting from an input layout in CIF [Caltech Interchange Form, shown in Fig. 18(a)], the rectangles in the layout are partitioned into a canonical representation, such that each rectangle (or cell) has only one neighbor on each side [shown in Fig. 18(b)]q. The use of the canonical representation allows the development of algorithms that are independent of the CAD software used to generate the input layout. The disadvantage of the canonical representation is a significant increase in the number of rectangles to be processed. Most of this increase comes from the presence of fingers in the design, hence they are removed, and the layout re-canonized, as shown in Fig. 18(c). The functionality of each of the cells is then determined by its shape, size and connectivity. Nonstructural mask layers (such as those that define anchors) are used to obtain hints for possible functional uses for each of the cells, using rules from a process description file. Also contained in this file are rules for atomic element recognition, for example, cells with one connected side are cantilever beam fingers, and cells with connections on opposing sides are considered to be beams. The partitioning due to the canonical representation algorithm results in multiple adjacent cells performing the same function. These multiple cells have to be combined to minimize the number of unnecessary nodes in the netlist. Cell merging, first in the horizontal direction, and then in the vertical direction accomplishes this for the mass and anchor cells. This merging reduces the total number of ports in the generated netlist, hence contributes to the management of the simulation time for the extracted netlist. The resulting netlist directly corresponds to the atomic elements (beams, plates, gaps and anchors) as shown in Fig. 18(e) in the circuit representation of Section V-A. Thus the primary objective of having a check on the designed layout can be achieved. Device function can also be confirmed via the "circuit" simulation in Section V-B on this netlist.

Higher level functional element models can be detected by processing the extracted netlist. A functional element library containing rules for detecting commonly used spring suspensions and comb-drive topologies is external to the extraction tool, and can be customized to alternate processes and design styles. Finger orientation, region of occurrence, and geometrical parameters (length, width, and interfinger gap) are used to partition the set of recognized fingers, which are then analyzed for connectivity resulting in the extracted comb-drives. Spring detection is accomplished via a finite state machine (FSM)-based algorithm. Starting from a start state (always an anchor atomic element), the type of beam and joint determines transitions into the intermediate states, and onto the final state, which indicates the type of spring detected. The joint transitions are classified according to the number of ports and the direction of rotation, and provide the fundamental abstraction on which this FSM-based detection works. The FSM for each of the springs is described in the component library. The connected sets of beams and springs obtained after the atomic recognition is passed through each of these FSMs to recognize their type, as shown by the example in Fig. 18(f). Simulation-based verification using this level of extraction is an order of magnitude faster than at the atomic element level, and is seen to be crucial for an iterative design methodology.

The challenge for the extraction methodology is to provide a rich set of basic recognition functions and a language for combining these functions in both process-independent element recognition, and process-dependent use of layer information to support recognition and extraction of all layouts that can be mapped to the circuit element library. This has yet to be demonstrated for non-Manhattan geometry structures in single-structural layer polysilicon micromachining processes. However, for the commonly used Manhattan-geometry suspended microstructure design style and polysilicon micromachining process, extraction has been extremely effective in microstructure layout verification.

VI. Conclusion

In this survey paper, we presented and contrasted three different approaches for extending circuit simulation to include micromachined devices. The most commonly used method, that of using physical insight to develop parameterized macromodels, is presented first. The issues associated with fitting the parameters to simulation data while incorporating design

attribute dependencies were shown to require sophisticated intervention. In addition, the semi-analytic approach did not seem to provide a very effective verification path. Then, the numerical model order reduction approach to macromodeling was presented, and it was shown that the key difficulty remains finding automatic methods for performing nonlinear model reduction. In addition, model-order reduction seemed to be ineffective during the synthesis and optimization phase of design, because no attribute sensitivities were computed. Finally, we described the recently developed circuit-based approach for simulating micromachined devices, and described the design hierarchy and the use of a catalog of parts. We also showed that the circuit-based approach can be combined with extraction from layout, providing an effective approach to verification. The only short-coming of this circuit-based approach is that only some design styles and technology can be supported.

REFERENCES

[1] C. Lu, M. Lemkin, and B. E. Boser, "A monolithic surface micromachined accelerometer with digital output," presented at the IEEE Int. Solid-State Circuits Conf., San Francisco, CA, Feb. 1995.

[2] P. F. Van Kessel, L. J. Hornbeck, R. E. Meier, and M. R. Douglass, "A MEMS-based projection display," in *IEEE Proc.*, Aug. 1998, pp. 1687–1704.

[3] A. V. Chavan and K. D. Wise, "A monolithic fully-integrated ncuum-sealed CMOS pressure sensor," in *Proc. 13th IEEE Int. Conf. Micro Electro Mechanical Systems (MEMS'00)*, Miyazaki, Japan, Jan. 23–27, 2000, pp. 341–346.

[4] C. T.-C. Nguyen, "MEMS for wireless communications," presented at the IEEE Micro Electro Mechanical Systems Workshop, Heidelberg, Germany, 1998.

[5] D. J. Harrison and P. G. Glavina, "Toward miniaturized electrophoresis and chemical analysis systems on silicon: An alternative to chemical sensors," *Sens. Actuators B*, vol. 10, 1993.

[6] A. H. Epstein *et al.*, presented at the Sensors and Actuators (Power MEMS and Microengines), Chicago, IL, June 1997.

[7] J. Voldman, M. L. Gray, and M. A. Schmidt, "Microfabrication in biology and medicine," *Annu. Rev. Biomed. Eng.*, vol. 1, 1999.

[8] B. F. Romanowicz, *Methodology for the Modeling and Simulation of Microsystems*. Norwell, MA: Kluwer Academic, 1998.

[9] K. Wang and C. T.-C. Nguyen, "High-order micro-mechanical electronic filters," in *Proc. IEEE MEMS Workshop*, Nagoya, Japan, Jan. 26–30, 1997, pp. 25–30.

[10] P. M. Osterberg and S. D. Senturia, "Membuilder: An automated 3D solid-model construction program for microelectromechanical structures," presented at the Transducers'95, Stockholm, Sweden, June 1995.

[11] J. M. Funk, J. G. Korvink, J. Buhler, M. Bachtold, and H. Baltes, "SOLIDIS: A tool for microactuator simulation in 3-D," *J. Microelectromech. Syst.*, vol. 6, no. 1, 1997.

[12] D. Ramaswamy, N. Aluru, and J. White, "Fast coupled-domain, mixed-regime electromechanical simulation," in *Proc. Int. Conf. Solid-State Sensors and Actuators (Transducers'99)*, Sendai, Japan, June 1999, pp. 314–317.

[13] S. Crary and Y. Zhang, "CAEMEMS: An integrated computer-aided engineering workbench for micro-electro-mechanical systems," presented at the Proc. IEEE Micro Electro Mechanical Systems, Napa Valley, CA, Feb. 1990.

[14] S. Senturia, "CAD challenges for microsensors, microactuators, and microsystems," presented at the Proc. IEEE, vol. 86, Aug. 1998.

[15] Y.-H. Cho, B.-M. Kwak, A. P. Pisano, and R. T. Howe, "Viscous energy dissipation in laterally oscillating planar microstructures: A theoretical and experimental study," in *Proc. IEEE Micro Electro Mechanical Systems. An Investigation of Micro Structures, Sensors, Actuators, Machines and Systems*, 1993, Cat. no. 93CH3265-6, pp. 93–98.

[16] W. Ye, X. Wang, and J. White, "A fast Stokes solver for generalized flow problems," presented at the Int. Conf. Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Diego, CA, Mar. 2000.

[17] Y. Gianchandani and S. Crary, "Parameteric modeling of a microaccelerometer: Comparing I- and D-optimal design of experiments for finite-element analysis," *J. Microelectromech. Syst.*, vol. 7, no. 2, June 1998.

[18] S. D. Senturia, R. M. Harris, B. P. Johnson, S. Kim, K. Nabors, M. A. Shulman, and J. K. White, "A computer-aided design system for microelectromechanical systems (MEMCAD)," *IEEE J. Microelectromechanical Systems*, vol. 1, no. 1, pp. 3–13, Mar. 1992.

[19] J. R. Gilbert, P. M. Osterberg, R. M. Harris, D. O. Ouma, X. Cai, A. Pfajfer, J. White, and S. D. Senturia, "Implementation of a MEMCAD system for electrostatic and mechanical analysis of complex structures from mask descriptions," presented at the IEEE Micro Electro Mech. Syst., Fort Lauderdale, FL, Feb. 1993.

[20] G. M. Koppelman, "OYSTER, a three-dimensional structural simulator for microelectromechanical system design," *Sensors Actuators*, vol. 20, no. 1/2, 1989.

[21] M. Bachtold, J. G. Korvink, and H. Baltes, "The adaptive, multipole-accelerated BEM for the computation of electrostatic forces," in *Proc. CAD for MEMS*, Zurich, Germany, 1997, p. 14.

[22] E. J. Grimme, D. C. Sorensen, and P. Van Dooren, "Model reduction of state space systems via an implicitly restarted Lanczos method," *Numer. Algorithms*, vol. 12, no. 1, 2, pp. 1–31, May 1996.

[23] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 105–126, 1986.

[24] J. Barnes and P. Hut, "A hierarchical $O(N \log N)$ force-calculation algorithm," *Nature*, vol. 324, pp. 446–449, 1986.

[25] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. New York: Adam Hilger, 1988.

[26] V. Rokhlin, "Rapid solution of integral equation of classical potential theory," *J. Comput. Phys.*, vol. 60, pp. 187–207, 1985.

[27] K. Nabors and J. White, "Fastcap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1447–1459, Nov. 1991.

[28] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White, "Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory," *SIAM J. Sci. Statist. Comput.*, vol. 15, no. 3, pp. 713–735, 1994.

[29] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 1059–1072, Oct. 1997.

[30] D. Ramaswamy, N. Aluru, and J. White, "Fast coupled-domain, mixed-regime electromechanical simulation," in *Proc. Int Conf. Solid-State Sensors and Actuators (Transducers'99)*, Sendai, Japan, June 1999, pp. 314–317.

[31] L. Greengard and V. Rokhlin, "A new version of the fast multipole method for the Laplace equation in three dimensions," *Acta Numerica*, pp. 229–269, 1997.

[32] W. Hackbusch and Z. P. Nowak, "On the fast matrix multiplication in the boundary element method by panel clustering," *Numer. Math.*, vol. 54, pp. 463–491, 1989.

[33] K. J. Bathe, *Finite Element Procedures*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[34] N. R. Aluru and J. White, "A coupled numerical technique for self-consistent analysis of micro-electro-mechanical systems," in *Microelectromechanical Systems (MEMS), ASME Dynamic Systems, and Control (DSC) Series* New York, 1996, vol. 59, pp. 275–280.

[35] E. Huang, Y. Yang, and S. Senturia, "Low-order models for fast dynamical simulation of MEMS microstructures," in *IEEE Int. Conf. Solid State Sensors and Actuators (Transducers'97)*, vol. 2, Chicago, IL, June 1997, pp. 1101–1104.

[36] M. Varghese, V. Rabinovich, M. Kamon, J. White, and S. Senturia, "Reduced-order modeling of Lorentz force actuation with mode shapes," presented at the Int. Conf. Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Juan, Puerto Rico, Apr. 1999.

[37] J. Phillips, "Automated Extraction of Nonlinear Circuit Macromodels," Cadence, Inc., San Jose, CA, Tech. Rep., Dec. 1999.

[38] E. S. Hung, Y. J. Yang, and S. D. Senturia, "Low-order models for fast dynamical simulation of MEMS microstructure," in *Proc. IEEE Int. Conf. Solid-State Sensors and Actuators (Transducers'97)*, vol. 4, June 1997, pp. 1101–1104.

[39] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.

[40] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," presented at the IEEE Conf. Computer-Aided Design, San Jose, CA, 1997.
[41] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L^{∞}-error bounds," *Int. J. Contr.*, vol. 39, no. 6, pp. 1115–1193, 1984.
[42] Y. Chen and J. White, "A quadratic method for nonlinear model order reduction," presented at the Int. Conf. Modeling and Simulation of Microsystems, Semiconductors, Sensors, and Actuators, San Diego, CA, Mar. 2000.
[43] J. Chen and S. Kang, "Techniques for coupled circuit and micromechanical simulation," presented at the Int. Conf. Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, San Diego, CA, Mar. 2000.
[44] Y. Chen, "Model Order Reduction for Nonlinear Systems," M.S. thesis, Massachusetts Inst. Technol., Cambridge, MA, Sept. 1999.
[45] L. Gabbay, J. Mehner, and S. Senturia, "Computer-aided generation of nonlinear reduced-order dynamic macromodels: I. Non-stress-stiffened case," J. Microelectromechanical Systems, submitted for publication.
[46] E. K. Antonsson, "Structured design methods for MEMS," presented at the NSF Sponsored Workshop Structured Design Methods for MEMS, Nov. 12–15, 1995.
[47] N. R. L. Lo, E. C. Berg, S. R. Quakkelaar, J. N. Simon, M. Tachiki, H.-J. Lee, and K. S. J. Pister, "Parametrized layout synthesis, extraction, and SPICE simulation for MEMS," in *Proc. 1996 IEEE Inl. Symp. Circuits and Systems*, Atlanta, GA, May 12–15, 1996, pp. 481–484.
[48] J. M. Karam, B. Courtois, H. Boutamine, P. Drake, A. Poppe, V. Szekely, M. Rencz, K. Hofmann, and M. Glesner, "CAD and foundries for microsystems," in *Proc. 34th Design Automation Conf. (DAC'97)*, Anaheim, CA, June 9–13, 1997, pp. 674–679.
[49] T. Mukherjee and G. K. Fedder, "Structured design of microelectromechanical systems," in *Proc. 34th Design Automation Conf. (DAC'97)*, Anaheim, CA, June 1997, pp. 680–685.
[50] G. K. Fedder, "Structured design for integrated MEMS," in *Proc. IEEE MEMS'99*, Orlando, FL, Jan. 17–21, 1999, pp. 1–8.
[51] T. Mukherjee, G. K. Fedder, and R. D. Blanton, "Hierarchical design and test of integrated microsystems," *IEEE Design Test*, vol. 16, pp. 18–27, Oct.–Dec. 1999.
[52] N. Swart, "A design flow for micromachined electromechanical systems," *IEEE Design Test*, vol. 16, pp. 39–47, Oct.–Dec. 1999.
[53] G. K. Fedder, "Top-down design of MEMS," presented at the 2000 Int. Conf. Modeling and Simulation of Microsystems (MSM 2000), San Diego, CA, Mar. 27–29, 2000.
[54] T. Mukherjee, "CAD for integrated MEMS design," presented at the Design, Test Integration, and Packaging of MEMS/MOEMS (DTIP 2000), Paris, France, May 9–11, 2000.
[55] (1998) ADXL202 Accelerometer Data Sheet. Analog Devices, Inc., Norwood, MA 02062-9106. [Online]. Available: http://www.analog.com
[56] J. E. Vandemeer, M. S. Kranz, and G. K. Fedder, "Nodal simulation of suspended MEMS with multiple degrees of freedom," presented at the 1997 Int. Mechanical Engineering Congress and Exposition: The Winter Annu. Meeting ASME and 8th Symp. Microelectromechanical Systems, Dallas, TX, Nov. 16–21, 1997.
[57] ——, "Hierarchical representation and simulation of micromachined inertial sensors," presented at the 1998 Int. Conf. Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators (MSM'98), Santa Clara, CA, Apr. 6–8, 1998.
[58] G. Lorenz and R. Neul, "Network-type modeling of micromachined sensor systems," presented at the 1998 Int. Conf. Modeling and Simulation of Microsystems, Semiconductors, Sensors, and Actuators (MSM'98), Santa Clara, CA, Apr. 6–8, 1998.
[59] G. K. Fedder and Q. Jing, "NODAS 1.3—Nodal design of actuators and sensors," presented at the IEEE/VIUF Int. Workshop Behavioral Modeling and Simulation, Orlando, FL, Oct. 27–28, 1998.
[60] ——, "A hierarchical circuit-level design methodology for microelectromechanical systems," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1309–1315, Oct. 1999.
[61] Q. Jing, H. Luo, T. Mukherjee, L. R. Carley, and G. K. Fedder, "CMOS micromechanical bandpass filter design using a hierarchical MEMS circuit library," in *Proc. 13th IEEE Int. Conf. Micro Electro Mechanical Systems (MEMS'00)*, Miyazaki, Japan, Jan. 23–27, 2000, pp. 187–192.
[62] B. Baidya, S. K. Gupta, and T. Mukherjee, "Feature-recognition for MEMS extraction," in *Proc. 1998 ASME DETC Design Automation Conf.*, Atlanta, GA, Sept. 13–16, 1998, [CD-ROM].
[63] ——, "MEMS component extraction," in *Proc. 1999 Int. Conf. Modeling and Simulation of Microsystems (MSM'99)*, San Juan, Puerto Rico, Apr. 19–21, 1999, pp. 143–146.
[64] J. Clark, N. Zhou, S. Brown, and K. S. J. Pister, "Nodal analysis for MEMS simulation and design," presented at the 1998 Int.. Conf. Modeling and Simulation of Microsystems (MSM 98), Santa Clara, CA, Apr. 6–8, 1998.
[65] ——, "Fast, accurate MEMS simulation with SUGAR 0.4," presented at the Sensors and Actuator Workshop, Hilton Head Is., SC, June 1998.
[66] J. Clark, N. Zhou, and K. S. J. Pister, "Modified nodal analysis for MEMS with multi-energy domains," presented at the 2000 Int. Conf. Modeling and Simulation of Microsystems (MSM 2000), San Diego, CA, Mar. 27–29, 2000.
[67] D. Teegarden, G. Lorenz, and R. Neul, "How to model and simulate microgyroscope systems," *IEEE Spectrum. Mag.*, vol. 35, p. 66, July 1998.
[68] Q. Jing, T. Mukherjee, and G. K. Fedder, "A design methodology for micromechanical bandpass filters," presented at the 1999 IEEE/ACM Int. Workshop Behavioral Modeling and Simulation (BMAS 99), Orlando, FL, Oct. 4–6, 1999.

Tamal Mukherjee (S'89–M'95) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1987, 1990, and 1995, respectively.

Currently, he is a Research Engineer in Electrical and Computer Engineering Department at Carnegie Mellon University. His research interests include automating the design of analog circuits and microelectromechanical systems. His current work focuses on the developing computer-aided design methodologies and techniques for integrated microelectromechanical systems, and is involved in modeling, simulation, extraction, and synthesis of microelectromechanical systems.

Gary K. Fedder (S'93–M'93) received the B.S. and M.S. degrees from Massachusetts Institute of Technology, Cambridge, 1982 and 1984, respectively, and the Ph.D. degree from the University of California, . Berkeley, in 1994 all in electrical engineering and computer science.

He is an Associate Professor at Carnegie Mellon University holding a joint appointment with the ECE Department and The Robotics Institute.

D. Ramaswamy, photograph and biography not available at time of publication.

Jacob White (S'80-M'83) received the B.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, and the S.M. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley.

He is currently a Professor at Massachusetts Institute of Technology. He worked at the IBM T.J. Watson research center from 1985 to 1987 and was the Analog Devices Career Developement Assistant Professor at the Massachusetts Institute of Technology from 1987 to 1989. His research interests are in serial and parallel numerical algorithms for problems in circuit, interconnect, device, and microelectromechanical system design.

Dr. White was a 1988 Presidential Young Investigator, and was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS from 1992 until 1996 and was Technical Program Chair of the International Conference on Computer-Aided Design in 1998.