Modeling, Simulation, and Verification of an Advanced Micromirror Using SUGAR

Jason Vaughn Clark and Kristofer S. J. Pister

***Abstract*—We present some of the design, modeling, and simulation features of a computer-aided engineering tool for microelectromechanical systems (MEMS) called SUGAR. The features include a flexible SPICE-like netlist language for MEMS design, a simple modeling framework for computationally efficient lumped models, an extensible architecture to which users can add features, and the ability to display 3-D circuits together with deflected electromechanical structures. Since SUGAR is programmed in MATLAB, many MATLAB functions and toolboxes may be used with SUGAR. Such attributes facilitate the exploration of design spaces and feature modifications. In this paper, we describe SUGAR's extensible architecture, flexible design methodology, modeling framework, and reduced-order modeling technique. We do not present the many other advances made for SUGAR by other developers. For a test case, we choose an advanced microdevice that is difficult to simulate with conventional MEMS software. We show that the relative errors of our lumped models are less than 3% of the finite-element analysis (FEA), that the computational costs are less than 1% of the FEA, and that simulation of the test case fairly agrees with the experiment. [2006-0268]**

***Index Terms*—Design methodology, finite-element methods, modeling, simulation software, SPICE, system analysis and design.**

I. Introduction

During the last two decades, the field of microelectromechanical systems (MEMS) has gone from producing simple-function devices to producing more complex systems. This has led to the development and widespread use of computer-aided engineering (CAE) tools for MEMS. Although these tools have been successful in simulating the behavior of simple-function devices, they have not been as successful in simulating the behavior of more complex systems on a personal computer (PC) nor within a practical amount of time. The needs and challenges of CAE for MEMS have been previously discussed in [1] and [2]. In essence, depending on how well the CAE software facilitates the design process, reduces the time of computation, and agrees with reality, the software can be an invaluable aid for technological advancements in MEMS. With the ultimate goal of quickly and accurately simulating complex systems, we present efficient methods to configure, model, and simulate MEMS that are composed of a large number of lumped components. These methods are packaged in a CAE for MEMS tool called SUGAR [3].

Commercial CAE tools are well suited to synthesize and characterize sophisticated models of MEMS components through the use of finite-element analysis (FEA) [4]–[7]. With the availability of such tools, is there also a need for tools that use lumped models and take more of a network- or system-level approach? Three reasons for this need are as follows: First, there are a significant number of designers who use a common set of MEMS components in their designs. For instance, a partial set of commonly used components may comprise electromechanical beams, comb drives, plates, and anchors. Second, there are a significant number of simple linear components for which the relative error between their lumped analysis and FEA is less than the relative error between their experimental analysis and FEA. This is due to the uncertainties in structural geometries and material properties from variations in the fabrication process. Third, as the number of MEMS components or time steps increases, FEA may need more memory than accuracy allows or may consume an impractical amount of time.

These three reasons for a network/system-level approach—a common set of components, small relative errors, and timeliness of solutions—are addressed in SUGAR as follows: To address the first reason, we have developed a common set of parameterized models. Moreover, new lumped models such as those extracted from theory, experiment, or FEA can be incorporated into SUGAR. Regarding the second reason, we show that the relative errors of our lumped models are less than 3% of their FEA counterparts, i.e., within the nominal operating range. We also show that the performance of a complete system composed of a multitude of these models fairly agrees with the true performance. Such results represent a good cost/accuracy tradeoff. As for the third reason, we demonstrate that by using a SPICE-like netlist, the time to configure a system can be significantly reduced, and by using computationally efficient models, the time for computation can be significantly reduced. Such capabilities facilitate the simulation of more complex systems and may give designers insights that they would not otherwise have.

Recent FEA-based tools for MEMS such as Coventorware [5], Intellsense [6], and Softmems [24] are able to generate lumped behavioral models from FEA. Such features have been motivated by the use of network tools for simulating MEMS. A

Manuscript received December 2, 2006; revised July 20, 2007. This work was supported in part by the Defense Advanced Research Projects Agency. Subject Editor C. Liu.

J. V. Clark was with the Berkeley Sensor and Actuator Center and the Applied Science and Technology Graduate Group, University of California, Berkeley, CA 94720 USA. He is now with the School of Electrical and Computer Engineering and the School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: jvclark@purdue.edu).

K. S. J. Pister is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: pister@eecs.berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JMEMS.2007.904947

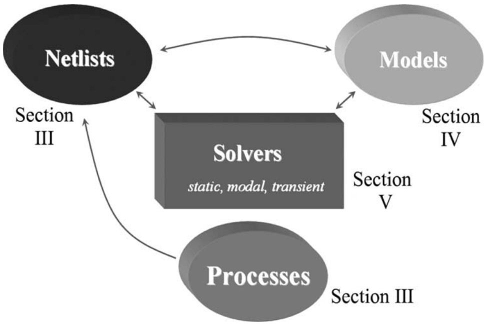


Fig. 1. Architecture of SUGAR and outline of this paper. The ellipses symbolize parameters and constitutive relationships of the system, the rectangle symbolizes solution algorithms that determine the system's response, and the arrows represent possible flow of information during the solution process. Experimental verification is discussed in Section V.

few examples are SUGAR [10]–[14], SPICE [9], NODAS [8], SABER [25], and SIMULINK [26], [27]. A survey paper is given by Mukherjee *et al.* [28]. As a network tool, features unique to SUGAR include its ease of extensibility and its animated 3-D graphics capabilities. Its programmability and seamless integration with MATLAB toolboxes are some of the benefits that invite innovative feature developments. For example, SUGAR has been used in graduate courses at Purdue, Berkeley, and Cornell, where new lumped models and features have been explored. In addition, researchers at the National Aeronautics and Space Administration have generated new MEMS designs by wrapping genetic algorithms around SUGAR routines [15].

We organized this paper as follows. In Section II, we present a high-level view of SUGAR's architecture, which illustrates its flexibility. In Section III, we describe our network configuration method and show how we represent our micromirror test case using a SUGAR netlist. In Section IV, we provide a brief overview of some of our modeling methods and compare the lumped components of the micromirror to FEA. In Section V, we verify our network methodology by comparing a static simulation of the test case to the experiment. Last, we discuss what such advancements might mean for the design, modeling, and simulation of complex MEMS. We use the CAE tools SUGAR-2.0 and COMSOL-3.2.

II. Architecture

MEMS are represented in SUGAR by decomposing the system into components of the linear and/or nonlinear lumped models. Then the components are assembled into a set of ordinary differential equations (ODEs). The architecture of SUGAR and the outline of this report are depicted in Fig. 1. The configuration of a system is entered into SUGAR in the form of a netlist text file. A netlist typically describes the following: which lumped models will be used, the network connectivity between the various models, and the parameters of the models, such as geometries and material properties. The netlist is converted within SUGAR into a system of ODEs by assembling state variable matrix coefficients and excitation vectors from the parameterized models. Following the static, modal, or transient solution, the deflected state(s) of the structure may be viewed in 3-D.

III. Network Configuration

In SUGAR, one configures a network of MEMS components by means of a netlist. This method for MEMS configuration in SUGAR is inspired by the method for integrated circuit configuration in SPICE. Compared to a SPICE netlist, a SUGAR netlist has greater utility because it is integrated into the MATLAB environment. For instance, MATLAB-generated arguments can be passed into a netlist data structure for processing. This processing involves converting the netlist text into a parsed MATLAB data structure. This data structure expedites processes such as assembling lumped components into a system, analyzing a solution, and displaying the state of the structure in 3-D.

Compared to the configuration methods used in the conventional CAE for MEMS packages, SUGAR's netlist language simplifies the configuration of complex structures. For instance, MEMS components branch off from one another in terms of geometry and orientation—global coordinates are not required. Moreover, MEMS components themselves may be composed of a system of components, i.e., a parameterized subnet. In this way, network- and system-level modeling become powerful aides in configuration and design.

Similar to SPICE, each component (or building block) in SUGAR can be expressed by a single line of netlist text that describes the model's name, its interfacing nodes, and its parameters, i.e.,

```
model name [node list] [parameter list].
```

External parameters and files are accommodated within the netlist. External files may include component libraries and material properties. Basic arithmetic expressions, variable node arrays, and loops are accommodated as well.

A. Test Case: Netlist Approach for a Micromirror

In this section, we show how we create a parameterized netlist of the micromirror shown in Fig. 2. We use this micromirror in the modeling, simulation, and verification sections of this paper. We choose this particular device because it is difficult to model by hand and difficult to simulate using FEA on a typical PC. For these reasons, we consider the successful configuration, modeling, simulation, and verification of such a device an adequate test case for arguing the benefits of our approach. We measure the success of our approach by computational efficiency and accuracy. First, we discuss the configuration details of the micromirror, and then we use these details to form a parameterized netlist.

B. Configuration Details

It is well known to those skilled in MEMS design that subtle changes in the design can significantly affect performance. For simulation to match performance, it is therefore necessary that such details be conveyed in the modeling and configuration of the device.

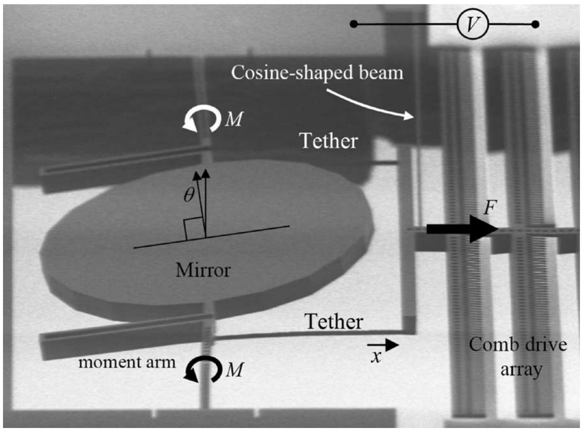


Fig. 2. Image of a micromirror during actuation [16]. The comb drive array converts an electric potential V into a mechanical force F that pulls the pair of tethers. The moment arm converts this translational force F into a moment M that rotates the circular mirror by θ. The other half of the comb drive array and the second cosine-shaped flexible beam are outside the view range of this figure. The difference in pigmentation of the structural material is an artifact of SEM imaging due to differing amounts of electric potential.

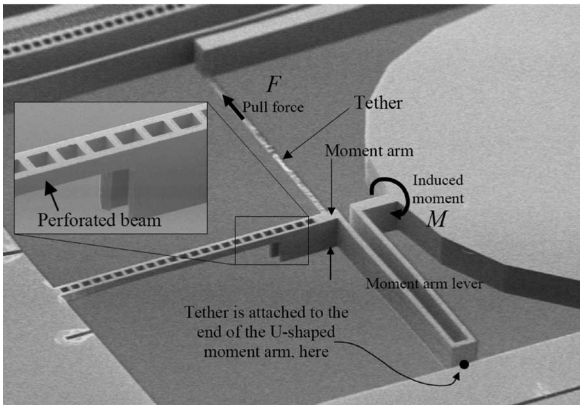


Fig. 3. Image of the moment arm assembly [16]. The perforated torsional hinge is formed out of the highest structural layer, the coarse tether is formed out of the lowest structural layer, and the rest of the device is formed out of the thick 50-μm bulk. The insert is a magnified view of the coupling between the perforated torsional hinge and the moment arm.

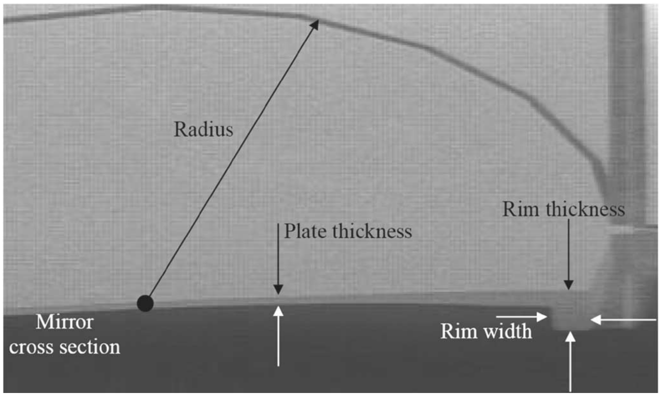


Fig. 4. Cross section of the circular mirror. The outer rim of the mirror has polygonal geometry due to geometrical limitations of the layout editor and/or mask-making tool. The plate thickness varies slightly due to variations in the reactive ion etch rate. In our model, we assume that the circumference is circular and that the plate thickness is constant.

The micromirror shown in Fig. 2 consists of several advanced features, i.e., a recessed circular mirror, a pair of perforated torsional hinges, a pair of cosine-shaped beams, three structural layers, a pair of translational-to-rotational transducers, and several hundred electrostatic comb drive fingers. The design objective of the micromirror project is to create a large-deflection fast-beam-steering micromirror for laser-based communication for SmartDust [16], [17]. Brief descriptions of its operation and components follow.

Operation of the Micromirror: As indicated in Fig. 2, a voltage V applied across the comb drives generates a net force $F \propto V^2$. This force pulls the pair of tethers and moment arms, which rotate the circular mirror out-of-plane by an angle θ. The moment arms couple the translating comb drive to the rotating mirror. As shown in Fig. 3, this couple is formed out of three structural layers. The highest structural layer comprises the membrane of the circular mirror and a pair of perforated torsional hinges that attach to either side of the mirror. The lowest layer consists of the pair of tethers only. The rest of the structure is formed out of the 50-μm-thick bulk, which includes the comb drive, various flexible beams, moment arm levers, and the rim of the mirror. We use the following descriptions to create a netlist and models.

Perforated Beam: The perforated beam indicated in Fig. 3 has several structural attributes. Compared to a solid unperforated beam with the same cross-sectional area, the perforated beam has a lower torsional stiffness. In addition, compared to a solid beam with the same torsional stiffness, the perforated beam has a higher lateral stiffness. Therefore, the perforated beam appears to be better suited as a torsional hinge over its solid beam counterpart. The perforated beam is also used as the structural support for the comb drive. In this case, the low mass of the perforated beam increases the frequency response while providing adequate lateral stiffness.

Moment Arm Assembly: The moment arm assembly is shown in Fig. 3. All three structural layers are used in its construction. The lowest layer (the tether) extends from the comb drive array to the low end of the moment arm lever. The highest layer (the perforated torsional beam) extends from the anchor to the high pivot point of the moment arm. These two layers are coupled through the bulk layer.

Circular Mirror: The large circular mirror of the device resembles the head of a drum, inasmuch as the upper membrane is formed out of the highest structural layer, and the rim is formed out of the bulk (see Fig. 4). The underside of the mirror is recessed to reduce the mass of the plate for a faster response.

Cosine-Shaped Beam: A pair of cosine-shaped beam flexures supports the comb drive array. A cosine-shaped beam is indicated in Fig. 2, where deflection has straightened its cosine shape. These predeformed beams help maintain comb drive stability during large fixed-guided deflections with high voltage. As the comb drive actuates, the cosine-shaped beams deflect laterally and tend to straighten. This straightening produces an effective beam-lengthening effect. Conversely, if these beams were straight before deflection, then they would become cosine-shaped upon the fixed-guided deflection. A beam-shortening effect would result along the axial direction, which causes undesirable asymmetry about the comb fingers.

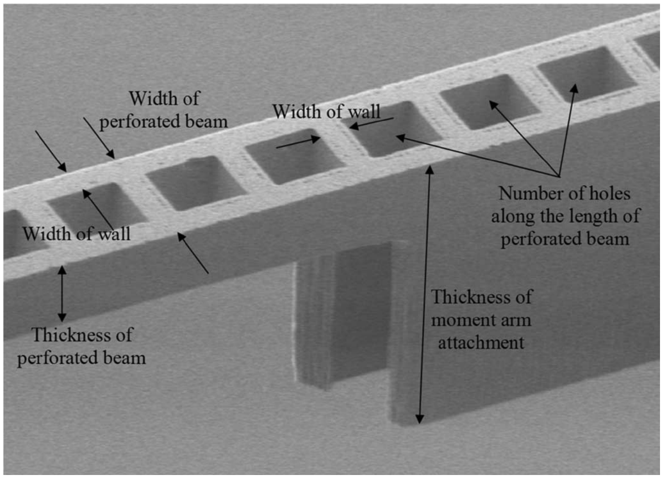

Width of perforated beam
Width of wall
Width of wall
Number of holes along the length of perforated beam
Thickness of moment arm attachment
Thickness of perforated beam

Fig. 5. Perforated beam and moment arm interface. Descriptions of the geometrical parameters of the model are indicated in the image. Material properties such as torsional stiffness, lateral stiffness, moment of inertia, and electrical resistance depend on such parameters.

C. Netlist Description

In this section, we create a netlist description of the micromirror by using the configuration details given above.

Component Usability: When creating netlist components, it is often advantageous to make them suitable for general use by other designers. For instance, Fig. 5 depicts a seemingly practical choice of parameters for the perforated beam.

Netlist: By inspecting Figs. 2–5, we choose to decompose the system into 21 netlist components. These components comprise eight distinct building blocks, which are depicted in Table I. The model names, parameters, and nodes are also given. Using these eight building blocks, we reconstruct the micromirror by assembling them together within a netlist. The node labels in the netlist are places where the building blocks are connected.

Fig. 6 shows a netlist of the micromirror along with its simulated 3-D image as displayed in SUGAR. In the netlist, components branch off from one another by way of their node labels, geometrical dimensions, and orientation. This branching method of construction simplifies subsequent modifications. For instance, as a consequence of changing the orientation of a single component, the relative positions of all attached components are changed accordingly. More details of the netlist syntax are given in the SUGAR manual [3]. However, to clarify parts of this netlist used in subsequent sections, there are three additional details that we describe, i.e., imported data, orientation parameters, and rigid links.

Imported Data: Data files such as model libraries and material properties can be imported into netlists. For instance, in Fig. 6, the data file mirror_process.net provides material properties (e.g., modulus of elasticity, electrical resistivity, density, residual stress, etc.) as well as subnets (e.g., perf_beam, moment_lever, etc.). Data may also be imported dynamically. For instance, in Section V, we examine a design space by sweeping two netlist parameters.

Orientation Parameters: The parameters oy, oz, and ox in the netlist (Fig. 6) are used to orient components. These parameters form an ordered sequence of rotations about the local axes of leading nodes. The result is equivalent to the transformation of coordinates by a direction cosine matrix. The parameters in our method are easily accessible, and the physical connections between the components are maintained during component rotations.

Rigid Links: A rigid link is an inertialess rigid moment arm that extends from a node. It is used to offset the position of a node. For example, the rigid links of a flexible beam are depicted in Fig. 7. In the present netlist, rigid links are used to facilitate electromechanical connections and proper positioning between the three nonplanar structural layers (see Fig. 8). Since the link is rigid, the designer is urged to use the link with caution. Preferably, rigid links may be used in structural regions where stiffness is approximately rigid.

Concluding this section, we presented an efficient method to configure a microdevice. The micromirror was decomposed into eight parameterized building blocks, where a line of netlist text represented each block. The success of our method can be seen by comparing the true images of the micromirror (Figs. 4 and 5) to those generated by the netlist (Figs. 8 and 9). Next, we overview some of our modeling methods, and develop and test the lumped models we use in the netlist shown in Fig. 6.

IV. Modeling

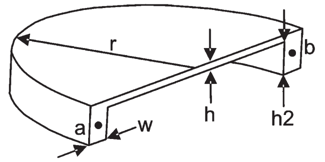
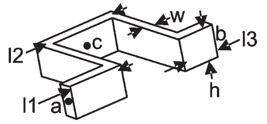
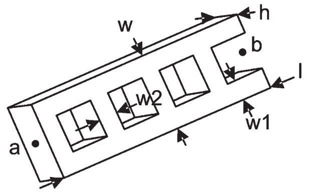
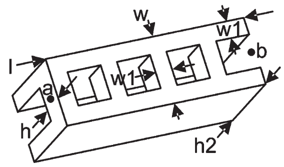
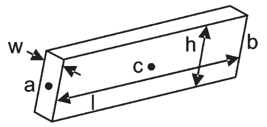
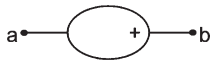
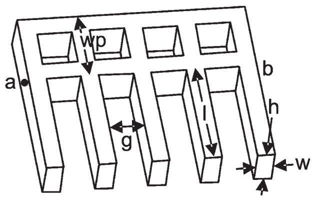
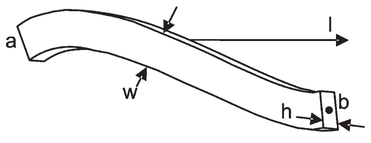
In the preceding section, we presented our netlist configuration method, which was fairly accurate and very efficient. In this section, we present our modeling method, which is fairly accurate and very efficient. We use computationally efficient lumped (or reduced-order) models rather than distributed element models. However, this efficiency often comes with tradeoffs due to modeling assumptions. For instance, the use of lumped models is often confined within particular operating ranges where the model is assumed to be valid. Moreover, there is often a lack of distributed information in the domain between the nodes of a lumped model. Typically, such modeling assumptions impose limits on geometrical dimensions, dynamics, material properties, and types of physical interactions. The analyst (or software) must be aware of such limitations. Fortunately, the limits imposed by the modeling assumptions often exceed the nominal operating ranges of many MEMS. In addition, there is a significant number of MEMS that can be decomposed into common sets of lumped electromechanical building blocks.

A. Modeling Framework

We construct a system of ODEs in SUGAR by assembling mathematical representations of lumped models that are prescribed by a netlist. After brief descriptions of the model function, system assembly, equation of motion, and graphics, we discuss several models for the micromirror test case. We discuss verification and simulation in Section V.

Model Function: Our model function is a MATLAB m-file that contains a mathematical description of a lumped model in the form of parameterized ODE coefficients and/or terms.

TABLE I
MICROMIRROR BUILDING BLOCKS

	Building block / Component	Model name, [node list]	Selected parameters
1		mirror (circular plate with rim) [a b]	r (radius) w (rim width) h (plate thickness) h2 (rim thickness)
2		moment_lever (moment arm lever) [a c b]	L1 (arm length) l2 (arm length) l3 (arm length) w (width) h (thickness)
3		perf_beam (perforated beam) [a b]	l (length) w (width) h (thickness) n (number of perforations) w1 (rail width) w2 (rung width)
4		perf_arm [a b]	l (length) w (width) h, h2 (thickness) n (number of perforations) w (main width) w1 (minor width)
5		beam [a b], & beamc (with center node) [a c b]	l (length) w (width) h (thickness)
6		V (voltage source) [a b]	V (voltage) l, w (cosmetic length, width)
7		perf_comb (perforated comb drive) [a b]	n (number of fingers) l (finger length) w (finger width) h (thickness) g (gap) wp (perforation width)
8		shaped_beam [a b]	l (length) w (width) h (thickness) rx1, ry1, rz1 (node-1 rotation) x2, y2, z2 (node-2 translation) rx2, ry2, rz2 (node-2 rotation)

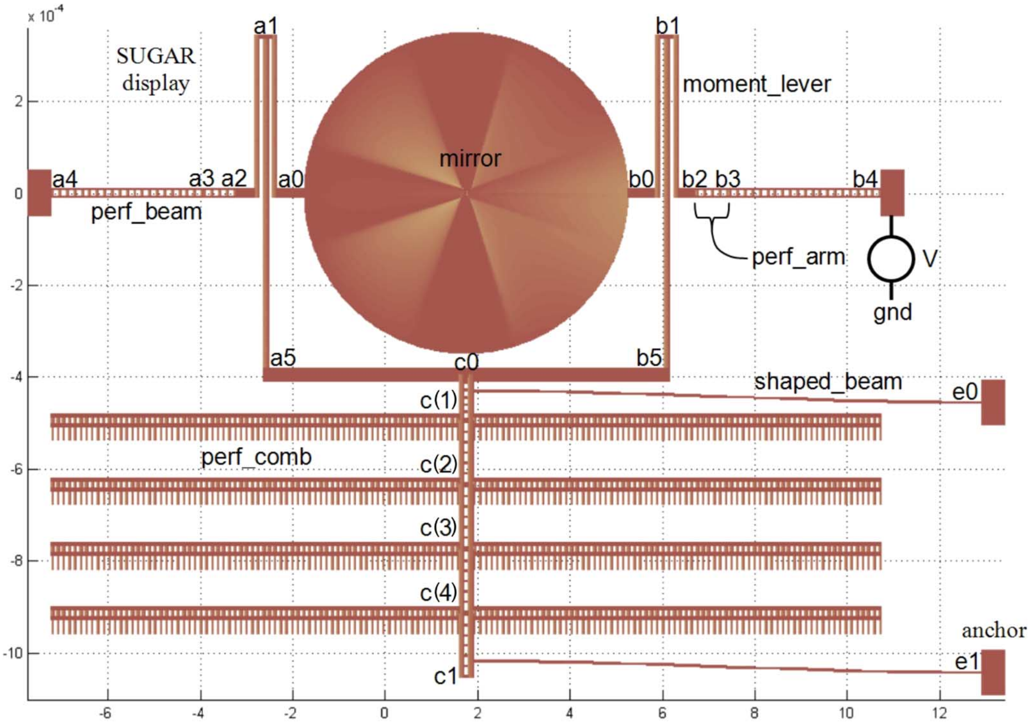
These quantities may stem from analytical analysis, reduced-order modeling, or experiment. The modeling parameters are given by the corresponding netlist component. Model functions also contain display routines.

System Assembly: A system is composed of N model functions. The ith model function contributes subsystem matrices M_i (mass/inductance), D_i (damping/resistance), K_i (stiffness/capacitance), and/or F_i (force/voltage) to the collective system matrices M, D, K, and/or F [18]. The placement of matrix elements from the subsystems into the system is determined by node labels and variables. Our map between nodes, coordinates, and matrix elements facilitates direct matrix element modifications of the system. For example, the varying efforts, change in mass due to chemisorption, finite state switches, and discrete events are types of simulations that benefit from such features.

Equation of Motion: Upon assembly, we can write an equation of motion of the microdevice as

$$M\ddot{q} + D\dot{q} + Kq = \sum F_{\text{ext}} \tag{1}$$

where $\dot{q}$ and q are the flow and displacement state vectors. Keeping the coefficients constant, a wide variety of phenomena



```
                       uses mirror_process.net
Circular mirror ——— mirror            [a0 b0]          [r=  h=  h2=  w=  ]
                    moment_lever      [b0 b1 b2]       [l1=  l2=  l3=  w=  h=  ]
Right torsional hinge perf_arm        [b2 b3]          [h=  h1=  w=  w1=  w2=  l=  n=  ]
                    perf_beam         [b3 b4]          [l=  w=  w1=  w2=   n=  h=  ]
                    anchor            [b4]             [l=  h=  w=  ]
                    moment_lever      [a2 a1 a0]       [l1=  l2=  l3=  w=  h=  ]
Left torsional hinge perf_arm         [a3 a2]          [h=  h1=  w=  w1=  w2=  l=  n=  ]
                    perf_beam         [a4 a3]          [l=  w=  w1=  w2=  n=  h=  ]
                    anchor            [a4]             [l=  h=  w=  ]
                    beam              [b1 b5]          [oz=  l=  w=  h=  ]
Tethers             beam              [a1 a5]          [oz=  l=  w=  h=  ]
                    beamc             [a5 c0 b5]       [l=  w=  h=  ]
                    perf_beam         [c0 c(1)]        [l=  w=  h=  n=  w1=  w2=  ]
                    for j = 1 : ndrives
                      perf_comb       [c(j) c(j+1)]    [nf= w= h= whoriz= wf= lf= gap= wvert= L1= L2= ]
Comb drive array    end
                    perf_beam         [c(ndrives) c1]  [nholes=  w=  h=  l=  whoriz=  wvert=  ]
                    shaped_beam       [c0 e1]          [l=  w=  h=  qy2=  L1=  oz1=  ]
                    anchor            [e1]             [l=  h=  w=  ]
Support beams       shaped_beam       [c1 e2]          [l=  w=  h=  qy2=  L1=  oz1=  ]
                    anchor            [e2]             [l=  h=  w=  ]
                    V                 [b4]             [V=  ]
```

Fig. 6. Netlist. The netlist and its display are shown. We indicate meaningful groups of netlist components with bracketed expressions. One of each component name and the names of all nodes are superimposed on the image. To preserve space, the numerical values of the parameters are not shown.

may be modeled by adding the effective forces to the right-hand side of (1). For example, nonlinear deflections of structural components may be included by adding

$$F_{\text{stiffness}} = K_1 q + K_2 q^3 \tag{2}$$

where K_1 and K_2 are piecewise continuous matrix functions of displacement, and q^3 is a vector of cubed displacements. Thermal expansion in the device may be included by adding

$$F_{\text{thermal}} = AE\alpha(T - T_0) \tag{3}$$

where T is the average temperature of a joule-heated beam due to electric current, T_0 is the temperature of the ambient, and α is the coefficient of thermal expansion. Planar stress and strain gradients may be included by adding

$$F_{\text{stress}} = A\sigma_{\text{residual}} \tag{4}$$

where σ_{residual} is a tensile (positive) or compressive (negative) residual stress of the material, and

$$F_{\text{strain}} = EI_y\Gamma \tag{5}$$

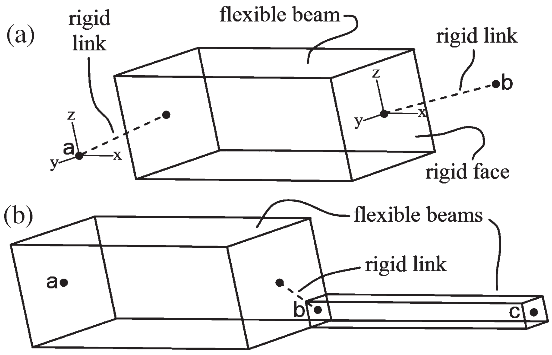


Fig. 7. Rigid links. (a) A transparent flexible electromechanical beam and its corresponding rigid links are depicted. The links are rigidly attached to the end faces of the flexible beam. The end faces of the beam are rigid as well. The reference frame of each link is depicted on the left node of each link. The initial orientations of all reference frames of the component are the same. Upon deflection, the links adhere to the orientation and translation of the end faces. (b) Preferred use of a rigid link, where it is used to physically offset the shared node b of two flexible beams.

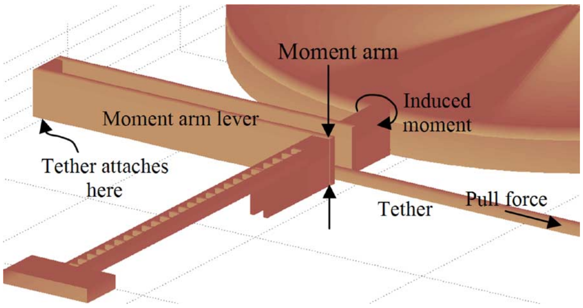


Fig. 8. Design of the torsional hinge assembly by the netlist. Compare this simulation to the real image in Fig. 3. Three nonplanar structural layers are shown, i.e., the perforated beam, the lever arm, and the tether. Previously, it was difficult for the node of a component on one layer to coincide with the node of a component on a different layer. Rigid links help alleviate this problem by offsetting a component's node position.

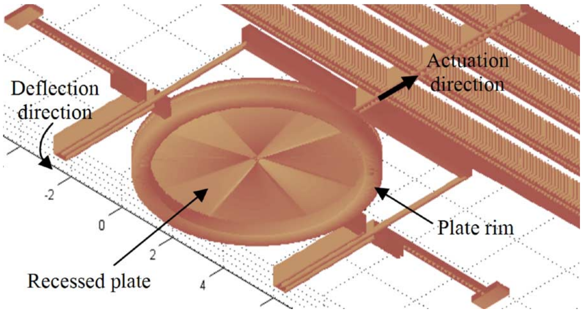


Fig. 9. Recessed plate. A view from underneath shows the recessed plate and the extent of both tethers. An SEM image of a cross section of the recessed plate is shown in Fig. 4.

where Γ is the concave up (positive) or concave down (negative) strain gradient of the material, and F_{strain} is the applied moment vector. As a final example, noninertial effects due to the micromirror operating in an accelerated reference frame may be included by adding

$$F_{\text{noninertial}} = M\ddot{R} - M\omega \times (\omega \times r) - 2M\omega \times \dot{r} - M\dot{\omega} \times r \tag{6}$$

where the right-hand terms are the translational force, centrifugal force, Coriolis force, and transverse force, respectively. The vector R is the position of the substrate, ω is the angular frequency vector of the substrate at R, and r is the position vector of all inertial nodes. See [11] and [12].

Graphics: In addition to abstract mathematical representations, complex engineered design and modeling often require tangible thinking, such as interactively forming an image that resembles the actual system and viewing the system's reaction. Conventional CAE for MEMS tools offer 2-D schematics that display both electrical and mechanical symbols, and network connections. However, micromachining and packaging issues in design and modeling often warrant 3-D visualization at the network or system level. Our model functions have extensible display capabilities. SUGAR is able to display electronic circuit elements with 3-D orientations along with deflecting electromechanical structures. Such capabilities allow designers to more easily identify potential close-proximity interactions between electromechanical components, e.g., parasitic capacitance, mutual inductance, heat conduction, electrostatic induction, and contact.

B. Lumped Modeling Approach for the Test Case

In Section II, we created a netlist description of the micromirror using its decomposed components. We now describe how we model such components. Our modeling approach uses the SPICE-like network features in SUGAR along with a few elementary relationships to build models that are at least 97% accurate within the operating range. We demonstrate the efficiency of this approach in terms of the modeler's effort and in terms of computation. We also show how this approach might be used for many complex building blocks that previously used FEA or advanced analytical derivations. Of the micromirror components listed in Table I, we discuss three of the most interesting, i.e., the cosine-shaped beam, the perforated beam, and the comb drive.

C. Cosine-Shaped Beam

Our approach to modeling the cosine-shaped beam is to reduce a chain of N components into a single component by a simple model-order reduction technique called *matrix condensation*. The model begins by assembling several straight electromechanical beam components that are positioned end to end as links of a chain. These links are oriented such that the chain forms the shape of a cosine from 0 to π. Then matrix condensation is used to reduce the number of elements from N to 1 and reduce the degrees of freedom (DOF) of the building block from $7 \times (N + 1)$ to 14. We choose this particular approach because it requires the least amount of work from the model developer, meets the accuracy requirement, and is based on the electromechanical beam element that we use to model many other components of the micromirror. After a brief description of this element and matrix condensation, we compare a condensed curved lumped element to a curved FEA model.

Electromechanical Beam Element: The mechanical properties of this component are based on the Bernoulli–Euler beam theory using matrix structural analysis [18], and the electrical properties are based on Ohm's law. This 14 × 14 matrix includes DOF for translations and rotations about the x-, y-, and z-directions, and voltages at both nodes.

Matrix Condensation: Matrix condensation is the simpler of the two model-order reduction methods we use in SUGAR. Our more sophisticated method based on Krylov subspaces is reported in [14]. In assembling our curved beam model, the minimum number of electromechanical beam elements that should be chained together is determined by the desired accuracy of the model. The nodes of each straight beam element lie along the backbone of the curved geometry. Since the amplitude of the cosine-shaped beam is much smaller than its length, a chain of ten beam elements, which comprise 70 DOF, is sufficient for our desired accuracy. However, regardless of the number M of DOF in the initial discretization, the final size of the condensed model only depends on the number m of *significant* DOF. We define significant DOF as coordinates that are subject to external efforts, coordinates that are vital for achieving particular vibrational modes, or coordinates of general interest.

Consider the partitioned effort–displacement relationship

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{7}$$

where q_1 is an $m \times 1$ vector of significant DOF, F_2 is an $(M - m) \times 1$ zero vector, and the partitioning is achieved by elementary row operations.

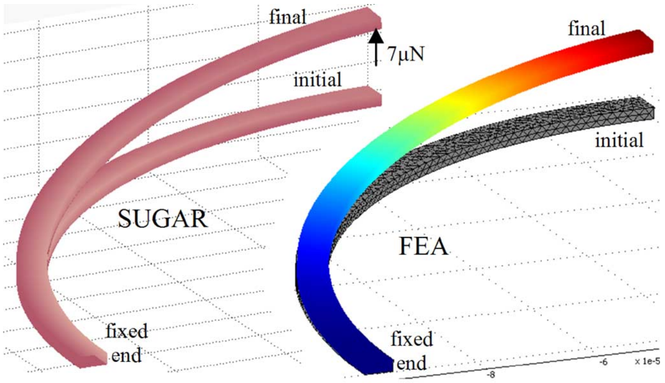
Equation (7) can be rewritten as

$$F_1 = \left[K_{11} - K_{12}K_{22}^{-1}K_{21}\right] q_1 \tag{8}$$

$$q_2 = -K_{22}^{-1}K_{21}q_1. \tag{9}$$

Equation (8) is the condensed and computationally efficient version of (7). The size of its effective stiffness $[K_{11} - K_{12}K_{22}^{-1}K_{21}]$ is $m \times m$. If necessary, (9) may be used to determine a possible set of coordinates q_2 after the significant coordinates q_1 are determined. For static displacements, (8) is just as accurate as (7) for significant DOF. However, matrix condensation must be used with caution in frequency or transient analysis because many higher-order modes may not be accessible due to the reduced DOF. Our reduced-order method reported in [14] is better suited for such higher-order modes.

Curved Beam Verification: We verify our lumped curved beam by comparing its deflection to FEA. Because of SUGAR's flexibility, a curved beam component may take the shape of a wide variety of 3-D parameterized curves. However, creating a cosine-shaped beam in COMSOL that exactly matches the geometry in SUGAR is difficult. To be sure that both SUGAR and COMSOL simulate the same geometry, we choose to model a beam in the shape of a quarter circle. We parameterize this shape in SUGAR by a radius of curvature and the angle subtended, and we form this shape in COMSOL by Boolean geometry. Since the geometry of a quarter circle is more extreme than a small-amplitude cosine, then agreement with FEA increases the confidence in our approach.



SUGAR:	**FEA:**
Elements = 1	Elements = 1487
DOF = 12	DOF = 9783
Time = 0.375 seconds	Time = 72.86 seconds
Deflection = 12.512µm	Deflection = 12.39µm

Percent relative error of SUGAR to FEA = 0.985%

Fig. 10. Curved beam simulation (SUGAR and FEA). Static simulations of circular-shaped beams are shown for SUGAR (left) and FEA (right). To better perceive the deflections, the final states are superimposed onto the initial states. The SUGAR model consists of one element, and the FEA model consists of 1487 elements (the discretization is shown on the initial FEA state). A transverse force of 7 μN was applied at the far ends of beams. The relative error in SUGAR deflection is with respect to FEA. Parameters for SUGAR and FEA include width = 5 μm, thickness = 2 μm, Young's modulus = 170 GPa, Poisson's ratio = 0.3, radius of curvature = 97.5 μm, and angle subtended = $\pi/2$.

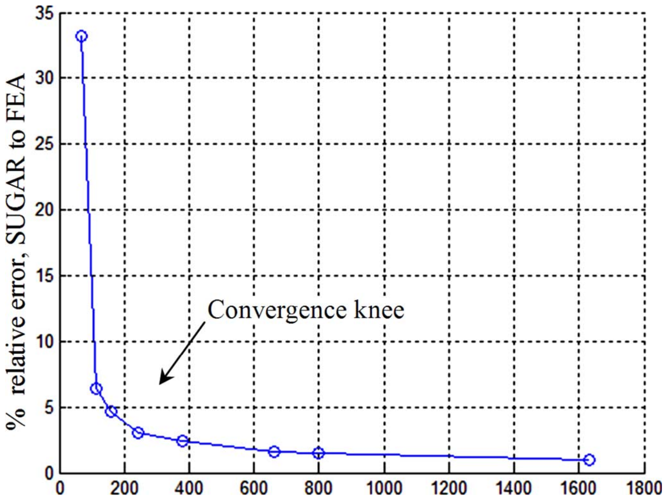


Fig. 11. Curved beam convergence between SUGAR and FEA. The eight absolute relative errors (circled) were determined by comparing one deflection from SUGAR and eight deflections from FEA. The number of elements in the FEA simulations ranged from 44 to 1487, and the corresponding DOF for FEA ranged from 411 to 9783. One curved beam element was used in the SUGAR simulation, and the element had seven DOFs (three translational + three rotational + one electrical). The relative error between SUGAR and FEA achieves less than 1% for small deflections. Fig. 10 shows an intermediate FEA mesh.

Simulations of curved beams in the shape of a quarter of a circle are shown in Fig. 10. SUGAR is over two orders of magnitude faster than FEA due its computational efficiency. The relative error of SUGAR is less than 1% of the nonlinear FEA. In Fig. 11, we show that the percent relative error between

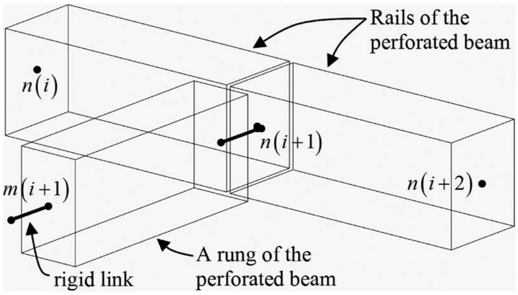


Fig. 12. Node connectivity for the perforated beam component. Three of the N subnet elements are depicted. The elements are slightly separated to help distinguish one from the other. The elements are transparent to help identify the nodes (dots). We array the node labels so that the number of perforations can be a parameter of the component. Two rail elements align end to end at node $n(i+1)$. To prevent structural overlap, we use a rigid link (recall Fig. 7) to offset the node $n(i+1)$ from the rung. The length of the rigid link is half the width of the rail.

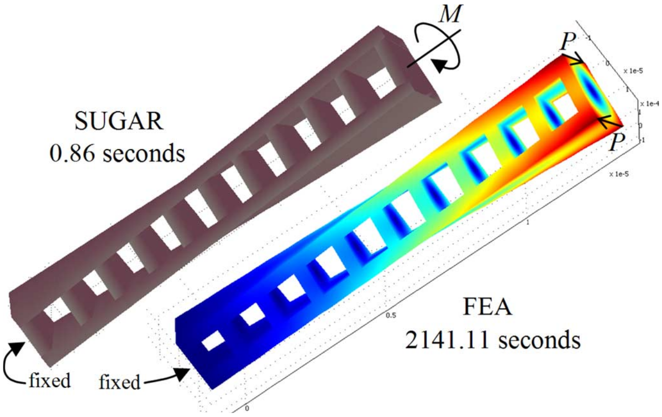


Fig. 13. Perforated beam simulation (SUGAR and 0020FEA). Displacement values are color mapped onto the surface of the FEA simulation. We produced the deflections by directly applying the moment M in SUGAR and a pair of surface pressures P in FEA. Solution time for SUGAR was over three orders of magnitude faster than FEA. SUGAR and FEA parameters include length $= 120$ μm, width $= 18$ μm, thickness $= 18$ μm, perforations $=$ 10 μm $\times$ 10 μm, Young's modulus $= 170$ GPa, and Poisson's ratio $= 0.3$.

SUGAR and FEA improves as FEA discretization increases. We indicate a characteristic *convergence knee* in the plot.

D. Perforated Beam

Our approach to modeling the perforated beam is to create a subnet that is parameterized by the parameters indicated in Fig. 5 and Table I. As shown in the third row of Table I, each node of the perforated beam is positioned at the center of either end face. This node position works well for beam elements that connect flush at their end faces. However, if the beams attach to one another at right angles, then there is considerable structural overlap of the end corners of the beam. This problem is somewhat remedied by offsetting the node positions by rigid links. Fig. 12 depicts how we assemble the rail and rung of a perforated beam using rigid links.

Simulations of SUGAR and FEA deflections are shown in Fig. 13. Although it is easy to apply a moment at the end node of SUGAR's perforated beam, applying an equivalent moment in FEA is not trivial. We obtain slightly different results depending on which way we create moments in FEA. Relative errors from these results range from 3% to 17%. The most agreeable FEA result is shown in Fig. 14, where a near-rigid crossbar is attached to the end of a perforated beam, and two coupling pressures are applied at the far ends of the bar to produce a moment. To improve the accuracy and precision of the FEA simulation, we shortened the beam to free up memory for a more refined mesh.

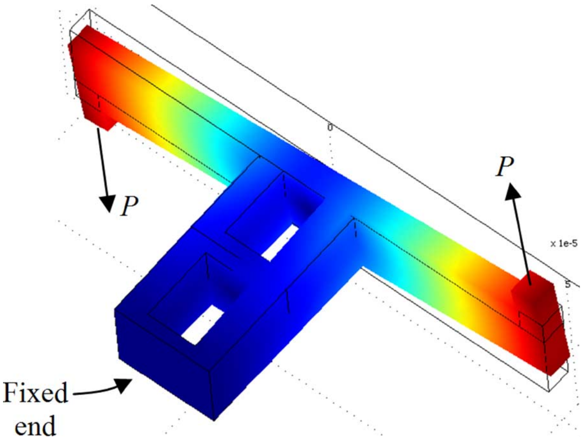


Fig. 14. Simulation of the FEA model used to test the perforated beam component. We apply a pair of normal pressures P on two surfaces located at the far ends of the crossbar. For the corresponding SUGAR model, we apply an equivalent moment. This shortened version of the perforated beam allows for a higher discretization density.

The perforated beams in the micromirror are subject to both moments and forces. Here, we compare maximum twist angles due to moments. Rotational displacements in SUGAR are given in its solution vector. We determine the twist angle in FEA from the tangent of the maximum translational displacement divided by half the crossbar length. A comparison between the twist angles of SUGAR and FEA is shown in Fig. 15. The relative error between SUGAR and FEA is less than 3%.

E. Comb Drive

Our comb drive model for the micromirror test case includes mechanical and electrical properties from the electromechanical beam element, and electrostatic forces derived from a parallel-plate approximation. We compare this model against FEA in 2.5-D. The SUGAR model agrees well with 2.5-D FEA due to the high aspect ratio between the layer thickness and the gap, which is 50 : 3. Because of our limited computer memory, the convergence analysis between 3-D FEA and SUGAR is currently inconclusive.

Our 2.5-D analysis includes three and four interdigitated conducting comb fingers in vacuum. The volume of the field surrounding the comb drive is unrealistically small, which results in a reduced capacitance. However, since we are primarily concerned with computing the force due to a change in capacitance, the change of any additional capacitance due to a larger field would be small since the additional field would be much further away from the electrodes. Moreover, a higher precision can be obtained from a maximally refined small field mesh as opposed to a larger field that cannot be as densely meshed. Fig. 16 shows the 2.5-D FEA configuration and field solution of the comb drive model. We determine the force F_{FEA} in (10)

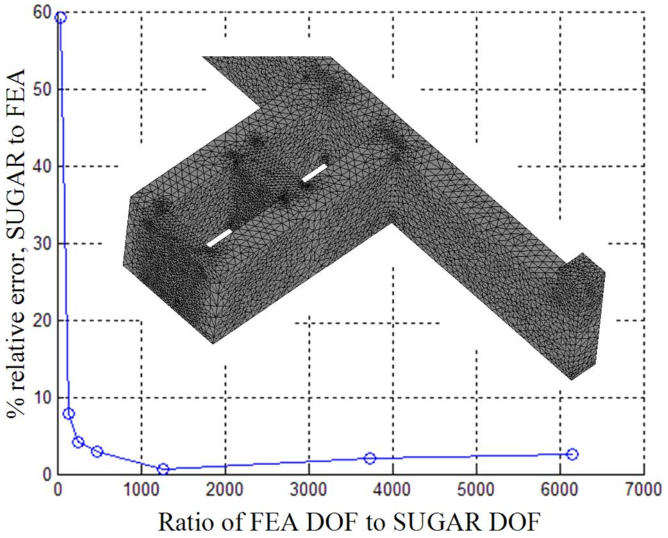


Fig. 15. Perforated beam convergence between SUGAR and FEA. We determine the data by comparing one twist deflection from SUGAR against several FEA simulations. The number of elements in the FEA simulations ranges from 314 to 31 247, and the corresponding DOFs for FEA range from 1038 to 147 654. The deflected geometry is shown in Fig. 14. We use seven electromechanical beam elements in the SUGAR simulation, which implies 60 DOFs. The relative error between SUGAR and FEA achieves less than 3% for the most refined FEA simulation. The reason that the graph is not monotonic is likely due to the size and positioning of the discretized elements from the automated mesh generator, which is different for each data point. The inset shows an intermediate mesh.

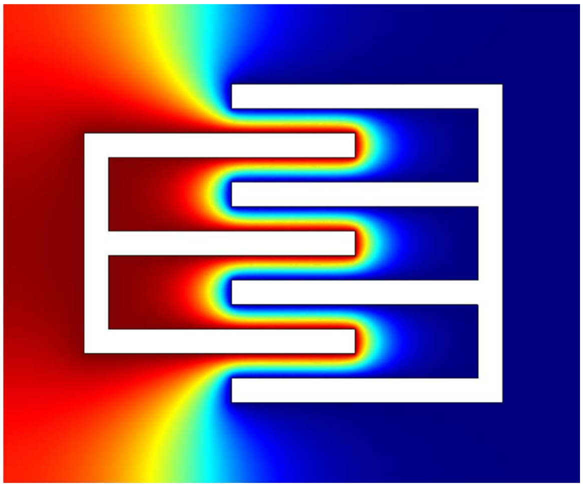


Fig. 16. 2.5-D FEA simulation of a comb drive. A plot of the equipotentials is shown. We approximate the out-of-plane third dimension by extrusion. The aspect ratio between the extrusion depth and the finger width is 16.6 to 1. The simulated comb drive is a perfect conductor with flat surfaces and sharp corners.

by first computing the capacitance $C_{\text{FEA},i} = \int 2W_{e,i}hdA/V^2$ of the ith configuration, where $W_{e,i}$ is the energy density. We horizontally displace one of the combs a small amount $x_{i+1} - x_i$ by using the arbitrary Lagrangian–Eulerian moving boundary technique [19]. Then we compute the new capacitance $C_{\text{FEA},i+1}$. In Fig. 17, this force

$$F_{\text{FEA}} = \frac{1}{2}V^2 \frac{(C_{\text{FEA},i+1} - C_{\text{FEA},i})}{x_{i+1} - x_i} \tag{10}$$

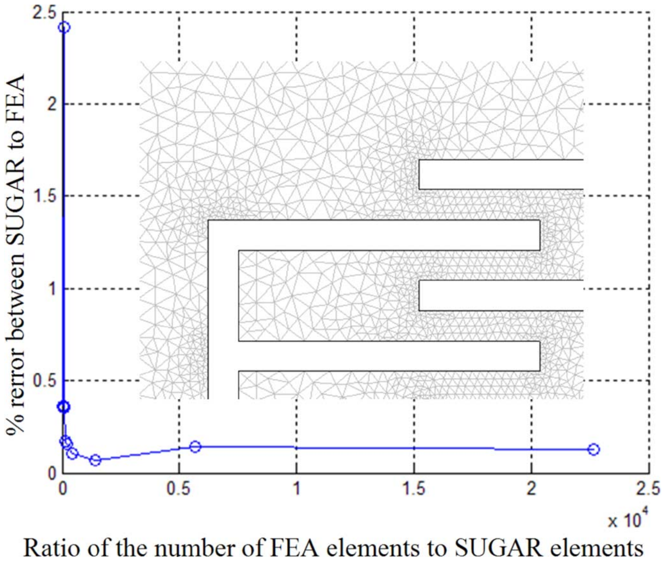


Fig. 17. Comb drive force convergence between SUGAR and FEA. We determine the data points by comparing the comb drive force in SUGAR against several FEA simulations. The number of elements in the FEA simulations ranges from 214 to 113 416, and the corresponding DOFs for FEA range from 612 to 229 195. We represent this model in SUGAR with five electromechanical beam elements and 42 DOFs. The relative error between SUGAR and FEA is less than 0.14% for the most refined FEA simulation. The inset shows an intermediate mesh.

is compared to the parallel-plate approximation

$$F_{\|} = N\varepsilon_0 V^2 \frac{h}{g} \tag{11}$$

where N is the number of fingers on the comb that has the least amount of fingers.

The force in (11) approximates the net force that acts upon each comb. Each comb of the micromirror consists of a multitude of fingers. For computational efficiency, instead of applying hundreds of force vectors, we treat these forces as a uniform distribution about the backbone of the comb drive and compute the equivalent lumped force and moment. We apply these lumped forces and moments at the far ends of each comb. That is, since the force per until length of the load is $P = F_{\|}/L$, the equivalent force and moment are

$$F = \int P\,\psi_F\,dx = \frac{N\varepsilon_0 h}{2g}V^2 \tag{12}$$

$$M = \int P\,\psi_M\,dx = \frac{N\varepsilon_0 hL}{12g}V^2 \tag{13}$$

where ψ_F and ψ_M are Hermitian shape functions [18]. We use a SUGAR simulation shown in Fig. 18 to exemplify this methodology.

To conclude this section on modeling, we have shown that our models are computationally efficient and are nearly as accurate as their FEA counterparts. We demonstrated the accuracy of our models by comparing them to FEA versions of the models, which we meshed as finely as our computer memory allowed. Comparisons of CPU times and DOF demonstrated the computational efficiency of our models. Next, we combine

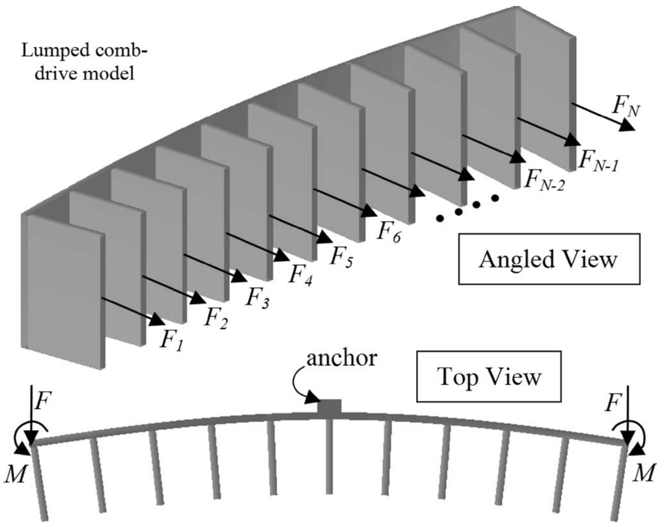


Fig. 18. SUGAR simulation of our efficient comb drive model. The Angled View shows a multitude of forces applied at the ends of comb fingers. The Top View shows the equivalent lumped forces and moments applied to the far ends to the comb drive. Since the comb drive forces are relatively small, we exaggerate the deflection so that the effect of forces and moments can be seen.

our models into a complete system and compare SUGAR simulation to experiment.

V. Experimental Verification

In Section III, we configured a micromirror using a netlist description. In Section IV, we created models for the netlist components. In this section, we show that the assembled system of models in SUGAR fairly agrees with the experiment.

The material properties were given by the wafer manufacturer, and the geometries of the micromirror were measured with a scanning electron microscope [16]. In the following analysis, we assume that these material and geometric parameters have smaller relative errors than the relative errors of our models. One exception however is the layer thickness of the two tethers. We were given a small range of likely thicknesses, and we chose to go with an intermediate value.

Measurements of the tilt angle of the micromirror were performed by deflecting laser light from it onto a distant wall. Thirteen angles were measured from 0° at 0 V to 22° at 125 V. Using voltage as an input parameter to the SUGAR netlist, we sweep the voltage from 0 to 150 V. The measured data and SUGAR simulations are shown in Fig. 19 in terms of mirror tilt angle as a function of voltage.

The goal of this micromirror design is to maximize the tilt angle per voltage. The design efficiency of our netlist and the computational efficiency of our models should work well with the various optimization techniques. For instance, in Fig. 20, we sweep a couple of parameters to examine their dependence on the tilt angle for a given voltage. The figure shows which combination of moment arm length and perforated beamwidth yields the largest tilt angle at 80 V.

After the design space is explored in SUGAR, the resulting netlist(s) may be exported into the standard Caltech Interchange Format (CIF) for fabrication or for further refinement and analysis in conventional MEMS software.

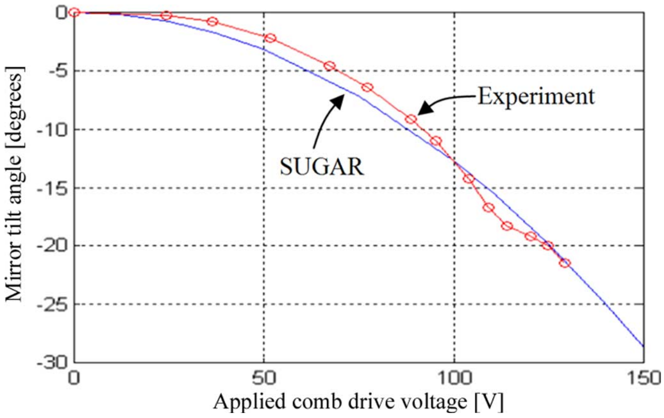


Fig. 19. Experiment versus SUGAR, and mirror angle versus voltage. Experimental measurements are shown as circles. We sweep the voltage in the SUGAR simulation (solid line) from 0 to 150 V.

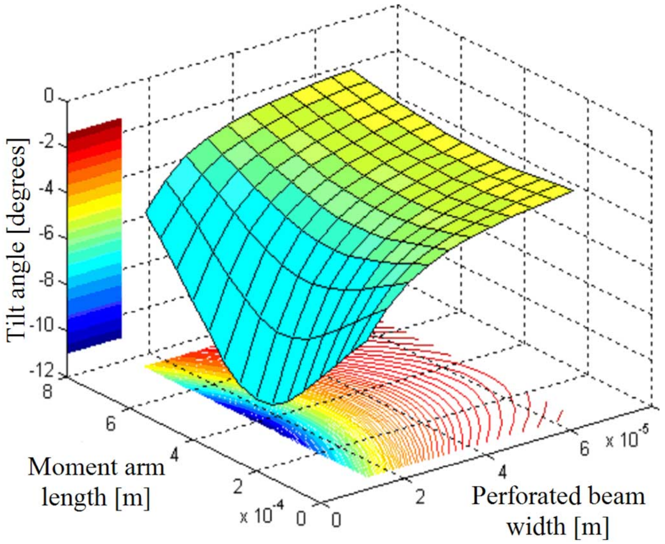


Fig. 20. Parameter sweeps. Tilt angle versus perforated beamwidth versus moment arm length is shown for an 80-V comb drive actuation.

VI. Discussion

In this section, we discuss a few simulation issues, relate the benefits of SUGAR to research and development (R&D), mention the importance of developing a library of network- and system-level MEMS models, look forward to the next level of system level, and mention some research work done by others that is based in SUGAR.

Simulation: We performed the simulations on a Pentium-M 2.13-GHz notebook with 1.5-GB random access memory. Two key reasons why it is difficult for us to accurately simulate the fully assembled micromirror using FEA on our notebook are due to the high mesh density required to surpass the convergence knee and the large number of elements required for multiphysics coupling between the mechanical structure and the electrostatic field. As was shown in the plots of relative errors in Section IV, we begin to achieve acceptable accuracy using FEA after the number of elements surpasses the abrupt convergence knee. This level of accuracy typically requires an automated high mesh density, the strategic refinement of mesh density in regions where physical quantities have a high rate of spatial change, and/or higher-order elements. Fine discretizations consume a large amount of memory and CPU time. In addition, multiphysics coupling between mechanical and electrostatic domains usually involves insignificant, but costly, meshed spaces about structures that are far removed from the significant fields that surround the comb fingers.

We used quadratic Lagrange tetrahedral elements and an automated mesh generator for our FEA simulations. This mesh generator increases the mesh density in regions that are relatively narrow or that have corners. However, the mesh generator does not produce an optimal mesh. That is, an engineer skilled in the art of meshing can likely achieve better accuracy with fewer elements. With respect to our plots of FEA convergence, optimizing the mesh and using higher-order elements may increase the rate of convergence. However, due to process variations, the analyst must weigh the precision of simulation against the precision of measuring geometric and material properties of the true device.

Recall that there are several FEA-based tools for MEMS to choose from. For our FEA comparisons, we chose to use COMSOL-3.2 mostly due to familiarity.

R&D: The micromirror test case used in this paper had undergone about a dozen fabrication runs—about a couple of years of design changes by trial and error—by a graduate student and a post doc before we were asked to model it using SUGAR. This amount of R&D time is common for MEMS researchers. However, we were able to make the netlist shown in Fig. 6, create the models shown in Table I, and explore a design space shown in Fig. 20 in about one day's time. The rate of comparing our models against FEA averaged to about one component per day. Although comparing our lumped models against FEA provides preliminary confidence, comparing simulation against experiment (Fig. 19) is a more definitive test of confidence. We suggest that by readily exploring what-if scenarios in SUGAR, then it is likely that the conventional time for R&D can be significantly reduced.

Model Libraries: As mentioned in Section I, there are growing sets of commonly used MEMS components. However, there is a lack of adequate lumped models for these components. Unlike FEA, network- or system-level tools are limited by their models. For models that are available, there appears to be no clearinghouse to collect or distribute them. We anticipate that by having such a MEMS tool in the public domain, then the use, development rate, and distribution of lumped models for MEMS will increase. A similar strategy was done with SPICE for electronics.

System of Systems: As memory and processing power increase, then the time required to simulate advanced designs using FEA will become practical. However, as micro- and nanotechnologies continue to grow, then the size and complexity of these systems will likely grow as well. There will likely be a need to predict the behavior of such systems as well as a need to predict the behavior of a multitude of such systems interacting with each other or other systems. This is a research area where clever system-level approaches are likely to lead.

Other Work: Since SUGAR's inception, there has been a growing number of applications and features developed for SUGAR from researchers and student class projects. Some significant developments have been the application of evolutionary synthesis and optimization to SUGAR by Zhou *et al.* [21], [22], the process variation analysis [11], the netlist-to-CIF converters by Kao and Garmire, a CIF-to-netlist converter for SUGAR by Mukherjee *et al.*, the case-based reasoning tools by Cobb and Agogino [23], a version 3 of SUGAR by Bindel [3], and the Krylov subspace-based reduced-order modeling techniques by Bai *et al.* [14].

VII. Conclusion

In this paper, we have presented efficient configuration, modeling, and simulation techniques packaged in a CAE for MEMS tool called SUGAR, version 2. On the topic of configuration, we have shown how a complex microdevice could be efficiently configured using parameterized netlists. For instance, we configured a thousand-plus components of a micromirror using just a few lines of netlist text. As shown, SUGAR's 3-D network/system-level display can be useful for designing advanced microdevices. On the topic of modeling, we have described our extensible modeling framework and shown that our lumped models were accurate and computationally efficient. In particular, we showed that the errors of decomposed SUGAR components were less than 3% of FEA, and the time for SUGAR computation took less than 1% of FEA. Moreover, on the topic of simulation, we have shown that SUGAR may be used to efficiently explore variations in performance by parameterizing the design space of an advanced micromirror. Most important, we showed that SUGAR's static analysis fairly agreed with the experiment.

From this paper, we learned that a fairly general set of parameterized network components can efficiently configure and quickly predict the performance of an advanced microdevice. We suggest that there exist a large set of advanced MEMS for which appropriate system-level tools can be used to reduce their conventional time for research and development.

Acknowledgment

The authors would like to thank M. Last and V. Milanovic for contributing their SEM images and experimental data, N. Zhou for numerous contributions to early versions of SUGAR, Prof. A. M. Agogino, Z. Bai, D. Bindel, J. Demmel, S. Govindjee, and C. H. Sequin for collaborative discussions, and the helpful comments from both national and international users of SUGAR.

References

[1] S. D. Senturia, "CAD challenges for microsensors, microactuators and microsystems," *Proc. IEEE*, vol. 86, no. 8, pp. 1611–1626, Aug. 1998.

[2] S. Senturia, N. Aluru, and J. White, "Simulating the behavior of MEMS devices: Computational methods and needs," *IEEE Comput. Sci. Eng. Mag.*, vol. 16, no. 10, pp. 30–43, Jan.–Mar. 1997.

[3] *SUGAR*. [Online]. Available: www-bsac.eecs.berkeley.edu/cadtools/sugar

[4] ANSYS Inc., Canonsburg PA. [Online]. Available: http:www.ansys.com

[5] Coventorware, San Mateo, CA [Online]. Available: http://www.coventor.com

[6] *IntelliSuite*, IntelliSense Corp., Woburn, MA. [Online]. Available: http://www.intellisense.com

[7] COMSOL Inc., Los Angeles, CA. [Online]. Available: http://www.comsol.com

[8] G. K. Fedder and Q. Jing, "Hierarchical circuit-level design methodology for micromechanical systems," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 10, pp. 1309–1315, Oct. 1999.

[9] L. Nagel, *SPICE: A Computer Program to Simulate Semiconductor Circuits*, May 1975. UCB/ERL M520.

[10] J. V. Clark, N. Zhou, and K. S. J. Pister, "MEMS simulation using sugar v0.5," in *Proc. Transducer's Solid-State Sensor and Actuator Workshop*, Hilton Head Island, SC, Jun. 8–11, 1998, pp. 191–196.

[11] J. V. Clark, N. Zhou, D. Bindel, L. Schenato, W. Wu, J. Demmel, and K. S. J. Pister, "3D MEMS simulation modeling using modified nodal analysis," in *Proc. Microscale Syst.: Mech. and Meas. Symp.*, Orlando, FL, Jun. 8, 2000, pp. 68–75.

[12] J. V. Clark, D. Bindel, N. Zhou, S. Bhave, Z. Bai, J. Demmel, and K. S. J. Pister, "Sugar: Advancements in a 3D multi-domain simulation package for MEMS," in *Proc. Microscale Syst.: Mechanics and Meas. Symp.*, Portland, OR, Jun. 4, 2001.

[13] J. V. Clark, D. Bindel, W. Kao, E. Zhu, A. Kuo, N. Zhou, J. Nie, J. Demmel, Z. Bai, S. Govindjee, K. S. J. Pister, M. Gu, and A. Agogino, "Addressing the needs of complex MEMS design," in *Proc. IEEE 15th Annu. Int. Conf. Micro Electro Mech. Syst.*, Las Vegas, NV, Jan. 20–24, 2002, pp. 204–209.

[14] Z. Bai, D. Bindel, J. V. Clark, J. Demmel, K. S. J. Pister, and N. Zhou, "New numerical techniques and tools in sugar for 3D MEMS simulation," in *Proc. Tech. 4th Int. Conf. Modeling and Simul. Microsyst.*, Hilton Head Island, SC, Mar. 19–21, 2001, pp. 31–34.

[15] J. Lohn *et al.*, Comput. Sci. Division, NASA Ames Res. Center, Moffett Field, CA. [Online]. Available: http://ic.arc.nasa.gov/people/jlohn

[16] M. Last and V. Melanovic, Univ. California, Berkeley, private communication.

[17] *Dust Networks*. [Online]. Available: http://www.dust-inc.com

[18] M. Paz, *Structural Dynamics Theory and Computation*. New York: Litton Education, 1980.

[19] I. Harouche and C. Shafai, "Simulation of shaped comb drive as a stepped actuator for microtweezers application," *Sens. Actuators A, Phys.*, vol. 123/124, pp. 540–546, 2005.

[20] T.-J. Su and R. R. Craig, Jr., "Model reduction and control of flexible structures using Krylov vectors," in *J. Guid.*, vol. 14, 1991, pp. 260–267.

[21] N. Zhou, B. Zhu, A. M. Agogino, and K. S. J. Pister, "Evolutionary synthesis of microelectromechanical systems (MEMS) design," in *Proc. ANNIE: IEEE Neural Netw. Council and Smart Eng. Syst. Lab.*, Nov. 4–7, 2001, vol. 11, pp. 197–202.

[22] Y. Zhang, R. Kamalian, A. M. Agogino, and C. H. Sequin, "Hierarchical MEMS synthesis and optimization," presented at the SPIE Conf. Smart Structures and Materials, San Diego, CA, Mar. 7–10, 2005, Paper 5763-12.

[23] C. L. Cobb and A. M. Agogino, "Case-based reasoning for the design of micro-electro-mechanical systems," in *Proc. IDETC/CIE*, 2006.

[24] Softmems. [Online]. Available: http://www.softmems.com

[25] G. Lorenz and R. Neul, "Network-type modeling of micromachined sensor systems," in *Proc. Int. Conf. MSM*, Santa Clara, CA, Apr. 1998, pp. 233–238.

[26] Y. J. Yang and P. Yen, "An efficient micromodeling methodology for lateral air damping effects," *J. Microelectromech. Syst.*, vol. 14, no. 4, pp. 821–828, Aug. 2005.

[27] C. L. Cobb and A. M. Agogino, "Case-based reasoning for the design of micro-electro-mechanical systems," in *Proc. ASME 2006 IDETC/CIE Conf.*, Sep. 10–13, 2006, #DETC2006-99120.

[28] T. Mukherjee, G. K. Fedder, D. Ramaswamy, and J. White, "Emerging simulation approaches for micromachined devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 12, pp. 1572–1589, Dec. 2000.

Jason Vaughn Clark received the B.S. degree in physics from California State University, Hayward, in 1996, and the Ph.D. degree in applied science and technology from the University of California, Berkeley, in 2005.

Since 2006, he has been an Assistant Professor of electrical and computer engineering and an Assistant Professor of mechanical engineering with Purdue University, West Lafayette, IN. He is also a faculty member with the Birck Nanotechnology Center (BNC) and the Network for Computational Nanotechnology (NCN). His current research interests include the design, modeling, simulation, and verification of complex engineered systems.

Kristofer S. J. Pister received the B.S. degree in applied physics from the University of California at San Diego (UCSD), La Jolla, in 1986, and the M.S. and the Ph.D. degrees in electrical engineering from the University of California, Berkeley (UCB), in 1989 and 1992, respectively.

From 1992 to 1997, he was an Assistant Professor of electrical engineering with the University of California, Los Angeles (UCLA), where he helped develop the graduate MEMS curriculum. Since 1996, he has been a Professor of electrical engineering and computer sciences with UCB. In 2003 and 2004, while he was on leave from UCB, he was the CEO and then CTO of Dust Networks, a company he founded to commercialize Smart Dust. The last 18 years of his academic career have been spent in the pursuit of millimeter-scale robots. His research efforts have yielded few robots but several commercially viable technologies including nodal CAD for MEMS, xenon difluoride etching, and wireless sensor networks. He has participated in many government science and technology programs, including the Defense Advanced Research Projects Agency's Innovative Space Based Radar Antenna Technology project (DARPA's ISAT) and Defense Science Study Groups. His research interests include MEMS, micro robotics, and low-power circuits.

Dr. Pister is currently a member of the Jasons.