Continuation-Based Pull-In and Lift-Off Simulation Algorithms for Microelectromechanical Devices

Zheng Zhang, *Student Member, IEEE*, Mattan Kamon, *Senior Member, IEEE*, and Luca Daniel, *Member, IEEE*

Abstract—The voltages at which Micro-Electro-Mechanical (MEM) actuators and sensors become unstable, known as pull-in and lift-off voltages, are critical parameters in MEMS design. The state-of-the-art MEMS simulators compute these parameters by simply sweeping the voltage, leading to either excessively large computational cost, or to convergence failure near the pull-in or lift-off points. This paper proposes to simulate the behavior at pull-in and lift-off employing two continuation-based algorithms. The first algorithm appropriately adapts standard continuation methods, providing a complete set of static solutions. The second algorithm uses continuation to trace two kinds of curves and generates the sweep-up or sweep-down curves, which can provide more intuition to MEMS designers. The algorithms presented in this paper are robust and suitable for general-purpose industrial MEMS designs. Our algorithms have been implemented in a commercial MEMS/IC co-design tool, and their effectiveness is validated by comparisons against measurement data and the commercial FEM/BEM solver CoventorWare.

Index Terms—MEMS CAD, pull-in, lift-off, continuation

I. INTRODUCTION

A. Motivation

MEMS devices are widely used in system-on-chip (SOC) applications such as sensors, actuators, as well as circuit functional blocks [1]–[7]. Due to the increasing design complexity and the emergence of MEMS/IC co-design, Electronic Design Automation (EDA) tools are highly desired for device-level and system-level MEMS design [8]–[15]. This paper presents a robust and general-purpose framework to simulate two key parameters in MEMS design: the pull-in and lift-off voltages.

It is well known that the coupling of electrostatic and mechanical forces leads to a small set of voltage values which are unstable and cause the device to rapidly shift from one state to another [1]. Depending on specific applications, such instability must be avoided or could be exploited. For instance, when designing MEMS switches, such as those used in RF [6], logic [2]–[5], or digital mirror [7] applications, these instabilities are exploited: once the instability is passed, the device rapidly switches either from "off" to "on" as the moving part is "pulled-in" to make contact with a fixed surface, or from "on" to "off" as the device "lifts-off" the fixed surface. On the other hand, for stable operations of some MEMS devices, small perturbations in the applied voltage bias should not produce large perturbations in the position of the moving components. Typical examples include MEMS

Z. Zhang and L. Daniel are with the Department of Electrical Engineering and Computer Science, the Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA (Email: z_zhang@mit.edu, luca@mit.edu).

M. Kamon is with Coventor, Inc., 625 Mount Auburn Street, Cambridge, MA 02138, USA (Email: matt@coventor.com).

resonators, such as those used for inertial sensors [16] or quartz replacement [17]. For these devices, the DC bias must thus be chosen adequately far from the pull-in point to avoid nonlinear or even unstable behavior. Therefore, pull-in and lift-off computation is key to a wide range of MEMS designs.

There are some well-established techniques for computing the pull-in voltage. Based on the formulation of boundary-element method (BEM) and finite element analysis (FEA) [18]–[21], these methods can be used for a wide range of MEMS design. However, these algorithms directly sweep the voltage (i.e., solves the static problem at a set of monotonically increasing input voltages), and once the pull-in point is passed, they must find a solution which is no longer "near" the solution at the previous voltage. As a result, these methods may fail to converge near the pull-in, or require many iterations even if they converge by luck. Some analytical and numerical approaches can avoid the computational issues near pull-in for some specific devices [22]–[26], but none of them are general or without heuristics. In [26], [27], the variable used in the DC sweep is changed to a variable dependent on voltage and position, such as charge. This change moves the instability to a larger voltage value enabling robust pull-in computation, but retains the difficulties in determining lift-off for designs other than the ideal parallel plate. The algorithms in [25], [26], [28] require *a priori* knowledge about the path taken by parts of the design, which limits their generality. Standard continuation methods were employed to help estimate the pull-in voltages [29]–[31] based on some simplified MEMS models or for some specific devices. Such algorithms have been extensively investigated by the applied mathematics community [32]–[34] and have been successfully applied to circuit simulation [35]. Nevertheless, standard continuation algorithms can be unsuitable to simulate industrial MEMS designs that are in a more general form (c.f. the end of Section II-C). Compared with pull-in, much less attention has been paid to the computation of lift-off, which can be as important as pull-in for MEMS switch devices [2]–[7].

B. Summary of Contributions

By appropriately modifying standard continuation methods, we present two algorithms to simulate the pull-in/lift-off behavior and demonstrate their application in industrial MEMS designs. Based on a hybrid FEA/analytical-model platform [36], our algorithms are applicable for general industrial MEMS designs with little heuristics.

The first method appropriately modifies standard continuation methods [32]–[34] to simulate the pull-in/lift-off effects in

real-world MEMS designs. By properly modeling the contact effects and modifying the stability analysis, this method can provide a complete plot of multiple solution branches for a practical MEMS design. The large computational cost or failures around the pull-in and lift-off points are avoided.

The second approach is motivated by the needs of some designers for more intuitive static solution curves of sweep-up or sweep-down. This method uses continuation to trace two kinds of curves. Specifically, it traces the solution curve of the original MEMS problems before and after pull-in/lift-off points, and it traces a newly defined curve to capture the jump between states that occur at the instabilities. Unlike a standard DC-sweep the convergence failures around pull-in and lift-off can be mitigated or avoided. In this algorithm, an approach is proposed to detect the pull-in/lift-off point with high accuracy, which can be useful for sensitivity analysis or for analyzing the noise tolerance in MEMS-based digital VLSI.

The proposed algorithms have been verified on industrial MEMS designs. In this paper, we will provide the simulation results of two practical MEMS designs, and we will also compare our proposed methods with measurement data and existing relaxation-based FEA/BEM-based MEMS simulator CoventorWare [37] to validate the efficiency and effectiveness.

II. PRELIMINARIES

A. An Illustrative Example

The pull-in and lift-off effects are best illustrated by the idealized parallel-plate electro-static actuator in Fig. 1a [1]. The movable top plate is attached to a simple spring, the fixed bottom plate is covered by a thin dielectric layer of thickness d, and a DC voltage V is applied across the plates. The gap between the two plates is g. The position z of the upper plate is restricted to move only in the z-direction. We assume that $z = 0$ when the DC voltage $V = 0$.

Pull-In: When increasing V from 0, z will decrease smoothly if V is not too big, since z is a stable equilibrium point [1]. When V approaches a critical value V_{PI}, called the **pull-in voltage**, the stability of the equilibrium point $z = z_{\mathrm{PI}}$ is lost (point A in Fig. 1b). If we further increase V slightly, due to instability the upper plate will abruptly crash down and come into contact with the dielectric layer (shown by point B in Fig. 1b). The solid green curve from $V = 0$ to point A in Fig. 1b shows the stable equilibrium solutions.

Lift-Off: Under a large input $V > V_{\mathrm{PI}}$, the upper plate keeps contact with the dielectric layer. As V is reduced to a critical value V_{LO}, called the **lift-off voltage**, the electrostatic force is equal to the spring force and the displacement z_{LO} (point C of Fig 1b) becomes an unstable equilibrium point. The upper plate will jump up and separate from the bottom plate (point D of Fig. 1b) if the input V is further reduced.

Hysteresis: Fig. 1b shows the pull-in/lift-off effects of this actuator. The displacement z depends on both the input V and its past history. Within the hysteresis region ($V_{\mathrm{LO}} < V < V_{\mathrm{PI}}$) there exists another equilibrium solution branch shown by the dashed curve between points A and C. This branch is an unstable equilibrium, therefore it is not practically useful, but its connection to the two stable branches is key to the success of the first simulation method proposed in this paper.
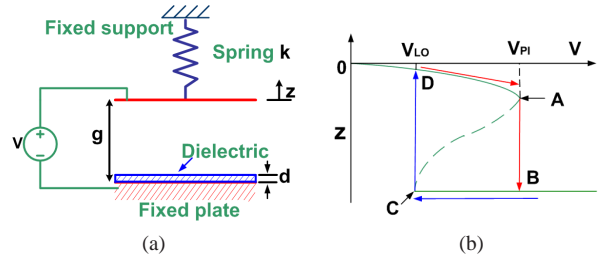

Fig. 1: (a) Parallel-plate electro-static actuator electrostatic actuator. (b) Hysteresis effect of this actuator.

B. Difficulties in Standard MEMS Simulators

A general MEMS design can be described by

$$\mathbf{M}(x)\ddot{x} + \mathbf{K}(x)\dot{x} + F(x, u(t)) = 0, \tag{1}$$

where $x \in \mathbb{R}^n$ denotes displacements and rotations; $u(t)$ denotes the inputs such as voltage sources; $\mathbf{M}(x), \mathbf{K}(x) \in \mathbb{R}^{n \times n}$ are the mass matrix and damping coefficient matrix, respectively; $F(x, u(t))$ denotes the net forces from inertia, damping, electrostatic and mechanical forces. Throughout this paper, x and Eq. (1) are properly scaled, such that x is on the order of 1 and $F(x, u(t))$ is on the order of 10^{-6}. If the MEMS model is included in a circuit simulator, 10^{-6} is consistent with the μA scale for currents.

In the static state, $u(t) = u$ is constant, $\ddot{x} = \dot{x} = 0$ and (1) reduces to

$$F(x, u) = 0. \tag{2}$$

In pull-in/lift-off analysis, the state-of-the-art simulators select a monotonic DC input sequence u_k ($k = 1, 2, \cdots$), and then solve $F(x_k, u_k) = 0$ for x_k by Newton's method using x_{k-1} as an initial guess. This idea works for many MEMS designs, but it is not robust. First, selecting a good initial guess is not easy when state jumping occurs. Second, Eq. (2) normally has multiple solutions, and the DC-sweep method may converge to the wrong one. Finally, the Jacobian matrix around pull-in/lift-off is ill-conditioned or even singular, causing Newton's method to fail. Sometimes the DC-sweep method can achieve convergence around the pull-in/lift-off points "by luck", but at excessive large computational cost.

C. Background of Continuation Methods

Continuation algorithms have been extensively studied and used in non-MEMS communities [32]–[34]. Let $\Psi(y) \in \mathbb{R}^n$ and $y \in \mathbb{R}^{n+1}$, continuation methods try to trace the curve:

$$\Psi(y) = 0. \tag{3}$$

Starting from a "previous" solution point y_{k-1}, a continuation algorithm computes the "next" point y_k by two steps. First, it generates a predictor p_k close to the "next" exact solution y_k. Second, starting from p_k, a corrector is applied to get the exact solution y_k located on curve (3). The predictor and corrector together pose a constraint equation $g(y) = 0 \in \mathbb{R}^1$. Combining this constraint with (3) yields a determined equation

$$\Phi(y) = \begin{pmatrix} \Psi(y) \\ g(y) \end{pmatrix} = 0. \tag{4}$$

One can solve (4) by a standard Newton's method with $y_k^0 = p_k$ as an initial guess to get the exact "next point" y_k. Specifically, one updates the solution by a Newton's iteration

$$\text{solve } \mathcal{J}_k^m \Delta y_k^m = \Phi(y_k^m),$$
$$\text{get } y_k^{m+1} = y_k^m - \Delta y_k^m \tag{5}$$

for $m = 0, 1, 2, \cdots$, until convergence. $\mathcal{J}_k^m \in \mathbb{R}^{(n+1)\times(n+1)}$ is the Jacobian matrix of $\Phi(y)$ at each iteration step.

1) Predictor: Tangent predictor is one of the most popular predictor schemes [32]. With y_{k-1} and its **tangent vector** v_{k-1}, one constructs a tangent predictor by

$$p_k = y_{k-1} + \Delta s \, v_{k-1}. \tag{6}$$

Here $\Delta s > 0$ is a properly selected step size. Graphically v_k is a unit-length vector tangent to the curve (3), defined by

$$M(y_k)v_k = 0, \ \ \|v_k\| = 1, \ \text{with } M(y_k) = \left.\frac{\partial \Psi(y)}{\partial y}\right|_{y=y_k} \tag{7}$$

with $\text{rank}\,(M(y_k)) = n$ if y_k is not a bifurcation point or it is a saddle node [32]. One can first get a vector w_k by solving

$$\begin{pmatrix} M(y_k) \\ v_{k-1}^T \end{pmatrix} w_k = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{8}$$

and then get v_k by normalization:

$$v_k = w_k / \|w_k\|_2. \tag{9}$$

2) Corrector: Here we introduce the perpendicular corrector and the arclength corrector, which are graphically shown in Fig. 2 and will be used in our MEMS simulator.

With p_k and v_{k-1}, a perpendicular corrector seeks y_k under the constraint: $\Delta y_k = y_k - p_k$ is perpendicular to v_{k-1} [32]. Therefore, the constraint function $g(y)$ is constructed as

$$g(y) = v_{k-1}^T (y - p_k), \tag{10}$$

the Jacobian of which is

$$\partial g(y)/\partial y = v_{k-1}^T. \tag{11}$$

In arclength corrector, the solution point y_k should locate on the following sphere [32]:

$$\mathbb{C}(y_{k-1}, \Delta s) = \{\, y \mid \|y - y_{k-1}\|_2 = \Delta s\,\}. \tag{12}$$

Consequently, the constraint function is written as

$$g(y) = \|y - y_{k-1}\|_2 - \Delta s. \tag{13}$$

The Jacobian of this constraint function is

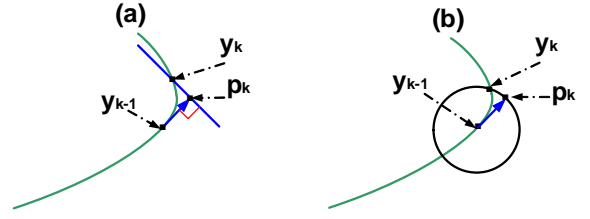$$\partial g(y)/\partial y = \frac{(y - y_{k-1})^T}{\|y - y_{k-1}\|_2}. \tag{14}$$

Fig. 2: (a) perpendicular corrector, (b) arclength corrector. The blue solid arrows represent $\Delta s v_{k-1}$.

3) Limitation in MEMS Simulation: Standard continuation methods have been employed to estimate the pull-in voltages [29]–[31] based on some simplified MEMS models or for some specific devices. In [31], multiple solution points are manually calculated to analyze the stability of a one-dimensional MEMS model. Such applications of continuation to MEMS simulation are limited because: 1) contact effects are not properly modeled, leading to abrupt changes in the structure of the governing equations; 2) existing continuation packages (c.f. [34]) use ordinary differential equations (ODEs) as a model description, which can correctly compute the static solutions of MEMS described by differential algebraic equations (DAEs) but may lead to erroneous stability analysis. Therefore, an automatic CAD framework to simulate both the pull-in and lift-off behavior of general industrial MEMS designs is highly desirable.

III. Two Continuation-Based Methods

In our simulator, a MEMS device is constructed from parametric component libraries. Such libraries can be behavior/analytical models or macromodels from high-order FEM/BEM analysis. Based on their physical locations, the components are automatically connected and then the 2nd-order equation (1) is constructed based on a network analysis. For details, we refer the readers to [38], [39].

To solve the pull-in/lift-off problem, we treat the DC input u in (2) as an unknown variable and obtain a solution curve

$$\Psi(y) = F(x, u) = 0, \ \text{with } y = [x; u] \in \mathbb{R}^{n+1}. \tag{15}$$

which forms a connected set similar to that in Fig 1b.

A. Method 1: Complete-Plot Analysis

Our first method appropriately modifies a standard continuation method to compute the whole solution curve for $u \in [u_{\min}, u_{\max}]$, where $u_{\min}$ and $u_{\max}$ are the minimum and maximum DC inputs of interest, respectively. Doing so avoids the discontinuity and state jumping of pull-in and lift-off that is challenging for previous algorithms.

1) Modeling the Contact Force: The contact force should be modeled carefully to make the governing equation smooth enough. The importance of smoothness has been well recognized in circuit simulation, but it has not attracted enough attention in the MEMS community. Consequently, discontinuous models are frequently used in MEMS simulation, making continuation algorithms infeasible to track the lift-off behavior.
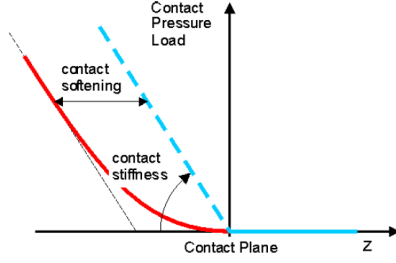
Fig. 3: Illustration of contact softening ($z < 0$ means that two layers make contact).

(a) Pull-in analysis. (b) Lift-off analysis.

Fig. 4: Hybrid continuation for pull-in and lift-off analysis.

The contact force is applied as a pressure load that is linearly dependent on the penetration of the two contact layers [40]. Both materials making contact are treated as compliant materials with elastic modulus E_1, E_2 and thickness H_1 and H_2, respectively. The effective contact stiffness per unit area of contact is given by

$$c = E_1 E_2 / (E_1 H_2 + E_2 H_1). \tag{16}$$

This stiffness is used to compute a contact pressure (force per unit area of contact) only when the materials are in contact (i.e., $z < 0$ in Fig. 3). The pressure-displacement relationship is shown by the blue dashed line in Fig 3, with effective stiffness c being the slope. However, this contact force is not differentiable at $z=0$. To improve the convergence of Newton's iterations that rely on the continuity of the first-order derivative, the corner at $z=0$ is smoothed with an exponential delay as shown by the red line. The "contact softening" distance shown in Fig 3 is 50 nm by default but can be chosen smaller in order to be very small value compared to the thicknesses and gaps of a given design.

2) Stability of the MEMS Static Solutions: In order to analyze the stability of the computed static solution $y=[x;u]$, one must consider the following DAE that is equivalent to (1):

$$\mathbf{E}(z)\dot{z} = \mathbf{F}(z,u), \text{ with } \mathbf{E}(z) = \begin{bmatrix} \mathbf{K}(x) & \mathbf{M}(x) \\ \mathbf{I}_n & 0 \end{bmatrix}$$
$$z = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \text{ and } \mathbf{F}(z,u(t)) = \begin{bmatrix} -F(x,u) \\ \dot{x} \end{bmatrix}. \tag{17}$$

Linearizing (17) around $z = [x,0]^T$ yields a linear model:

$$\mathbf{E}(z)\dot{\tilde{z}} = \mathbf{A}(z)\tilde{z}, \text{ with } \mathbf{A}(z) = \begin{bmatrix} J_{\mathrm{DC}}(y) & 0 \\ 0 & \mathbf{I}_n \end{bmatrix}, \tag{18}$$

where $J_{\mathrm{DC}}(y) = -\frac{dF(x,u)}{dx}$, and $\tilde{z}$ is a small deviation from z. The stability of $y=[x;u]$ is checked by computing the bounded generalized eigenvalues of the matrix pencil $(\mathbf{A}(z), \mathbf{E}(z))$. It is trivial to show that this matrix pencil may have positive generalized eigenvalues even if all eigenvalues of $J_{\mathrm{DC}}(y)$ have negative real parts. When (x_k, u_k) is a turning point of (15), it is a bistable equilibrium of ODE $\dot{x}=F(x,u)$ and of our MEMS equation (1). At this point, $J_{\mathrm{DC}}(y_k)$ is singular, and the linear system (18) has a pole at zero. As a result, we only need to check the stability of one point after pull-in/lift-off, since pull-in/lift-off is a bistable point that represents possible changes in the stability of solution branches. In practice, we
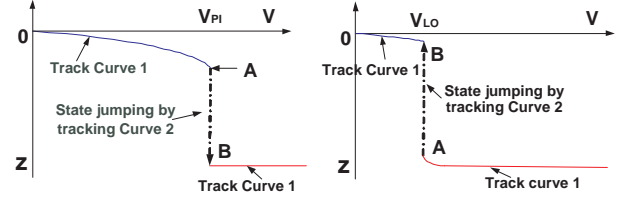
only need to check the bounded generalized eigenvalue with the largest real part (denoted as σ_{mr}).

In order to compute σ_{mr}, we first build a shifted matrix pencil $(\mathbf{A}(z), \mathbf{E}(z) - \epsilon\mathbf{A}(z))$, where the shift parameter ϵ is a small positive scalar. The shift parameter maps all unbounded generalized eigenvalues of $(\mathbf{A}(z), \mathbf{E}(z))$ onto the left-hand complex plane but has little influence on its bounded eigenvalues. Then we compute the generalized eigenvalue with the maximum real part for this shifted matrix pencil, and use it as an approximation to σ_{mr}. Some iterative algorithms [41], [42] have been well developed based on deflated Arnoldi iterations. Such algorithms can efficiently solve the eigenvalue problem for the shifted matrix pencil. Therefore, the computational overhead caused by the stability checking is negligible.

3) Implementation Issues: Our experience shows that the perpendicular corrector can fail around pull-in/lift-off where sharp turning points appear; and the arclength corrector may cause improper branch jumping when a large step size is used. Therefore, our simulator selects the correctors adaptively: when $\Delta s \geq \Delta s_{\mathrm{th}}$, the perpendicular corrector is used; otherwise, we use the arclength corrector. Our simulator uses 10^{-3} as a default value for Δs_{th}.

Point y_k can be identified as a pull-in/lift-off point if the voltage changes from increasing to decreasing, or vice versa:

$$v_{k+1}(n+1)v_k(n+1) < 0, \tag{19}$$

implying that the stability may be changed when the state is switched from y_k to y_{k+1}.

B. Method 2: Hybrid Continuation Approach

This subsection presents an alternative approach that computes the pull-in and lift-off voltages with high precision and then directly computes the jump in state that occurs in a typical capacitance-voltage curve. It is motivated by some specific needs in MEMS design: 1) many MEMS designers prefer a solution curve from monotone DC inputs that is physically more intuitive than the solutions generated by the first algorithm; 2) sometimes the pull-in/lift-off points need to be calculated with high accuracy (e.g., when analyzing the sensitivity of pull-in/lift-off with respect to a design parameter; 3) for some devices (e.g., gyroscopes), the designers are only interested in the pull-in behavior, and thus only sweep-up solution curves are needed. It is hard to tell which method is superior over the other, since their performances depend on the designer's needs and the specific design cases.

1) New Curve for State Jumping: Denote the pull-in point A in Fig. 4a by $y_A = [x_A; u_A]$ and a point B on the solution branch after pull-in as $y_B = [x_B; u_B]$, where $u_B = u_A + \Delta u$ and $\Delta u > 0$ is a perturbation of the DC input. Picking an already computed point on the continuous solution branch **before** point A, we form a new curve by

$$\Psi(y) = (1 - \lambda) F(x, \mu_1) + \lambda F(x, \mu_2) = 0 \qquad (20)$$

with $y = [x; \lambda] \in \mathbb{R}^{n+1}$, $\mu_1 \leq u_A$ and $\mu_2 = u_B$. Starting from $\lambda = 0$ and applying a continuation method, we can trace the curve $y = [x; \lambda]$ until $\lambda = 1$, obtaining point B. Note that curve (20) is different from that used in the two-parameter Homotopy circuit simulator [35]. Furthermore, $F(x, \mu_1)$ is a nonlinear function rather than a linear one. Adding a linear term does not help much in solving our MEMS problems, which is also observed in previous circuit problems [35]. To trace this curve we use the algorithm in Section III-A, with the function and Jacobian evaluations modified accordingly.

2) Basic Simulation Flow: We denote the curves in (15) and (20) as **Curve 1** and **Curve 2**, respectively. Here is the basic flow of pull-in analysis: 1) we first use continuation (rather than DC sweep) to track Curve 1 until a jumping point (e.g., point A in Fig 4a) is approached; 2) by tracking Curve 2 with continuation, we obtain a solution point after state jumping (Point B in Fig 4a); 3) starting from Point B, our simulator tracks Curve 1 again, until another jumping point appears. As shown in Fig 4b, the only difference in lift-off analysis (if required) is keeping DC input monotonically decreasing.

3) Accurate Detection of Pull-in/Lift-Off Points: The pull-in/lift-off points must be automatically detected such that the simulator can switch correctly from Curve 1 to Curve 2. Let y_k be a pull-in/lift-off point, then $\text{rank}(J(y_k)) = n - 1$ and

$$w_k(n + 1) = v_k(n + 1) = 0. \qquad (21)$$

Our simulator sets a small threshold $\epsilon_1 > 0$ and regards y_k as a pull-in or lift-off point if

$$|v_k(n + 1)| < \epsilon_1. \qquad (22)$$

Since in this hybrid scheme we are not interested in the unstable solutions, we shall prevent passing over the pull-in/lift-off points when tracking Curve 1. To achieve this, the solution point y_k is disregarded if $v_k(n+1)v_{k-1}(n) < 0$, then the step size Δs is reduced to recompute a new y_k that satisfies the constraint in (22). For some complex MEMS switches with extremely thin dielectric layers, the curvature around the lift-off points may be very small and the solution curve is still rather flat after scaling. As a result, the last component of v_k may decrease slowly even if the solution point y_k is very close to the lift-off. In this case, y_k is also regarded as a lift-off point if the step size is below a small threshold:

$$\Delta s < \epsilon_2, \text{ with } \epsilon_2 > 0. \qquad (23)$$

By default, we set $\epsilon_1 = 10^{-4}$ and $\epsilon_2 = 10^{-5}$.

IV. NUMERICAL RESULTS

A. Simulation Setup

The proposed algorithms have been implemented and integrated within the commercial MEMS/IC co-design tool MEMS+ [36], which combines high-order mechanical finite elements with a commercial implementation of the MEMS behavioral modeling ideas of [9]–[12]. As a hybrid finite-element/behavioral platform, it can simulate a wide range of MEMS devices both for MEMS design and MEMS/IC co-design. Given a user-created "3D schematic" representing the design and the corresponding material data, the design tool constructs the nonlinear equations as in (1) and the corresponding Jacobians. This system is then accessible either for MEMS design in MATLAB, or for MEMS/IC co-design as a component in system and IC simulators (e.g., MATLAB Simulink [43] or Cadence Spectre [44]). Designs are typically represented with 1 to 1000 degrees of freedom, depending on the desired level of accuracy.

We report the simulation results for an RF-MEMS capacitor [45] and a 4-terminal nano-electro-mechanical (NEM) relay [2]–[5]. To validate the proposed algorithms, we compare the simulation results of the first example with measurement data and the second example with CoventorWare [37]. We compare our continuation-based simulation with the DC-sweep simulation in MEMS+ to show the robustness and efficiency. In MEMS+, the maximum iteration number of the Newton's method in continuation-based algorithms is set as 5, versus 100 in the DC-sweep based simulation approach. For stability checking, the MATLAB routine "**eigs**" is employed to compute the generalized eigenvalue with the largest real part for the shifted matrix pencil. The total CPU time required for stability checking is below 1 second for each MEMS example. Such time is negligible compared with the overall computational cost.

B. Tunable RF-MEMS Capacitor

We first consider a tunable RF capacitor [45]. As shown in the left part of Fig. 5, this capacitive switch has a movable long beam with an octagonal capacitor at the center. On one side (left in the 3-D schematic) of the movable beam, the top part works as an RF conducting path. On the other side (right part of the schematic), the electrode underneath the beam works as the primary actuation electrode, while the remaining area not taken up by the conducting path is used as a secondary actuation electrode. Small bumps (shown by red in the 3-D schematic) underneath the electrodes are used to improve the reliability and limit the contact. The bump height influences not only the shape of the structure after contact, but also the entire CV curve as investigated in [38]. In this work, we consider only the design case with a 0.4-μm bump height to demonstrate the proposed algorithms. This device is fabricated using IBM 0.18-μm CMOS process, and the die photos are shown in the right part of Fig. 5. The fabrication process is detailed in [46]. Since this device has a symmetric structure, we construct a model for only half of the design with symmetric boundary conditions. This model has one J-beam component, 15 electrode contacts and 22 shell components, leading to a nonlinear system with 855 degrees of freedom.

1) Complete-Plot Continuation in MEMS+: Using the complete-plot pull-in/lift-off simulation algorithm described in Section III-A, the hysteresis curve is obtained after 99 seconds
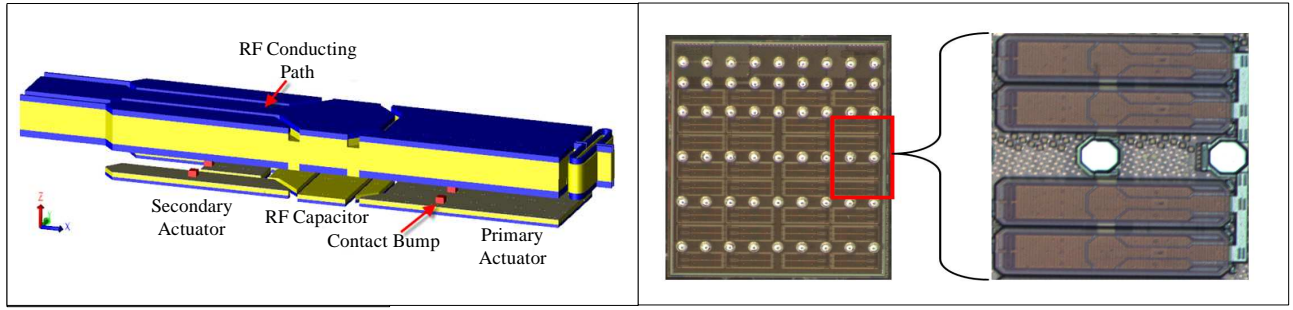
Fig. 5: Left: 3-D schematic of the RF capacitor. Right: die photo and a cell of 4 RF capacitors.
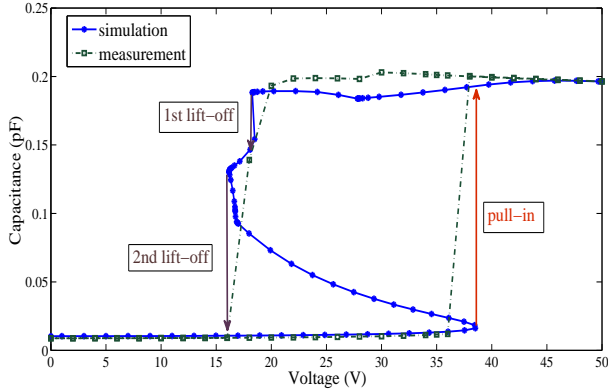

Fig. 6: Pull-in/lift-off result for the RF capacitor, by complete-plot pull-in/lift-off analysis.
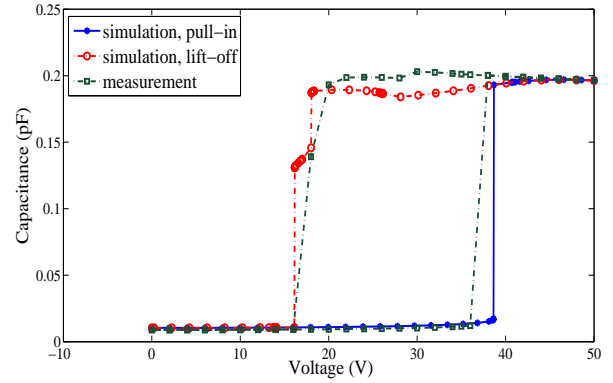

Fig. 7: Pull-in/lift-off result for the RF capacitor, using the hybrid continuation method.

and 420 Newton iterations in total. This curve consists of 88 data points, with smaller step sizes around the pull-in/lift-off points. Fig. 6 compares the simulation results with the measurement data provided by WiSpry [47]. The simulated pull-in voltage is 38.52 V, which is close to the measurement result (between 36 V and 38 V). The capacitance after pull-in is about 0.20 pF, which agrees with the measurement. The simulated curve shows a first lift-off point at 18.27 V leading to an abrupt reduction of the capacitance to 0.145 pF at 18 V. In the measurement, we observe a first lift-off between 18 V and 20 V and a capacitance value of 0.139 pF at 18 V. The secondary lift-off point from our simulation is 16.16 V, after which the movable beam is completely released. The measured secondary lift-off point is between 16 V and 18 V, which again matches our numerical simulation.

2) Hybrid Continuation in MEMS+: The hybrid continuation algorithm in Section III-B generates the pull-in and lift-off curves separately as shown in Fig 7, which is also very close to the measurement data. The pull-in curve consists of 39 data points with much smaller step sizes around state jumping, which is obtained at the cost of 262 seconds and 484 iterations in total. The lift-off curve has 57 data points, which is obtained after 382 seconds of CPU time and 758 Newton iterations. Note that the more intuitive results are obtained at the cost of more CPU time, since in this algorithm we trace multiple curves by continuation methods. The computed pull-in voltage is 38.634 V, and the computed first and secondary lift-off voltages are 18.061 V and 16.164 V, respectively.

3) Comparison with DC Sweep in MEMS+: We use 2 V as a step size to perform a DC sweep simulation between 0~50 V. The sweep-up simulation converges fast before pull-in since the solution curve is very smooth and flat, but it fails to converge at 40 V. This implies a pull-in voltage between 38V and 40 V, which agrees with our continuation-based simulations but is different from the measurement results. The mismatch between the MEMS+ model and the practical fabricated device is mainly caused by the fabrication process variation. In order to obtain the lift-off curve, a sweep-down simulation is performed, which fails at the first starting point 50 V. For this example, increasing the maximum number of iterations does not help achieve convergence, since the updated solution becomes unbounded.

C. 4-Terminal NEM Relay

This 4-terminal nano-electro-mechanical (NEM) relay shown in Fig. 8 is used in ultra low-power VLSI design [2]–[5]. It models the U-shaped springs with beam elements, the gate with shell elements, and the electrostatics with conformal-mapping based elements, leading to 192 degrees of freedom in total. The shells contribute most to state variables and are necessary in order to capture the overly-driven case described below and the effect of stress in the gate. The device is switched by applying a voltage between gate and body to pull the moving plate toward the substrate to make a connection between the contact dimples and the drain/source. We set the voltages at the drain, source and substrate (body) nodes to 0V,
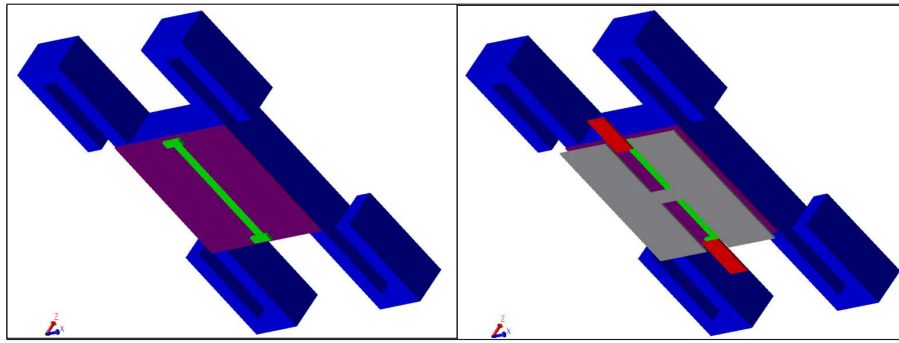
Fig. 8: The 3-D schematics of the 4-Terminal NEM Relay (exaggerated by $10\times$ in z-direction). The left part shows the moving part only. The contact dimple and channel are shown in green and the gate in blue. The right part adds the fixed contact electrodes (source and drain) in red and actuation (body) electrodes in gray.
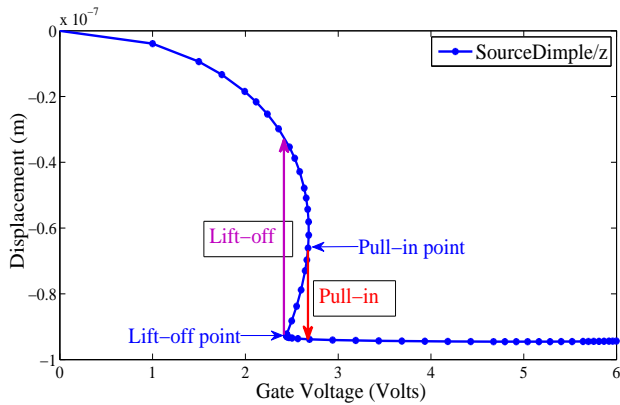

Fig. 9: Pull-in/lift-off result for the "properly-driven" NEM Relay, using complete-plot pull-in/lift-off analysis.

and we consider the gate voltage as a variable. Due to the symmetry of the device and of the inputs, the position of the drain contact is the same as the source contact. This device is expected to turn on under high gate voltage and turn off under low gate voltage. Driving the gate with a higher voltage than pull-in can improve switching times, but care must be taken that the gate is not driven so high as to cause the gate to collapse onto the body. To that end we investigate both contact pull-in/lift-off and collapse pull-in/lift-off.

1) Complete-Plot Continuation in MEMS+: First we consider the "properly-driven" case by setting $u_{min} = 0V$ and $u_{max} = 6V$ (i.e., the maximum gate voltage). This simulation costs 150 seconds and 479 Newton iterations, generating 59 solution points. The results produced by our complete-plot continuation show that the pull-in and lift-off voltages are 2.68V and 2.44V, respectively. The pull-in and lift-off behavior is shown in Fig. 9. The results show that the NEM relay can be switched on and off correctly when $u_{max} = 6V$. Next we consider the much more complicated "overly-driven" case by setting $u_{max} = 10V$. For this case, the complete-plot pull-in/lift-off analysis generates 380 solution points at the cost of 802 seconds and 2886 iterations. Multiple state jumpings are observed from the solution curve in Fig. 10. Increasing the gate voltage from 0V to 10V, the sweep-up curve is: O → A → B → C → D → E→ F→ G. The bottom images of this

device are given in the right of Fig. 10. At point A, the contact dimples do not touch drain or source and thus the device is off. When the gate voltage is further increased, the contact dimples touch the drain and source and turn the device on (point B). If the gate voltage is further increased, after point C the plate collapses onto the body electrodes while the contact dimples rise slightly, losing contact with the drain and source (point D). Similarly, when the gate voltage decreases from 10V to 0V, the sweep-downward curve is: G → H → I → J → K → O, implying that the relay is always off. This observation indicates that to ensure correct switching, the gate voltage should not be higher than 6V. Due to the many solution branches, getting the pull-in and lift-off curves needs some knowledge of nonlinear dynamic theory, and thus a hybrid continuation simulation may be necessary for some users.

2) Hybrid Continuation in MEMS+: Now we use the hybrid continuation method to simulate both the properly and overly driven cases. The simulation results for the properly driven case is shown in Fig. 11, which costs 383 seconds and 903 steps in total (250 seconds and 491 iterations for pull-in analysis, 133 seconds and 412 iterations for lift-off analysis). The resulting pull-in and lift-off curves consist of 38 and 42 points, respectively. Clearly, the devices are correctly turned on and off by controlling the gate voltage. The pull-in and lift-off voltages are calculated accurately without regard for the simulator's step size to be 2.683V and 2.442V, respectively.

For the overly-driven design, the hybrid continuation simulation generate 77 points at the cost of 674 seconds and 1260 iterations for pull-in analysis, and 43 points at the cost of 886 seconds and 983 iterations for lift-off simulation. The displacement of the source dimple is intuitively shown in Fig. 12. In this plot, it becomes obvious that:

1) When the gate voltage is increased from 0V to 10V, the contact dimples touch drain and source after 2.683V. However the contact is lost after 6.091V although the contact dimple lowers down slightly again after 8.928V.
2) When the gate voltage is reduced from 10V to 0V, the contact dimples always lose contact with drain and source. When the gate voltage reduces to 1.684V, the contact dimples rise slightly, and it suddenly jumps up as the gate voltage reduces to 1.120V.

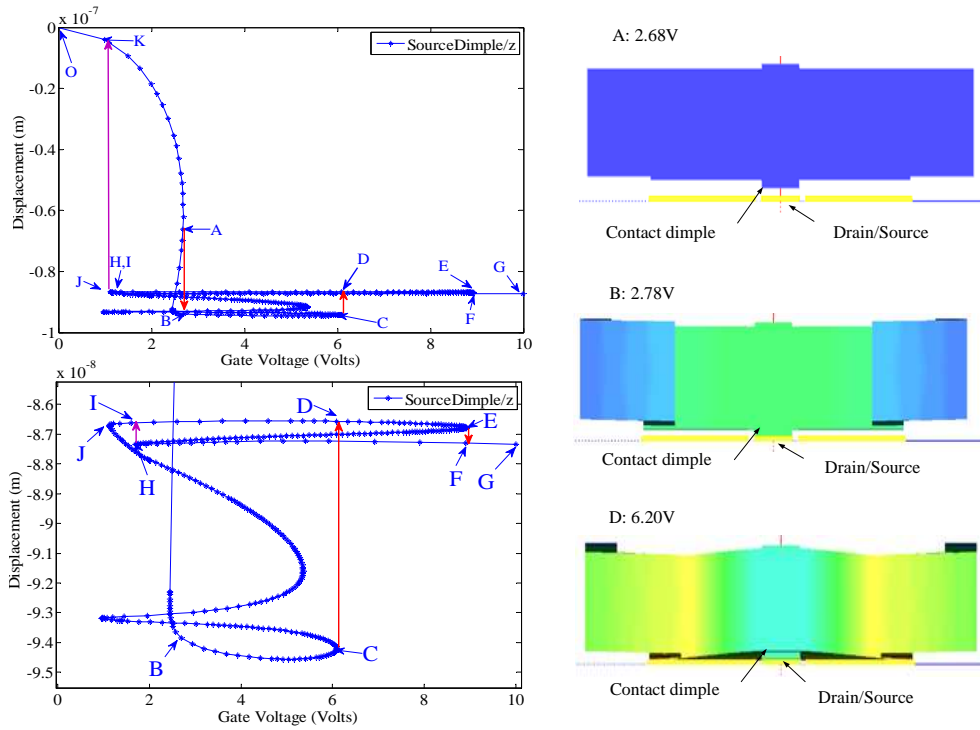The observed state jumpings are consistent with our results

Fig. 10: Pull-in/lift-off result for the "overly-driven" NEM Relay, by complete-plot pull-in/lift-off analysis. The plot in the left bottom is a zoom-in showing the multiple pull-in/lift-off points. The right part shows the positions of the contact dimple while the gate voltage is monotonically increased, which is viewed from the source/drain terminal (i.e., along the channel) and is exaggerated by $20\times$ in z-direction.
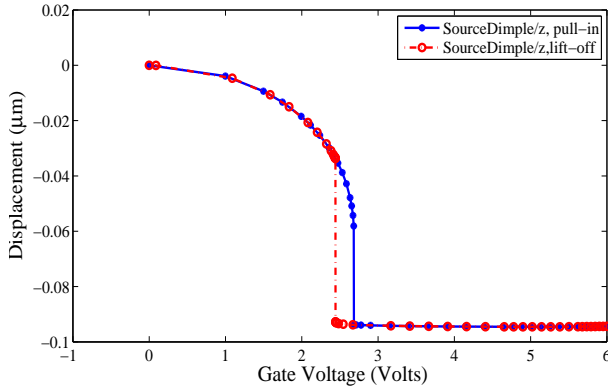

Fig. 11: Pull-in/lift-off result for the "properly-driven" NEM Relay, using hybrid continuation method.
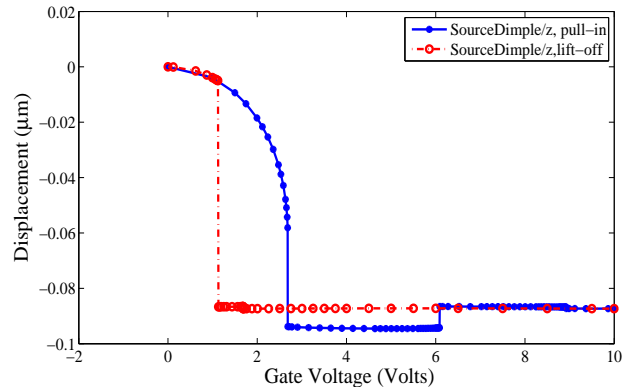

Fig. 12: Pull-in/lift-off result for the "overly-driven" NEM Relay, using the hybrid continuation method.

from complete-plot continuation plotted in Fig. 10.

3) Comparison with DC Sweep in MEMS+: For the properly-driven case, the DC sweep-up fails around pull-in when the step size is below 0.2V. For example, with a input step size of 0.1V the sweep-up simulation cannot proceed to u=2.7V after 320s. Meanwhile, this implies that the pull-in occurs around 2.7V, which agrees with the results from our continuation-based simulations. The sweep-down simulation fails at the first point u=6V after the maximum number of iterations. The DC-sweep method cannot handle the overly-driven case. With the input sequence u=[0:0.5:10]V, the DC

sweep-upward simulation fails at u=6.5V. The sweep-down simulation fails to converge for the first DC input u=10V regardless of the choice of the sweep step size. In addition, when the input step size is reduced below 0.5V, it becomes even more difficult to achieve convergence.

4) Comparison with CoventorWare: To validate the result, we compare our algorithms with the pull-in/lift-off analysis in the latest version of CoventorWare [37]. In Coventor-Ware, the MEMS designs are described by a coupled finite-element/boundary-element (FEM/BEM) formulation, which can model the geometry in full detail for highly accurate

results. CoventorWare computed the pull-in point for the NEM relay to be between 2.74V and 2.75V. This result is slightly different from our simulation results from complete-plot pull-in/lift-off analysis (with pull-in at 2.68V) and hybrid continuation method (with pull-in at 2.683V), resulting in an relative error of 2.4%. The lift-off voltage from CoventorWare for the properly-driven case is 2.53-2.56V. Therefore, the relative error of the results from our continuation algorithms is about 4.6%. If desired, the errors of pull-in/lift-off analysis could be reduced by using higher-order elements in MEMS+ at the cost of greater simulation time. The FEM/BEM formulation in CoventorWare generates an entirely different nonlinear system with an extremely large problem size, and thus our FEM/behavioral hybrid MEMS+ platform will naturally be orders of magnitude faster (seconds or minutes in MEMS+ versus several hours in CoventorWare).

Remarks. Although the complete-plot pull-in/lift-off analysis costs less CPU time over the hybrid continuation scheme for the examples shown in this paper, but in practice it is case-dependent. Our experience shows that for some devices with complex solution spaces, the hybrid continuation algorithm can be faster than the complete-plot continuation. For some MEMS designs (such as resonators) where lift-off is not important, the simulation can be terminated once the pull-in is found to further reduce the simulation time.

V. CONCLUSIONS

This paper has presented two continuation-based methods to simulate both the pull-in and lift-off behavior for industrial MEMS design. These two methods are implemented in a hybrid FEA/analytical-model platform to facilitate design verification. The first one produces a complete plot of multiple solution branches, and the second one produces more intuitive single-solution curves. Both algorithms are robust and can avoid the convergence difficulties around pull-in/lift-off that cannot be handled by previous algorithms. Our proposed algorithms have been tested on a tunable MEMS RF capacitor and a 4-terminal NEM relay, showing better robustness than the DC-sweep simulation. Our algorithms have been validated by practical measurement data and by the commercial MEMS simulator CoventorWare, respectively.

ACKNOWLEDGEMENTS

REFERENCES

[1] S. D. Senturia, *Microsystem Design*. Boston, MA: Kluwer Academic, 2000.

[2] R. Nathanael, V. Pott, H. Kam, J. Jeon, and T.-J. K. Liu, "4-terminal relay technology for complementary logic," in *Proc. Intl. Electron Devices Meeting*. Baltimore, Mar. 2010, pp. 1–4.

[3] H. Kam, T.-J. K. Liu, V. Stojanovic, D. Markovic, and E. Alon, "Design, optimization, and scaling of MEM relays for ultra-low-power digital logic," *IEEE Trans. Electronic Devices*, vol. 58, no. 1, pp. 236–250, Jan. 2011.

[4] F. Chen, M. Spencer, R. Nathanael, C. Wang, H. Fariborzi, A. Gupta, H. Kam, V. Pott, J. Jeon, T.-J. K. Liu, D. Markovic, V. Stojanovic, and E. Alon, "Demonstration of integrated micro-electro-mechanical (MEM) switch circuits for VLSI applications," in *Proc. Intl. Solid-State Circuit Conference*. San Francisco, CA, Feb. 2010, pp. 150–151.

[5] F. Chen, H. Kam, D. Markovic, T.-J. K. Liu, V. Stojanovic, and E. Alon, "Integrated circuit design with NEM relays," in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 2008, pp. 750–757.

[6] S. Pacheco, L. Katehi, and C. Nguyen, "Design of low actuation voltage RF MEMS switch," in *Proc. IEEE Microwave Symp.*, Jun. 2000, pp. 165–168.

[7] L. J. Hornbeck, "Digital light processing and MEMS: an overview," in *Proc. IEEE/LEOS Summer Topic Meetings*, Aug. 1996, pp. 5–7.

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

[9] G. Fedder and Q. Jing, "A hierarchical circuit-level design methodology for microelectromechanical systems," *IEEE Trans. Ciruits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 10, pp. 1309–1315, Oct. 1999.

[10] J. V. Clark, N. Zhou, and K. S. J. Pister, "MEMS simulation using SUGAR v0.5," in *Proc. Solid-State Sensors Actuators Workshop*. Hilton Head Island, June 1998, pp. 191–196.

[11] M. Maher and H. Lee, "MEMS systems design and verification tools," in *Proc. SPIE Smart Structures Materials*. San Diego, CA, Mar. 1998, pp. 40–48.

[12] G. Lorenz and R. Neul, "Network-type modeling of micromachined sensor systems," in *Proc. Intl. Conf. Modeling Simulation of Microsystems*, Apr. 1998, pp. 233–238.

[13] T. Mukherjee and G. K. Fedder, "Structured design of microelectromechanical systems," in *Proc. Design Automation Conference*, Jun. 1997, pp. 680–685.

[14] T. Mukherjee, "Design automation isseus for biofluidic microchips," in *Proc. Intl. Conf. Computer-Aided Design*, Nov. 2005, pp. 462–469.

[15] T. Kurzweg, S. Levitan, P. Marchand, J. A. Martinez, K. Prough, and D. Chiarulli, "A CAD tool for optical MEMS," in *Proc. Design Automation Conference*. New York, NY, Jun. 1999, pp. 879–884.

[16] N. Yazdi, F. Ayazi, and K. Najafi, "Micromachined inertial sensors," *IEEE Proceedings*, vol. 86, no. 8, pp. 1640–1659, Aug. 1998.

[17] Y. Lin, S. Lee, Z. Ren, and C. Nguyen, "Series-resonant micromechanical resonator oscillator," in *Proc. Intl. Electron Devices Meeting*, Dec. 2003, pp. 39.4.1–39.4.4.

[18] X. Cai, H. Yie, P. Osterberg, J. Gilbert, S. Senturia, and J. White, "A relaxation/multipole-accelerated scheme for self-consistent electromechanical analysis of complex 3-D microelectromechanical structures," in *Proc. Intl. Conf. Computer-Aided Design*. Santa Clara, CA, Nov. 1993, pp. 270–274.

[19] H. Yie, X. Cai, and J. White, "Convergence properties of relaxation versus the surface-Newton generalized-conjugate residual algorithm for self-consistent electromechanical analysis of 3-D micro-electromechanical structures," in *Proc. Numerical Modeling of Processes and Devices for Integrated Circuits*. Honolulu, Hawaii, June 1994, pp. 137–140.

[20] P. Osterberg, H. Yie, X. Cai, J. White, and S. Senturia, "Self-consistent simulation and modeling of electrostatically deformed diaphragms," in *Proc. Micro Electro Mechanical Systems*. Oiso, Japan, Jan. 1994, pp. 28–32.

[21] A. Alwan and N. Aluru, "Analysis of hybrid electrothermomechanical microactuators with integrated electrothermal and electrostatic actuation," *IEEE Journal of Microelectromechanical Systems*, vol. 18, no. 5, pp. 1126–1136, Oct. 2009.

[22] S. Pamidighantam, R. Puers, K. Baert, and H. Tilmans, "Pull-in voltage analysis of electrostatically actuated beam structures with fixed-fixed and fixed-free end conditions," *Journal of Micromechanical Microengineering*, no. 12, pp. 458–464, 2002.

[23] O. Degani, E. Socher, A. Lipson, T. Leitner, D. Setter, S. Kaldor, and Y. Nemirovsky, "Pull-in study of an electrostatic torsion microactuator," *IEEE Journal of Microelectromechanical Systems*, vol. 7, no. 4, pp. 373–379, Dec. 1998.

[24] H. Sadeghian, G. Rezazadeh, and P. Osterberg, "Application of the generalized differential quadrature method to the study of pull-in phenomena of MEMS switches," *IEEE Journal of Microelectromechanical Systems*, vol. 16, no. 6, pp. 1334–1340, Dec. 2007.

[25] D. Elata, "Modeling the electromechanical response of RF-MEMS switches," in *Proc. EusoSime*, Apr. 2006, pp. 1–4.

[26] E.-R. Konig and G. Wachutka, "Multi-parameter homotopy for the numerical analysis of MEMS," *Senors Actuators A*, no. 110, pp. 39–51, 2004.

[27] E. Konig and G. Wachutka, "Analysis of unstable behaviour occuring in electromechanical microdevices," in *Proc. Modeling Simulation of Microsystems*, Apr. 1999, pp. 330–333.

[28] J. Bielen and J. Stulemeijer, "Efficient electrostatic-mechanical modeling of C-V curves of RF-MEMS switches," in *Proc. EusoSime*, Apr. 2007, pp. 1–6.

[29] K. Das and R. Batra, "Symmetry breaking, snap-through and pull-in instability under dynamic loading of microelectromechanical shallow arches," *Smart. Mater. Struct.*, vol. 18, pp. 1–15, 2009.

[30] D. Bernstein, P. Guidotti, and J. Pelesko, "Mathematical analysis of an electrostatically actuated MEMS devices," in *Proc. Modeling and Simulation of Microsystems*, 2000, pp. 489–492.

[31] J. Liu, "Nonlinear dynamics of a dual-backplate capacitive MEMS," PhD Dissertation, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, 2007.

[32] R. Seydel, *Practical Bifurcation and Stability Analysis: from Equilibrium to Chaos*. New York: Springer-Verlag, 1994.

[33] A. Ushida and L. O. Chua, "Tracing solution curves of nonlinear equations with sharp turning points," *Intl. Journal of Circuit Theory and Applications*, vol. 12, no. 1, pp. 1–21, Jan. 1984.

[34] A. Dhooge, W. Govaerts, and Y. Kuznetsov, "MATCONT: a MATLAB package for numerical bifurcation analysis of ODEs," *ACM Trans. Mathematical Software*, vol. 29, no. 2, pp. 141–164, June 2003.

[35] J. Roychowdhury and R. Melville, "Delivering global DC convergence for large mixed-signal circuits via homotopy/continuation methods," *IEEE Trans. Computer-Aided Design*, vol. 25, no. 1, pp. 66–78, Jan. 2006.

[36] http://www.coventor.com/products/mems/.

[37] http://www.coventor.com/products/coventorware/.

[38] M. Kamon, S. Maity, D. DeReus, Z. Zhang, S. Cunningham, S. Kim, J. McKillop, A. Morris, G. Lorenz1, and L. Daniel, "New simulation and experimental methodology for analyzing pull-in and release in MEMS switches," in *Proc. IEEE Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS)*, Jun. 2013.

[39] G. Schröpfer, G. Lorenz, S. Rouvillois, and S. Breit, "Novel 3D modeling methods for virtual fabrication and EDA compatible design of MEMS via parametric libraries," *Journal of Micromechanics and Microengineering*, vol. 20, no. 6, p. 064003, Jun. 2010.

[40] "MEMS+ user's mannual," Coventor, Inc.

[41] R. B. Lehoucq and D. Sorensen, "Deflation techniques for an implicitly re-started Arnoldi iteration," *SIAM J. Matrix Analysis and Applications*, vol. 17, no. 4, pp. 789–821, Oct. 1996.

[42] D. Sorensen, "Implicit application of polynomial filters in a k-step Arnoldi method," *SIAM J. Matrix Analysis and Applications*, vol. 13, no. 1, pp. 357–385, Jan. 1992.

[43] http://www.mathworks.com.

[44] http://www.cadence.com.

[45] D. R. Dereus, S. Natarajan, S. J. Cunningham, and A. S. Morris, "Tunable capacitor series/shunt design for integrated tunable wireless front end applications," in *Proc. IEEE Micro Electro Mechanical Systems*, Jan. 2011, pp. 805–808.

[46] A. K. Stamper, C. V. Jahnes, S. R. Depuis, A. Gupta, Z.-X. He, R. T. Herrin, S. E. Luce, J. Maling, D. R. Miga, W. J. Murphy, E. J. White, S. J. Cunningham, D. R. Dereus, I. Vitomirov, and A. S. Morris, "Planar MEMS RF capacitor integration," in *Proc. IEEE Solid-State Sensors, Actuators and Microsystems Conference (TRANSDUCERS)*, Jun. 2011, pp. 1803–1806.

[47] http://www.wispry.com/.
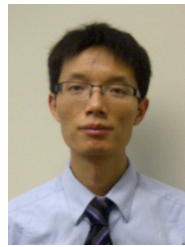
Zheng Zhang (S'09) received the B.Eng. degree from Huazhong University of Science and Technology, China, in 2008, and the M.Phil. degree from the University of Hong Kong, Hong Kong, in 2010. He is a Ph.D student in Electrical Engineering at the Massachusetts Institute of Technology (MIT), Cambridge, MA. His research interests include uncertainty quantification (UQ), numerical methods for the computer-aided design (CAD) of integrated circuits and microelectromechanical systems (MEMS), and model order reduction.

In 2009, Mr. Zhang was a visiting scholar with the University of California, San Diego (UCSD), La Jolla, CA. In 2011, he collaborated with Coventor Inc., working on CAD tools for MEMS design. He was recipient of the Li Ka Shing Prize (university best M.Phil/Ph.D thesis award) from the University of Hong Kong, in 2011, and the Mathworks Ph.D Fellowship from MIT, in 2010.

Mattan Kamon received the B.S. degree in engineering science and the M.A. degree in mathematics in 1991 from the Pennsylvania State University, University Park, and the M.S. and Ph.D. degrees in electrical engineering and computer science in 1994 and 1998, respectively, from the Massachusetts Institute of Technology, Cambridge. For his graduate work, he developed efficient algorithms for 3-D electrical interconnect parameter extraction and simulation. One result of that work is FastHenry, a public-domain inductance extraction tool.

He joined Coventor, Inc. after graduation, where he switched focus to numerical algorithm development for MEMS. Since joining he has either contributed to or led the development of all of Coventor's simulation tools, including tools for electromechanical and magnetomechanical simulation, squeeze-film damping, reduced-order modeling, and coupled MEMS-IC co-simulation. He is currently Principal Technologist at Coventor where he leads prototype and algorithm research for semiconductor process modeling. He holds patents in both MEMS simulation and semiconductor processing.

Luca Daniel (S'98-M'03) received the Laurea degree (*summa cum laude*) in electronic engineering from the Universita di Padova, Italy, in 1996, and the Ph.D. degree in electrical engineering from the University of California, Berkeley, in 2003.

He is an Associate Professor in the Electrical Engineering and Computer Science Department of the Massachusetts Institute of Technology (MIT), Cambridge. His research interests include accelerated integral equation solvers and parameterized stable compact dynamical modeling of linear and nonlinear dynamical systems with applications in mixed-signal/RF/mm-wave circuits, power electronics, MEMs, and the human cardiovascular system.

Dr. Daniel received the 1999 IEEE Transactions on Power Electronics best paper award, the 2003 ACM Outstanding Ph.D. Dissertation Award in Electronic Design Automation, five best paper awards in international conferences, the 2009 IBM Corporation Faculty Award, and 2010 Early Career Award from the IEEE Council on Electronic Design Automation