Process variation analysis for MEMS design

Luca Schenato

December 23, 1999

Abstract

Process variations, incurring during the fabrication stage of MEMS structures, may lead to substantially different performances from the nominal ones. This is mainly due to the small variation of the geometry of the structure with respect to the ideal design. In this paper we propose an approach to estimate performance variations for planar suspended MEMS structure for low frequency applications. This approach is based on two complementary techniques, one probabilistic and the other deterministic. The former technique, based on the Monte-Carlo algorithm, defines a random distribution on the geometric variables and evaluates the possible outcome performance by sampling that distribution. The latter technique, instead, based on Ellipsoidal Calculus, finds bounds on performance parameters given the bounds on the geometric variables, i.e. it considers the worst case scenario. Both techniques have been integrated with SUGAR, a simulation tool for MEMS devices available to the public.

1 Introduction

MEMS is a fast growing micromachining technology intended to fabricate tiny sensors, actuators and mechanisms and to integrate them with electronics, giving rise to very large and complex systems. Several MEMCAD tools are appearing to help the design both at device-level and at the system-level. [6].

However, most of them cannot take into account possible variation of design after fabrication process, which may lead to non negligible effect on the overall structure behavior.

Our intent is to address this issue by estimating bounds on the variation of performance specifications arising from process variation in the geometric design variables.

There are several methods intended to deal with the problem of optimization with uncertain data. . The *stochastic* approach models uncertainties as random variables as in the Monte-Carlo algorithms [5], and gives estimates of the probability distribution on the performance parameters. The *robust optimization* approach, as in the ellipsoidal calculus, instead, tries to estimate the worst case among infinite number of scenarii.

They are not mutually exclusive, and the designer should be given the possibility to choose between them, depending on the application. These methods have been implemented and integrated into SUGAR1.0, a MEMS simulation software developed at UC Berkeley [3] and public ally available [2].

The next section describes the model adopted for planar suspended MEMS structure and describes the effect of process variation on the performance. Sections 3 and 4 respectively describe the Monte-Carlo and the Ellipsoidal Calculus algorithms and their implementations. In the Section 5 several simulations are presented for common test structures. Finally, Section 6 offers some concluding remarks and proposes future extensions.

2 Model

We are following the same methodology adopted in SUGAR to analyze a surface micromachined MEM structure. This methodology is based on a nodal analysis: it decomposes the whole structure into atomic elements, such as beams, anchors, gaps, comb drives, whose behavior is described by a simple lumped dynamical system (Figure 1). Finally these atomic elements are connected together (Figure 2), giving rise to a internally coupled dynamical system. The displacements variables of each element are modeled by a nonlinear second order mechanical model as follows:

$$[m(\mathbf{x},\mathbf{q})]\ddot{\mathbf{q}} + [d(\mathbf{x},\mathbf{q},\dot{\mathbf{q}})]\dot{\mathbf{q}} + [k(\mathbf{x},\mathbf{q})]\mathbf{q} = \mathbf{f}(\mathbf{q}) \qquad (1)$$

where $\mathbf{q} = (x_1, y_1, \theta_1, x_2, y_2, \theta_2)^T$ is the vector of nodal displacements, and $\mathbf{x} = (L_1, w_1, t_1)^T$ is the vector of design variables, which correspond to the geometric dimensions. The matrices $[m], [d], [k]$ are respectively the effective mass, damping and spring stiffness of the dynamical system and they depend nonlinearly both in the design variables and displacement vector.

Since in this project we are considering only the DC equilibrium state, we can neglect the mass and damping term, simplifying Equation (1) as

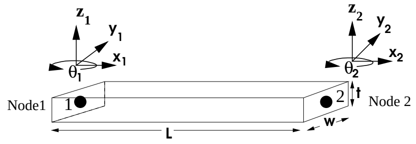

Figure 1: *The atomic element description of a beam. The geometric variables, L, w, t, correspond to design variables. The displacements $x_1, y_1, \theta_1, x_2, y_2, \theta_2$ are related to the design specification and performance parameters.*

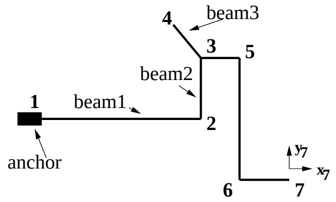

Figure 2: *A simple MEMS structure where the nodes are explicitly enumerated*

$$[k(\mathbf{x}, \mathbf{q})]\mathbf{q} = \mathbf{f_{DC}}(\mathbf{q}) \tag{2}$$

For small displacements the force vector, $\mathbf{f_{DC}}(\mathbf{q})$, and the stiffness matrix, $[k]$, are independent of $\mathbf{q}$. In this case the stiffness matrix, $[k]$, takes the following expression:

$$[k(\mathbf{x})] = \frac{E}{L^3}\begin{bmatrix} wtL^2 & 0 & 0 & -wtL^2 & 0 & 0 \\ 0 & 12I & 6IL & 0 & -12I & 6IL \\ 0 & 6IL & 4IL^2 & 0 & -6IL & 2IL^2 \\ -wtL^2 & 0 & 0 & wtL^2 & 0 & 0 \\ 0 & -12I & -6IL & 0 & 12I & -6IL \\ 0 & 6IL & 2IL^2 & 0 & -6IL & 4IL^2 \end{bmatrix} \tag{3}$$

where E is the Young modulus for the beam and I is the moment of inertia of the beam given by:

$$I = \frac{tw^3}{12} \tag{4}$$

Since each structure may have different orientations, all local coordinates are transformed into global coordinates by a rotation matrix, $T(\alpha)$, where α is the orientation of the beam measured counterclockwise from the positive x axis [3]. The stiffness matrix in global coordinates is thus given by $[k]_{global} = [T]^T[k]_{local}[T]$.

The assemblage of the set of individual matrices $[k]$ into the collective system matrix, [K], and vectors $\mathbf{f_{DC}}$ into $\mathbf{F_{DC}}$, where all structures are coupled at common nodes, is accomplished by nodal superposition. Thus, the entire system DC equilibrium can be expressed as

$$[K(\mathbf{x})]\mathbf{q} = \mathbf{F_{DC}} \tag{5}$$

where the nodal displacement $\mathbf{q} = (x_1, y_1, \theta_1, ..., x_n, y_n, \theta_n)^T$ is the vector of all nodal displacements, and $\mathbf{x} = (L_1, w_1, t_1, ..., L_n, w_n, t_n)^T$ is the vector including all geometric design variables, and $\mathbf{F_{DC}}$ is the vector of applied forces and torque. Note that this is a common linear equation of the form $Ay = b$, if the geometry vector, $\mathbf{x}$, is fixed.

If $\mathbf{x}_{nom}$ is fixed, the previous system has (in general) a unique equilibrium solution, $\mathbf{q}_{nom}$, which can be efficiently computed by SUGAR. However, due to process variations the design variables may be slightly different from their nominal values, thus giving rise to different equilibrium solution. This phenomenon can be characterized mathematically by adding an unknown perturbation term, $\delta\mathbf{x}$, to the nominal value of the design variables, $\mathbf{x}_{nom}$, i.e. $\mathbf{x} = \mathbf{x}_{nom} + \delta\mathbf{x}$, where $\delta\mathbf{x}$ belongs to a know set $\mathcal{U}$. Therefore Equation (5) must be rewritten as

$$[K(\mathbf{x}_{nom}) + \Delta K]\mathbf{q} = \mathbf{F_{DC}} \tag{6}$$

where $K(\mathbf{x}_{nom})$ depends only on the nominal design vector and is thus known, while ΔK is the unknown stiffness that depends on the vector $\delta\mathbf{x}$. It is now clear that the Equation (6) maps the set $\mathcal{U}$ into the set of all possible displacements $\mathcal{Q}$, which , of course, include the vector $\mathbf{q}_{nom}$. Though this mapping seems harmless, it results to be very complex and it is hopeless to solve analytically.

One option to tackle this problem is to define a probability distribution for the vector $\delta\mathbf{x}$ and then to solve Equation (6) for many samples drawn from that distribution. This is the underlying idea of Monte-Carlo algorithms described in the next section.

Another possibility would be to calculate an ellipsoid of confidence which encloses *all* possible equilibrium points, $\mathbf{q}$, as shown in Figure 3. This can be done via semidefinite-programming (SDP) as proposed in [4]. The main difficulty for this approach is finding an appropriate form for the matrix ΔK in order to apply this algorithm. This is discussed in Section 5.

3 Monte-Carlo Algorithm

For this approach, modeled the process variation assuming $\delta\mathbf{x}$ to be a vector of independent random variables. Only the width and thickness variations are considered, since length is unlikely to be affected by process variation.

In order to have a flexible simulation we allowed three different random distribution: 1) a *uniform* distribution the interval $(x_{nom} - \delta, x_{nom} + \delta)$, where δ is a parameter defined by the user, 2) a *Gaussian* distribution with mean x_{nom} and variance δ), 3) a *uniform corner* dis-

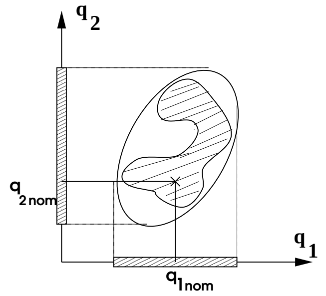

Figure 3: *The shaded region corresponds to the set of possible equilibrium points due to process variation. Note that even if the nominal designed equilibrium belongs to this set, it is not generally centered in the middle of the solution set. Ellipsoidal calculus tries to find the smallest ellipsoid that encloses this set, thus giving bounds on displacement variations*

tribution where $x = x_{nom} \pm \delta$ takes value only at the extremes of the interval.

The first two distributions are motivated by their ubiquity use in modeling parameters variation in practical applications. The last distribution is interesting because the largest variations on the performance parameters are the effect of some combinations of those geometry. In our simulations we considered only a fixed δ for all beams.

Since the number of samples is finite, it is necessary to give probabilistic bounds on the estimates of performance. These bound are carachterized by two parameters, the *accuracy*, ϵ, and the *confidence*, $1-\eta$. Suppose, for example, that we have drawn l number of samples and we have calculated the fraction of samples, r_l, which have a performance between $\pm 20\%$ the nominal performance. Then, we can say that the true fraction of samples, r, satisfies the following ralation :[7]

$$P[|r - r_l| > \epsilon] < 2 \exp^{-2\epsilon^2 l} = \eta, \tag{7}$$

where P stands for probability. Therefore, given any desired accuracy and confidence, we can find the minimum number of samples that must be drawn to achieve them, as follows:

$$l > -\frac{1}{2\epsilon^2} \ln(\frac{\eta}{2}) \tag{8}$$

4 Confidence Ellipsoid Algorithm

As mentioned above, a different approach consists in finding the smallest ellipsoid which includes all possible solutions, $\mathbf{q}$, of Equation (6), as shown in Figure 3. This can by done through the solution of an SDP problem as long as the matrix ΔK takes a particular form called *linear fractional representation.* A particular case of this representation is the *affine representation*, which has the following expression:

$$(K + L\,\Delta\, R)\mathbf{q} = \mathbf{F_{DC}} \tag{9}$$

where the matrices, K, L and R, are constant and depend on the structure of the uncertainty, i.e. on how the geometric perturbation affects the system matrix. The matrix $\Delta = \{\mathbf{diag}(\delta_1 I_{r_1}, \ldots, \delta_m I_{r_m}) \,|\, \delta \in \mathbf{R}^m, ||\delta||_\infty \leq 1\}$, where m is the number of independent variables and I_{r_i} is the identity matrix of size r_i.

These four matrices can be found easily by considering the local stiffness matrix (3). First, we assume that the length, L, does not vary. This is a reasonable assumption since the relative variation of L is very small in comparison with the thickness, t, and the width, w. Moreover the thickness, t, and the width, w, have maximum variation of $\pm\delta$. Then, we can notice that the elements of the matrix, $[k]$, depends only on the area, $A = wt$, and the moment of inertia, I, of the beam. If we consider these as independent variables, we can rewrite $[k]$ as follows:

$$\begin{aligned} [k] &= \begin{bmatrix} \frac{E}{L} & 0 & 0 & -\frac{E}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{E}{L} & 0 & 0 & \frac{E}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} A + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12E}{L^3} & \frac{6E}{L^2} & 0 & -\frac{12E}{L^3} & \frac{6E}{L^2} \\ 0 & \frac{6E}{L^2} & \frac{4E}{L} & 0 & -\frac{6E}{L^2} & \frac{2E}{L} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12E}{L^3} & -\frac{6E}{L^2} & 0 & \frac{12E}{L^3} & -\frac{6E}{L^2} \\ 0 & \frac{6E}{L^2} & \frac{2E}{L} & 0 & -\frac{6E}{L^2} & \frac{4E}{L} \end{bmatrix} I \\ &= [k_A]A + [k_I]I \end{aligned} \tag{10}$$

where the matrices, $[k_A]$ and $[k_I]$, are constant, and only the area, A, and the moment of inertia, I, may change due to process variations. Since the variables, A and I, belong to an interval, it is more convenient to rewrite equation (10) in terms of the interval mean, $\bar{A} = (A_{max} + A_{min})/2$, and interval half-width, $d_A = (A_{max} - A_{min})/2$, as follows:

$$\begin{aligned} [k] &= \bar{A}[k_A] + d_A[k_A]\,\delta_A + \bar{I}[k_I] + d_I[k_I]\,\delta_I \\ &= [k_{A0}] + [k_{A1}]\,\delta_A + [k_{I0}] + [k_{I1}]\,\delta_I \\ &= [k_0] + L_A(\delta_A I_{r_A})R_A + L_I(\delta_I I_{r_I})R_I \end{aligned} \tag{11}$$

where the matricies, $[k_{A0}]$ $[k_{A1}]$ $[k_{I0}]$ $[k_{I1}]$ and $[k_0]$, are constant. The variables, δ_1 and δ_2, may range in the interval $(-1, +1)$, giving rise to all possible scenarii. The matrix $[k_0] = [k_{A0}] + [k_{I0}]$, is the sum of the two mean matrices, and $[k_{A1}] = L_A\, R_A$ and $[k_{I1}] = L_I\, R_I$, are obtained by running the singular value decomposition (SVD) on the matricies $[k_{A1}$ and $[k_{I1}$. Now, we can see from Equation (11), that the local uncertainty matrix has an affine representation. The global affine representation is simply obtained by stacking all the matrices L_i, δ_i and R_i into a single global matrix as in Equation (9).

The algorithm that computes the confidence ellipsoid from this representation is quite complex and it is described in [4]. We simply mention that it recasts the problem as a Semi Definite Programming (SPD) problem, which can be solved numerically by the lmitool toolbox in Matlab.

Before concluding this section, we want to remark that the two variables δ_A and δ_I are not independent, since the area, A, and the moment of inertia, I, are strictly related. However, assuming they are independent, simply results in adding infeasible solutions. Therefore, the ellipsoid may be loose, i.e. there may be a gap between the set of feasible solutions and the ellipsoid boundary.

5 Simulations

Now we show some simulations of the previous algorithms on several MEMS structures. For the following simulations we chose the following parameters: $\delta = 0.1\mu m$, $\eta = 0.01$, $\epsilon = 0.09$, $l = 328$. Most of the beams in these structures have minimum width and thickness in order to magnify the effect of process variations.

Figure (4) shows a simple dog-bone-shape suspension. This structure is design to be symmetric so that a force applied on the y-axis would generate a displacement only along the y-axis. However, due to variations of geometric variables, this is not true, as shown on the plots where for several samples the rotation angle, θ, is nonzero. In the middle plot of Figure (4) are shown the possible displacements for Node 4 due to a uniform distribution. The ellipsoid of 100 % confidence seems to be over-pessimistic, however it is tight, in the sense that there exist some possible solutions which lie on its boundary, though they are very "unlikely". This is clear in the bottom plot, where we used a uniform corner distribution. In fact we can see that there are solution on the boundary. A fairer comparison between the Ellipsoid approach and the stochastic approach is shown in the folded spring of Figure (5), where we chose a Gaussian distribution instead of a uniform distribution for the middle plot. Another interesting property of the ellipsoid approach is that it capture also the correlation among different displacement variables, as shown in the bottom plot of Figure (5), even though this time the ellipsoid seems to be not so tight.

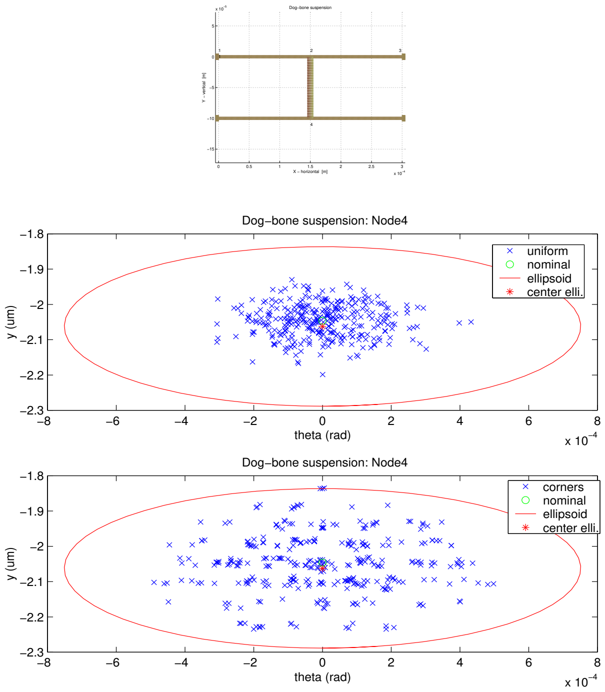

Figure 4: *A Dog-bone-suspension where a force of -100 μN is applied at node 4 along the y-axis. Middle plot: uniform distribution. Bottom plot: uniform corner distribution*

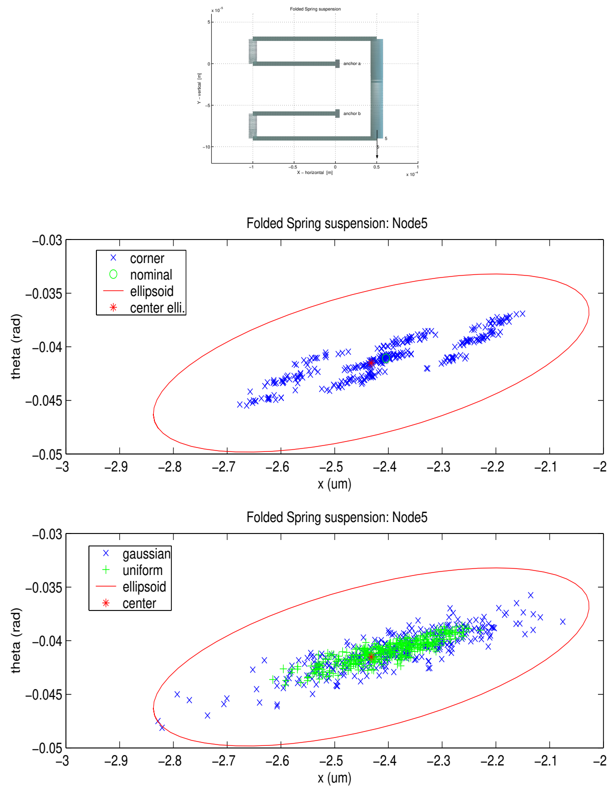

Figure 5: *A folded spring where a force of -100 μN is applied at node 5 along the y-axis. Middle plot: uniform corner distribution. Bottom plot: '+' uniform distribution, 'x' gaussian distribution*

6 Conclusions

In this paper we have proposed and simulated two approaches to analyze the effect of process variations on performance design. These approaches are reasonably flexible, can be easily modified by the user and seems to give meaningful results. All the code used for the simulations has been adapted to run with SUGAR1.0 and can be down-loaded following the instructions in [1]. Easy extensions of our work, with no additional computation expense, would be the introduction of different δ for each geometric variables. However, a limiting factor for the extensive use of our algorithms is the computational time they require. At present they work reasonably fast with few hundreds of sample and less than ten elements. Future work must be directed in the solution of this problem.Another interesting issue is understanding why the ellipsoid approach is sometimes loose, since it may either depends on the Lagrange relaxation [4] or on the approximated affine representation in equation (11).

7 Acknowledgment

The results in this project are part of another class project (*EE245-Introduction to MEMS design*), which has been developed with Wei-Chung Wu. I would like to thank Prof. K. Pister for his suggestions and interest in the project.

References

[1] ftp://home/robo6/sastry/lusche/EE245-MEMS_design/code/README.

[2] http://www-bsac.eecs.berkeley.edu/ cfm/.

[3] K.S.J.Pister J.V. Clark, N. Zhou. Mems simulation using sugar v0.5. In *Technical Digest. Solid-State Sensor Workshop*, pages 191–196, 1998.

[4] G. Calafiore L. El Ghaoui. Confidence ellipsoids for uncertain linear equations with structure. In *Proceeding of the IEEE CDC*, 1999.

[5] D.J.C. MacKay. *Introduction to Monte Carlo methods.* Michael I. Jordan, 1999.

[6] S.D. Senturia. Cad challenges for microsensors, microactuators, and microsystems. *Proceedings of the IEEE*, pages 1111–26, 1998.

[7] V.N. Vapnik. *Statistical learning theory.* Wiley, New York, 1998.