# Numerical Spring Models for Behavioral Simulation of MEMS Inertial Sensors

Sitaraman V. Iyer and Tamal Mukherjee

Electrical and Computer Engineering Department
Carnegie Mellon University. Pittsburgh, PA 15213-3890

## ABSTRACT

Design of springs is a very important step in the design process of inertial sensors. A procedure for computing the spring stiffness for any single-chain configuration of beams and a translator which converts beam-based schematic representation of inertial sensors to higher-level behavioral representation are implemented. Combining the spring stiffness computation with the translator, spring-mass behavioral models of inertial sensors are generated. The behavioral representation is used for rapid design-space exploration. Simulations of the higher-level behavioral schematics are 10 to 100 times faster than simulation of the atomic-elements based schematics and the results match to within 5%.

## 1. INTRODUCTION

Over the past few years a number of micromachined accelerometer and gyroscope designs have been published [1]. A number of inertial sensors are made up of one or more mechanical proof-masses suspended by four springs. Currently, the design of these sensors involves extensive use of finite element analyses (FEA) in mechanical, electrostatic and mixed domains. Generally, performing FEA on 3D structures entails considerable time and effort on part of the designer for meshing the structure as well as CPU time. Additionally, the results of FEA usually need post processing in order to identify the design components which require modification. FEA does not lend itself naturally to a hierarchical design paradigm, wherein, the effect of changes in the layout geometry is rapidly translated to changes in system performance or even on the performance contributions of the different behavioral components of the system.

There are on-going efforts to establish a hierarchy of design levels for MEMS [2]. The basis for the hierarchy is decomposition of MEMS devices into MEMS atomic elements such as plate masses, beam springs, electrostatic air-gaps and anchors which are at a similar level as resistors, capacitors and inductors in the electronics design hierarchy. A schematic representation of MEMS using such elements bears a strong correspondence to the underlying layout. At higher design levels, a chain of beam springs can be combined to form crab-leg springs, u-shaped springs or serpentine springs. At an even higher (behavioral) level, all the springs which connect two rigid elements (for instance, a plate and an anchor) can be lumped together into a single behavioral *spring* element. At this level the behavioral representation cannot be visually correlated to the layout of the device. However, this level closely approximates the spring-mass-damper way of visualizing an inertial sensor, which the designer is trying to realize through the MEMS device. In order to simulate the performance of an entire system like a 3 axis inertial measurement unit, behavioral models of the inertial sensors which encapsulate the non-idealities will be highly useful.This design hierarchy lends more insight into the design problem by capturing the effects of geometry in high-level behavioral parameters such as spring stiffness, damping coefficients and comb-drive sensitivities. A seamless translation between the different levels will allow the designer to investigate more complex topologies and rapidly identify the critical components in a design.

In this paper, the use of spring models at the behavioral level and translation from layout geometry and atomic-element (e.g. *plate*, *beam* etc.) based schematics to higher-level behavioral representation of spring-mass systems is demonstrated. There is prior work aimed at building behavioral models through FEA [3]. The commercial tool MEMCAD also has an in-built methodology for generating macromodels of springs [4]. This work is different in the sense that it utilizes the specialized geometry of single-chain-of-beams springs in which every beam is connected to at most one other beam at each end. Such springs are commonly used in inertial sensors. Each spring can be represented by a lumped-element stiffness matrix. Stiffness matrices (which are composed of analytical models for each stiffness constant) have been derived previously for beams [5], crab-leg, u-shaped and serpentine springs [6][7]. However, there are a wide variety of spring topologies and it is practically impossible to pre-derive the stiffness matrix for each of them. In order to cater to new spring topologies, we have implemented an algorithm, based on energy methods, to compute the spring stiffness matrix for any single-chain configuration of beams. The stiffness constants thus obtained are used as parameters of a higher-level *spring* element.

Correspondence: Sitaraman Iyer. Other Author Information: S. Iyer: Email: sita@ece.cmu.edu; Telephone: 1-412-268-4275.
T. Mukherjee: Email: tamal@ece.cmu.edu; Telephone: 1-412-268-8522

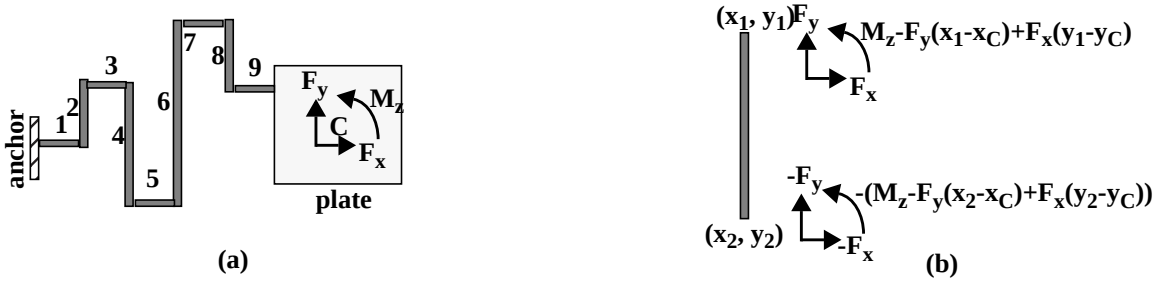


**FIGURE 1. (a) Spring with single-chain of 9 beams attached to a plate. C is the point of application of force. The other end of the spring is anchored. (b) Free-body diagram of beam 6 and the bending moment along beam 6.**

The atomic-elements based schematics and the higher behavioral schematics can be built using the NODAS tool developed in CMU [8]. NODAS (NOdal Design of Sensors and Actuators) is a framework which implements the hierarchical representation and simulation of MEMS. In this framework schematics of MEMS sensors can be created using elements such as *beams*, *plates*, *anchors* and *comb-drives* and electrical and mechanical independent sources. Different analyses such as DC, AC and transient analysis can be performed on the MEMS sensors. A behavioral *spring* element has been implemented which is compatible with lower-level *beam* elements in NODAS. The procedure mentioned above is used to compute the parameters which go into the behavioral spring element.

The computation of the spring stiffness matrix is presented first. This is followed by a description of the algorithms for translation from low-level schematic to higher-level behavioral schematic. Following this, verification of the spring computation procedure is done by comparison with FEA. The usefulness of the simulation at the higher level is then demonstrated by design-space exploration of two inertial sensors. Finally discussions and conclusions from the results obtained are presented.

## 2. SPRING STIFFNESS COMPUTATION

Figure 1 shows a spring composed of 9 beams in a single-chain configuration attached to a plate at one end and anchored at the other end. The procedure for computing the spring stiffness matrix is described below. A force (or moment) is applied to the point C, in the direction of interest, and the displacement is calculated symbolically (as a function of the design variables and the applied force).When forces (moments) are applied at the end-points of the flexure, the total energy of deformation, *U*, is calculated as:

$$U = \sum_{beam\ i=1}^{N} \int_0^{L_i} \frac{M_i(\xi)^2}{2EI_i} d\xi \tag{1}$$

where, $L_i$ is the length of the *i*'th beam in the flexure, $M_i$ is the bending moment transmitted through beam *i*, *E* is the Young's modulus of the structural material and $I_i$ is the moment of inertia of beam *i*, about the relevant axis. The bending moment is a linear function of the forces and moments applied to the end-points of the flexure. Further, for a single chain of beams, as shown in Figure 1(a), the bending moment and, therefore, the energy stored in a beam, depends only on the position of the end-points of the beam relative to the point of application of force C. The displacement of point C in any direction $\zeta$ is given as:

$$\delta\zeta = \frac{\partial U}{\partial F_\zeta} \tag{2}$$

where, $F_\zeta$ is the force applied in that direction [9]. Similarly, angular displacements can be related to applied moments. Since, the moment is linearly dependent on the applied forces and moments, the displacement is also a linear function of the applied forces, i.e.,

$$\delta\zeta_i = \sum_{\xi} \alpha_{\zeta\xi i} F_{\xi i} \tag{3}$$

where $\delta\zeta_i$ is a generalized displacement (includes translation and rotation), $F_{\xi i}$ is the generalized force in the direction $\xi$ and $\alpha_{\zeta\xi i}$, the compliance of the *i*'th beam. The displacement is derived using the equation for the moment in the beam. The total displacement is obtained by summing up the individual displacements of each beam. This is possible only for a single-chain configuration of beams, because at any point in the spring, there is no splitting of forces into different beams. Thus, we can obtain the overall displacements as linear functions of the applied forces with the compliance matrix composed of $\alpha_{\zeta\xi} = \sum_i \alpha_{\zeta\xi i}$ as the coefficients. Solving this matrix equation for different displacements gives the stiffness matrix [5].

The compliance matrix for a beam is:

$$\begin{bmatrix} \frac{l(y_1^2+y_2^2+y_1(y_2-3y_C)-3y_2y_C-3y_2y_C+3y_C^2)}{3EI_z} & -\frac{l(x_1(2y_1+y_2-3y_C)+x_2(y_1+2y_2-3y_C)-3x_C(y_1+y_2-2y_C))}{6EI_z} & \frac{l(y_1+y_2-2y_C)}{2EI_z} \\ -\frac{l(x_1(2y_1+y_2-3y_C)+x_2(y_1+2y_2-3y_C)-3x_C(y_1+y_2-2y_C))}{6EI_z} & \frac{l(x_1^2+x_2^2+x_1(x_2-3x_C)-3x_2x_C-3x_2x_C+3x_C^2)}{3EI_z} & -\frac{l(x_1+x_2-2x_C)}{2EI_z} \\ \frac{l(y_1+y_2-2y_C)}{2EI_z} & -\frac{l(x_1+x_2-2x_C)}{2EI_z} & \frac{l}{EI_z} \end{bmatrix}$$

where,

$(x_1,y_1)(x_2,y_2)$ are the coordinates of the beam end-points

$(x_C,y_C)$ are the coordinates of the point of application of force

$l$ is the length of the beam

$E$ is the Young's Modulus of the material of the beam

$I_z$ is the moment of inertia of the beam cross-section about the *z*-axis

The above matrix is for the in-plane forces and displacements only. The actual implementation also computes the stiffness for the out-of-plane directions.

## 3. TRANSLATION FROM ELEMENT LEVEL SCHEMATIC TO BEHAVIORAL SCHEMATIC

The layout of a nested resonator system is shown in Figure 2(a). A similar topology has been employed in a gyroscope designed earlier [10]. The system is composed of an inner resonator which is suspended inside a movable rigid frame. The inner resonator consists of four suspension springs and a central proof-mass. The frame is suspended by four springs which are anchored at the outer ends. The micromechanical part of the layout is passed through a MEMS layout extractor [11]. The layout extractor recognizes the different components in the layout and generates the NODAS schematic representation of the nested-resonator system shown in Figure 2(b). The different components which are used in the schematic are rigid *plates*, flexible *beams* and attachment points called *anchors*. In the schematic shown, the central proof-mass is modeled by a single *plate* at the center. This *plate* is connected through four identical chains of *beams* to the rigid frame. The rigid frame is modeled by four

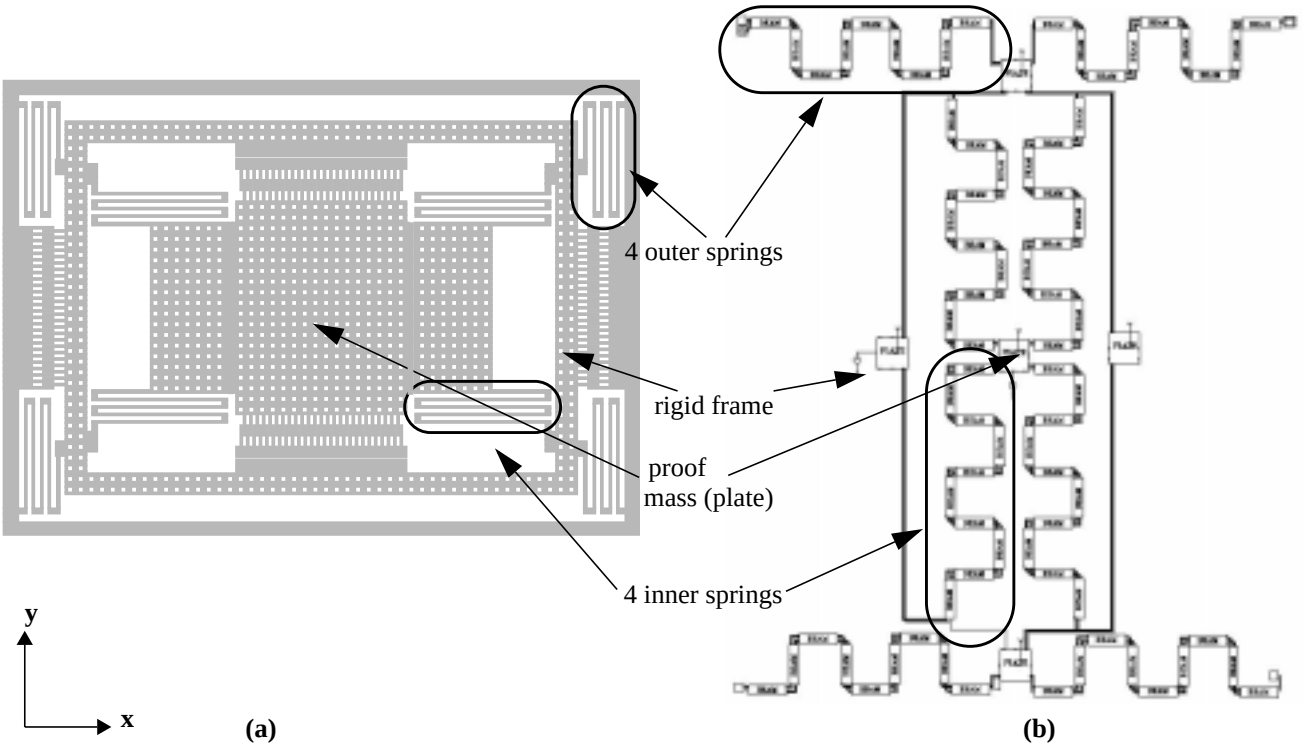


**FIGURE 2. (a) Layout of a nested-resonator system (b) Corresponding NODAS schematic obtained through layout extraction. The schematic consists of a central plate connected through the four inner springs to the frame. The frame is composed of four plates which are suspended by the four outer springs. The other ends of the four outer springs are connected to the chip substrate.**

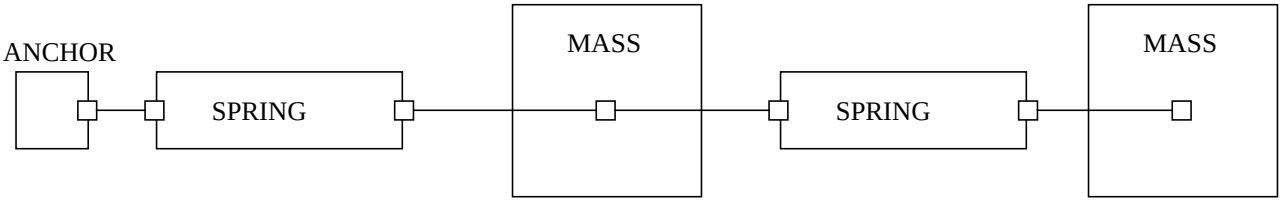


**FIGURE 3. Behavioral model generated from the schematic representation of the nested-resonator system shown in Figure 2(b). The behavioral model comprises only of 2 *springs* and 2 *masses*, while the schematic representation has 5 *plates* and 76 *beams***

*plates*. These *plates* are anchored to the substrate using another set of four identical chains of *beams*. The NODAS schematic representation is the starting point for the translation to the behavioral representation.

The translator implements the following tasks during the conversion to the behavioral representation.

1. Identifies groups of plates which are adjacent, lumps them into a single rigid body and computes the effective mass for this rigid body.
2. Identifies chains of beams and collects these chains into springs.
3. Uses the spring computation procedure described earlier to compute the stiffness matrices for the springs collected in 2.
4. Identifies groups of springs which are connected between the same rigid bodies identified in 1 and sums up their stiffness matrices so that there is at most one composite spring between any two rigid bodies.

When the schematic shown in Figure 2(b)is passed through this translator, the resulting behavioral representation is shown in Figure 3 . The central plate is translated to a *mass* element at the behavioral level. The four plates forming the rigid frame are adjacent. Therefore, they are combined by the translator to form another *mass* element. The four inner springs are combined to form the spring which connects the two *mass* elements in the behavioral model. The four outer springs are combined to form another spring element which connect the *mass* element representing the rigid frame to the anchor.

Figure 4(a) shows the layout of an spring designed for use in a *z*-axis accelerometer. This spring has about 50 beams. The schematic representation of this layout obtained by using the layout extractor is shown in Figure 4(b). Four such springs were used to symmetrically suspend a proof-mass at the center of the layout. The resulting schematic representation is then translated to the behavioral model which is shown in Figure 5. The schematic with all the four springs has more than 200 *beams* in it.

In both the above examples, the nested-resonator and the accelerometer, the behavioral model has much fewer elements com-

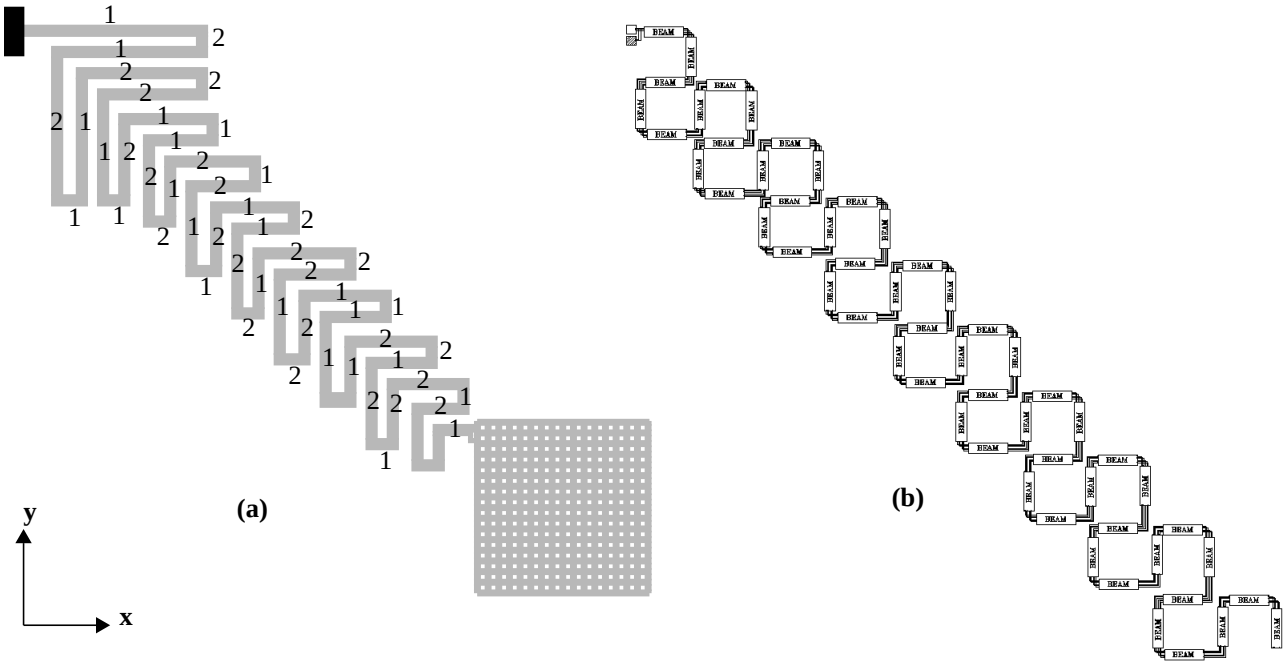


**FIGURE 4. (a) Layout of a spring with about 50 beams connected to a proof-mass at one end and anchored at the other end (b) Corresponding NODAS schematic of the spring obtained through layout extraction. Beams marked with “1” and “2” have widths $w_1$ and $w_2$ respectively.**

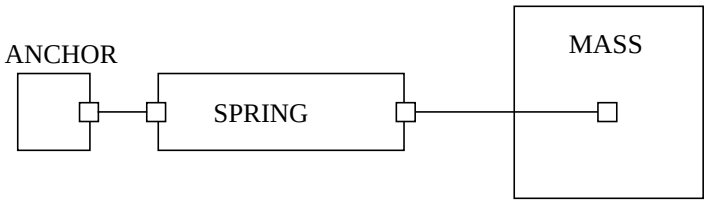


**FIGURE 5. Behavioral model generated from the schematic representation of the accelerometer shown in Figure 4(b). The behavioral model comprises only of a *spring* and a *mass*, while the schematic representation has 1 *plate* and 200 *beams***

pared to the schematic representation. Therefore, it is expected that the simulation of the behavioral model will be considerably faster than the schematic. However, the behavioral model captures only the lowest resonant modes of the structure, whereas the schematic representation shows many more modes.

## 4. VERIFICATION

The methodology described above for behavioral model generation is evaluated at two levels. First, the accuracy of the spring stiffness computation is verified by comparison with FEA. Second, the two topologies described previously are simulated at the schematic level and at the higher behavioral level and the results are compared with respect to the accuracy and the simulation speed.

### 4.1 Verification of Spring Stiffness Computation

The layout of the spring used for verification is shown in Figure 6. The length and the width (measured from centers of the adjacent beams) of the vertical beams are varied while the horizontal beams are unchanged. The comparisons of the spring stiffness computations with the FEA results are shown in Figure 7, Figure 8 and Figure 9 for $k_{xx}$, $k_{yy}$ and $k_{\theta z \theta z}$ respectively. For the range of widths and lengths considered, the match between the FEA and the spring stiffness computation procedure is very good. The error surfaces for $k_{xy}$, $k_{y\theta z}$ and $k_{x\theta z}$ with respect to the beam length and the width are not presented here, but are within 6%. The errors are more prominent at higher values of the beam width *w*. This is because of the ambiguity in measuring beam lengths i.e., whether the length is to be measured from the center of the horizontal beams or from the edge of the horizontal beams. Similarly for the horizontal beams, the effective length is strongly correlated to the beam width since, the beam width (4 μm) is a significant fraction of the length (10 μm). For the comparisons shown the beam length was measured from the edge

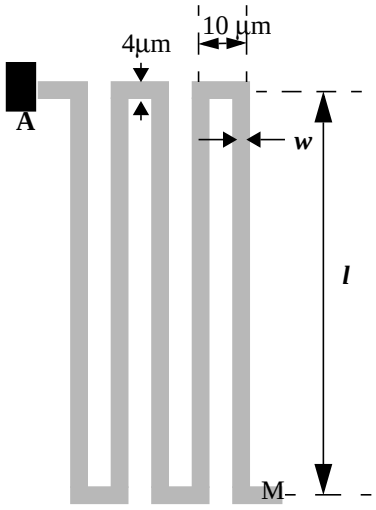


**FIGURE 6. Layout of the spring used for FEA. A is the anchored point. M is the point to which the mass is attached. The length *l* and the width *w* of the vertical beam are varied over a range of values.**

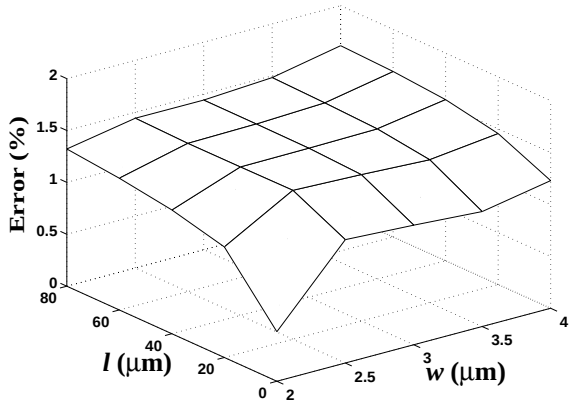


**FIGURE 7. Comparison of spring stiffness computation for $k_{xx}$ with FEA**

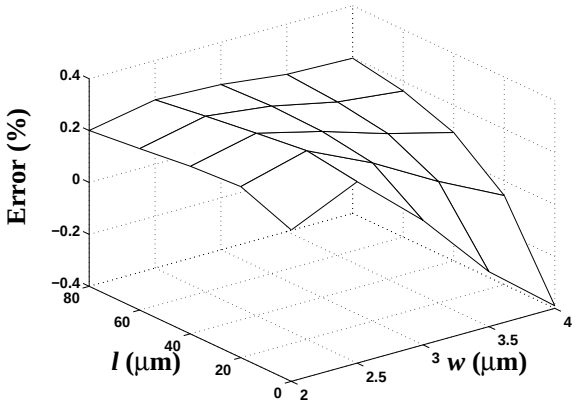


**FIGURE 8. Comparison of spring stiffness computation for $k_{yy}$ with FEA**

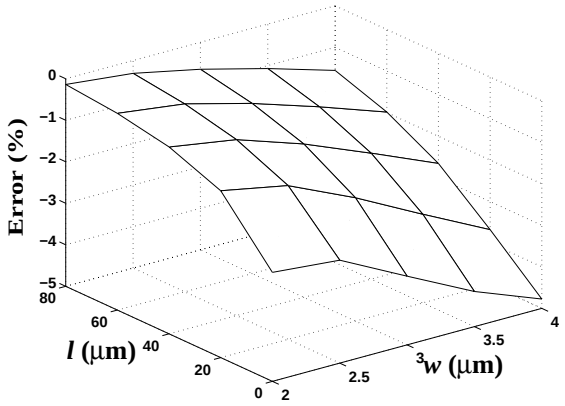


**FIGURE 9. Comparison of spring stiffness computation for $k_{\theta z \theta z}$ with FEA**

of the adjacent beams and a correction of:

$$0.25\left(\frac{w_{adjacent}}{w_{beam}}\right)(w_{adjacent}) \tag{4}$$

is applied to both the ends of each beam. $w_{beam}$ is the width of the beam itself and $w_{adjacent}$ is the width of the adjacent beam.

## 4.2 Comparison of Schematic Simulation with Higher Level Behavioral Simulation

### 4.2.1 Example 1: Nested-Resonator Design Space Exploration

One of the design issues in a gyroscope which uses the nested-resonator topology is the difference in frequencies between the drive-direction resonant mode (*x*-mode of the outer rigid frame) and the sense-direction resonant mode (*y*-mode of the inner proof-mass). For maximizing sensitivity and maintaining manufacturability at the same time, it is desired that the sense resonant mode be higher than the drive resonant mode and also that the sense resonant mode be close enough to the drive resonant mode. The design of the suspension springs is crucial to achieve this goal. One of the ways of controlling the spring stiffness is by changing the widths of the beams forming the spring. By this method the overall dimensions of the spring do not change much and therefore, the design process is simplified.

The widths of the beams in the outer and inner springs were chosen as the two design variables. AC analysis of the schematic

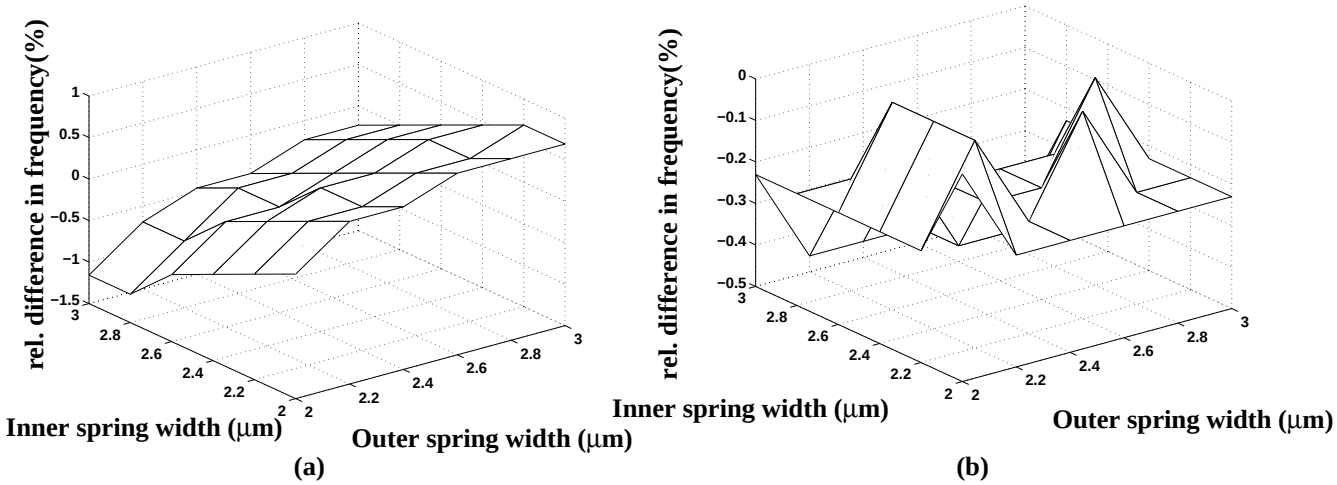


**FIGURE 10. Difference in the resonant frequency extracted from the schematic simulations and behavioral simulations (a) drive-mode (b) sense-mode**

shown in Figure 2(b) was done for a range of these design variables. The dependence of the drive-mode and the sense-mode resonant frequencies on the design variables was obtained from these analyses.

Similarly, these design variables were also passed to the spring computation code and the stiffness of the behavioral spring was calculated for all the settings of these design variables. With these stiffness values AC analysis of the behavioral model of the nested resonator system (shown in Figure 3) was done and again, the dependence of the drive-mode and the sense-mode resonant frequencies on the design variables was obtained. The higher-level behavioral simulation was about 10 times faster than the beam-based schematic simulation. The comparison between the resonant frequencies is shown in Figure 10(a) and (b) for the drive and sense modes respectively. The difference is less than 2% for all values of the design variables. The difference is larger for larger values of the inner spring width. This is because, in the current implementation, only the spring stiffness value is computed. The effective mass contribution of the beams is not taken into account in the behavioral model. From the results obtained it is possible to choose the beam widths so that the design objective is achieved.

#### 4.2.2 Example 2: Resonance Frequency Analysis

The sensitivity of an accelerometer is inversely proportional to the resonance frequency in the sensing direction. The resonance frequency of the other modes determine the mechanical cross-axis sensitivity of the accelerometer. Therefore, it is usually preferred to have the other modes at much higher frequencies.

The schematic built using 4 of the springs shown in Figure 4(b) has more than 200 *beams*. Two different beam widths, $w_1$ and $w_2$, were chosen as the design variables as shown in Figure 4(a). For different settings of these design variables AC analysis was done on both the schematic as well as the behavioral model combined with the spring computation. The higher-level behavioral simulation combined with the spring computation was almost 100 times faster than the beam-based schematic simulation and the resonance frequencies match within 5%. This difference is because the springs are considered massless.

## 5. CONCLUSIONS

A simple method for computing stiffness matrix for springs has been implemented which can handle any single-chain configuration of beams. This method is accurate to within 5% as long as all beams are atleast as long as they are wide. Currently, this procedure does not compute the effective masses for the springs, but this can be incorporated easily. Nodal schematic representations of inertial sensor layouts were generated by use of a MEMS layout extractor. This was coupled with a translator for converting these schematics to a higher level of abstraction involving only *spring* and *mass* elements. Using this flow from layout to schematic to behavioral model, faster design techniques for inertial sensors were demonstrated. Depending upon the complexity of the inertial sensor design, the simulations using the behavioral model coupled with the spring computation procedure were upto 100 times faster than the nodal simulation of the schematic representation using *beams* and *plates.*

## ACKNOWLEDGEMENTS

The authors thank Bikram Baidya for his help with the MEMS layout extractor, Hao Luo and Hasnain Lakdawala for layouts.This research effort is sponsored by the Defence Advanced Research Projects Agency (DARPA) and U. S. Air Force Research Laboratory, under agreement number F30602-96-2-0304. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, the U. S. Air Force Research Laboratory, or the U. S. Government.

## REFERENCES

[1] N. Yazdi, F. Ayazi, and K. Najafi, "Micromachined Inertial Sensors", *Proc. IEEE*, vol. 86, no. 8, pp. 1640-1659, Aug. 1998

[2] T. Mukherjee, “CAD for Integrated MEMS Design”, *Design, Test, Integration and Packaging of MEMS/MOEMS*, Paris, France, May 9-11, 2000

[3] B. F. Romanowicz, *Methodology for the Modeling and Simulation of Microsystems*, Kluwer Academic Publishers, Boston, 1998

[4] MEMCAD 4.6 User Guide, Microcosm Technologies, Inc.

[5] J. S. Przemieniecki, *Theory of Matrix Structural Analysis*, Dover Publishing Inc., New York, 1985

[6] G. K. Fedder, *Simulation of Microelectromechanical Systems*, Ph. D. Thesis, EECS Department, University of California at Berkeley, September 1994

[7] S. Iyer, Y. Zhou, and T. Mukherjee, "Analytical Modeling of Cross-axis Coupling in Micromechanical Springs", *Modeling and Simulation of Microsystems*, San Juan, Puerto Rico, April 19-21, 632-635, 1999

[8] G.K. Fedder, Q. Jing, "A Hierarchical Circuit-level Design Methodology for Microelectromechanical Systems", *IEEE Transactions on Circuits and Systems II*, vol. 46, no. 10, pp 1309-1315, 1999

[9] J. M. Gere and S. P. Timoshenko, *Mechanics of Materials*, PWS Publishing Company, Boston, 4th edition, 1997

[10] M. Kranz, G. K. Fedder, “Micromechanical Vibratory Rate Gyroscopes Fabricated in Conventional CMOS”, *Symposium on Gyro Technology*, Stuttgart, Germany, pp 3.0-3.8, 1997

[11] B. Baidya, S.K. Gupta and T. Mukherjee, "MEMS Component Extraction", *Modeling and Simulation of Microsystems*, San Juan, Puerto Rico, April 19-21, 143-146, 1999