

Modeling and Simulation of Non-idealities in a Z-axis CMOS-MEMS Gyroscope

by

Sitaraman V. Iyer

B. Tech. (Indian Institute of Technology, Bombay) 1996

M. S. (Carnegie Mellon University) 1998

A dissertation submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Thesis Committee:

Dr. Tamal Mukherjee, Chair

Prof. Gary K. Fedder

Prof. L. Richard Carley

Prof. Jonathan Cagan

Dr. Steve Bart

Department of Electrical and Computer Engineering

Carnegie Mellon University

Pittsburgh, Pennsylvania, USA

April 2003

CARNEGIE MELLON UNIVERSITY

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

TITLE: MODELING AND SIMULATION OF NON-IDEALITIES IN A
Z-AXIS CMOS-MEMS GYROSCOPE

PRESENTED BY Sitaraman V. Iyer

ACCEPTED BY THE DEPARTMENT OF Electrical and Computer Engineering

MAJOR PROFESSOR

DATE

DEPARTMENT HEAD

DATE

APPROVED BY THE COLLEGE COUNCIL

DEAN

DATE

Copyright © 2003 Sitaraman V. Iyer
All Rights Reserved

To Amma and Appa

Abstract

MEMS gyroscopes have proved to be extremely difficult to manufacture reliably. The MEMS gyroscope is required to sense picometer-scale displacements, making it sensitive to spurious vibrations and other coupling mechanisms. This thesis aims to quantitatively capture, through models and simulations, the sensitivity of a MEMS gyroscope to manufacturing variations in the widths of suspension beams and gaps between fingers in electrostatic actuation and capacitive sensing combs. The gyroscope considered in this thesis is manufactured in a CMOS-MEMS process. The suspended MEMS structures are composed of the multi-layer stack of interconnect metals and dielectrics in a CMOS process. The effect of misalignment between the metal layers in the suspended microstructures is also modeled in the gyroscope. A number of fundamental issues related to the modeling and simulation of MEMS gyroscopes are addressed. Models in elastic and electrostatic domains are developed. Numerical tools such as finite element analysis or boundary element analysis are used for model verification. Behavioral simulation is used throughout this thesis to analyze the gyroscope and system-level design issues.

The elastic modeling effort is primarily aimed at a thorough understanding of cross-axis coupling in micromechanical springs and at multi-dimensional curvature in the multi-layer suspended structures in the CMOS-MEMS process. Cross-axis stiffness constants are derived for basic spring topologies such as crab-leg, u-spring and serpentine springs. Techniques to reduce, and even completely eliminate, elastic cross-axis coupling are discussed. In the electrostatic domain, a methodology which combines analytical equations with numerically obtained data is developed to model CMOS-MEMS combs. Particular attention is paid in this methodology to make the resultant behavioral model energy conserving. Convergence problems found in behavioral simulations of gyroscopes lead to a detailed comparison of different Analog Hardware Description Language (AHDL) model implementation of mechanical second-order systems, such as the resonating structure in a gyroscope. AHDL model implementation guidelines for improved convergence in behavioral simulations are deduced from the comparisons.

Using the elastic and electrostatic models as the basis, analytical equations relating gyroscope non-idealities: the Zero Rate Output, acceleration and acceleration-squared sensitivity and cross-axis sensitivity to manufacturing effects are derived. The equations are compared with results of behavioral simulation. Monte Carlo simulations using the behavioral models are run in order to verify the trends predicted by the analytical equations. The analysis and simulations result in several insights into gyroscope non-idealities and design pointers to reduce them.

Acknowledgments

There are a number of individuals who have contributed to the completion of this thesis, technically as well as philosophically. First, I would like to thank my advisor Dr. Tamal Mukherjee. In the course of my graduate studies I have benefitted from his experience in a number of aspects of conducting scientific research. In addition to the numerous technical discussions we had, which have shaped this thesis considerably, his feedback on documents and presentations have vastly improved my communication skills. His patience, encouragement and dedicated mentoring of his students is, in my opinion, extraordinary, if not uncommon.

I would also like to thank Prof. Gary Fedder whose grasp of the physics of MEMS has aided my comprehension, particularly, in developing elastic models. His considerable experience in MEMS inertial sensors has also provided a better perspective on many occasions. Technical feedback from Prof. Rick Carley, Prof. Jon Cagan and Dr. Steve Bart has helped me set a broader context for this thesis, and is also greatly acknowledged. Prof. Wojciech Maly and Prof. Jim Hoburg were also gracious in sharing their expertise and helped in guiding the direction of the thesis.

Simulations and measurements done were greatly expedited due to colleagues from the MEMS laboratory at Carnegie Mellon as well as the Center for Silicon System Implementation. My latest officemates Bikram Baidya, Kai He and Qi Jing have helped me on numerous occasions. In particular, Bikram with extraction and FEM simulations, Qi Jing with several discussions on NODAS, and Kai with support for layouts and CAD. I would also like to acknowledge Hasnain Lakdawala for his collaboration with a number of measurements, modeling discussions and his general opinion on the state of the art, Xu Zhu for releasing the chips used for measurements, Hao Luo for sharing his gyroscope design for use in this thesis and Huikai Xie for discussions on gyroscope measurements. Mary Moore and Drew Danielson in the MEMS Laboratory, Roxann Martin in the Center for Silicon System Implementation and Elaine Lawrence and Lynn Philibin in the graduate office have helped in sorting out procedural and administrative issues on numerous occa-

sions. Their friendly countenance made them very approachable and things were that much smoother with them around.

Financial support, under various grants from the Defense Advanced Research Projects Agency (DARPA), the National Science Foundation (NSF) and the MARCO Center for Circuits, Software and Systems is gratefully acknowledged.

A number of different officemates at different times made courses, research and graduate student life all the more enjoyable. Additionally, friends in CMU and Pittsburgh have made the sometimes tedious and bumpy graduate school, a pleasant experience. All of them have left their positive impressions on me. Kumar Dwarakanath, Vikram Talada, Hasnain Lakdawala, Sanjay Rao, Sudha Dorairaj, Madhavi Vuppalapati, Bikram Baidya, Debanjali Dasgupta and many others have kept my spirits aloft. Jam sessions with Aneesh Koorapaty and Sundar Vedula were also immensely enjoyable and always lightened the pressure.

I would like to acknowledge the completely selfless and unrelenting support of my two sisters Shanthi and Vidya, their husbands and my nephew and niece. Their high expectations and confidence in me, have pushed me farther than I would have ever attempted. Some of the most difficult times in the last couple of years would have been unthinkable without them. My wife Vidhya, whose cheerfulness and wisdom has made some very stressful times bearable, has been a great companion in the past few months. Finally, it should be mentioned that, the sheer will power of my mother and the excellent engineering mind of my father, have not only aided and guided me in my career, but have also set me high standards to emulate. These and many of their extraordinary qualities have carried me thus far and will continue to do so.

Table of Contents

Abstract	ix
Acknowledgments	xi
Table of Contents	xiii
List of Tables	xviii
List of Figures	xx
Chapter 1. Introduction	1
1.1 Introduction.....	1
1.2 Motivation.....	2
1.3 Scope of This Thesis.....	5
1.4 Thesis Organization	6
Chapter 2. Background	7
2.1 Introduction.....	7
2.2 Micromachined Gyroscopes	7
2.2.1 Brief History.....	7
2.2.2 Common Features of Surface-Micromachined Vertical Axis Gyroscopes.....	8
2.2.3 Vertical Axis CMOS-MEMS Gyroscope	9
2.3 CMU CMOS-MEMS Process.....	10
2.4 Structured Design Methodology for MEMS.....	13
2.4.1 Modeling	15
2.4.2 Elastic Models	16
2.4.3 Electrostatic Modeling.....	19
2.5 Summary	20
Chapter 3. Elastic Modeling	21
3.1 Introduction.....	21
3.2 Stiffness Matrices.....	23
3.3 Modeling.....	24
3.4 Model Verification	31
3.5 Accelerometer Simulation	36
3.6 Symmetric springs	37

3.7	In-plane to Out-of-plane Elastic Cross-axis coupling.....	43
3.7.1	Rotation of Principal Axes in Multi-layer Beam.....	44
3.8	Geometrical Interpretation of Cross-axis Coupling.....	50
3.9	Manufacturing Variations and Elastic Cross-axis Coupling.....	53
3.10	Curl Modeling.....	58
3.10.1	Extension of Multimorph Analysis.....	59
3.10.2	FEA Verification.....	63
3.10.3	Measurements.....	65
3.11	Summary.....	67
Chapter 4.	Reduced Order Models: From Beams to Springs.....	69
4.1	Introduction.....	69
4.2	Spring Stiffness Computation.....	70
4.2.1	Background.....	70
4.2.2	Stiffness Computation Procedure.....	71
4.3	Translation from Circuit-level to Functional Schematic.....	74
4.3.1	Nested gyroscope.....	74
4.3.2	Z-axis accelerometer.....	76
4.4	Verification.....	77
4.4.1	Verification of Spring Stiffness Computation.....	77
4.5	Comparison of Circuit-level Schematic with Functional Schematic Simulation.....	79
4.5.1	Example 1: Nested-Gyroscope Design Space Exploration.....	79
4.5.2	Example 2: Resonance Frequency Analysis of Z-axis Accelerometer.....	80
4.6	Inertial and Viscous Effects and Extension to Arbitrary Spring Topologies . 81	
4.6.1	Model Formulation.....	81
4.6.2	Model-order Reduction.....	84
4.6.3	General Reduced-order Modeling.....	85
4.7	Summary.....	86
Chapter 5.	Electrostatic Modeling of CMOS-MEMS Comb.....	87
5.1	Introduction.....	87
5.2	Background.....	89
5.3	Modeling Goals and Approach.....	91

5.4	Analytical Model for Movement in Gap Direction	91
5.5	Design of Experiments and Simulation	96
5.5.1	Comb Parameterization and Variable Selection	96
5.5.2	Variable Screening.....	97
5.5.3	Choice of Variable Ranges	97
5.5.4	Data Collection.....	99
5.6	Modeling Methodology	101
5.6.1	Capacitance Modeling.....	101
5.6.2	Combined Capacitance-Force Modeling.....	103
5.7	Differential Comb Modeling.....	104
5.8	BEA Summary	107
5.9	Results.....	107
5.9.1	Model Generation by Curve Fitting	107
5.9.2	Behavioral Model Implementation.....	109
5.10	Experimental Verification	111
5.11	Summary.....	112

Chapter 6. Convergence and Speed Issues in MEMS Behavioral

Simulation	114	
6.1	Introduction.....	114
6.2	Background.....	115
6.2.1	Numerical Integration.....	116
6.2.2	Time Discretization of Components.....	117
6.3	Model Formulation	119
6.3.1	Multi-domain Simulation	120
6.3.2	Implementation of Time-derivatives	121
6.3.3	Displacement as across variable.....	122
6.3.4	Velocity as across variable.....	124
6.4	Simulation Results	125
6.5	Model Implementation Example: Squeeze-film Damping	127
6.6	Summary.....	129

Chapter 7. Analysis and Simulation of a CMOS-MEMS Gyroscope 131

7.1	Introduction.....	131
7.2	Gyroscope Description and Circuit-level Representation	133

7.2.1	Gyroscope Description	133
7.2.2	Gyroscope Parameters	136
7.2.3	Notation	137
7.2.4	CMOS-MEMS Gyroscope Design Parameters.....	139
7.3	Gyroscope Sensitivity	141
7.4	Effect of Asymmetrical Drive.....	148
7.5	Zero Rate Output (ZRO).....	152
7.5.1	Beam Width Variation.....	153
7.5.2	Comb gap variation.....	159
7.5.3	Mask Misalignment	165
7.5.4	ZRO Summary	168
7.6	Acceleration Sensitivity	168
7.6.1	Beam Width Variation.....	170
7.6.2	Comb Gap Variation	171
7.6.3	Mask Misalignment	173
7.6.4	Summary of Acceleration Sensitivity	174
7.7	Cross-axis Sensitivity	174
7.7.1	Rotation About Sense Axis.....	175
7.7.2	Rotation About Drive Axis ()	177
7.7.3	Summary of Cross-axis Sensitivity.....	178
7.8	Simulation Results	178
7.8.1	Mismatch Simulation Results	179
7.8.2	Monte-Carlo Simulations.....	181
7.9	Summary	185
Chapter 8.	Summary and Future Work	186
8.1	Thesis Summary and Contributions.....	186
8.1.1	Modeling.....	186
8.1.2	Simulation.....	187
8.1.3	Gyroscope Design.....	187
8.2	Future Directions of Work	188
	References	190
	Appendix A1 Mathematica program to derive Crab-leg stiffness matrix ..	199
	Appendix A2 Equations for out-of-plane off-diagonal stiffness constants	

Appendix A3 Mesh Refinement Steps	206
Appendix A4 Comb Model Listing	213
Appendix A5 Comparison of Rotational and Translational Modal Frequencies	238
Appendix A6 Beam Widths and Comb Gaps for Monte-Carlo Analysis ..	240
Appendix A7 OCEAN Scripts and Sample Netlist for Gyroscope Monte-Carlo Simulations	242
A7.1 OCEAN Script	242
A7.2 Gyroscope 2D netlist	244

List of Tables

3.1	Comparison of FEA and analytical stiffness (in-plane) values for the crab-leg spring.....	32
3.2	Comparison of FEA and analytical stiffness (in-plane) values for the u-spring.....	33
3.3	Comparison of FEA and analytical stiffness (in-plane) values for the serpentine spring.....	33
3.4	Comparison of k_{yz} from NODAS and FEA.....	50
3.5	Vertical deflection (in μm) of tip in 100 μm : Macromodel vs. FEA	64
3.6	Vertical deflection (in μm) for simplified accelerometer	65
5.1	Comparison of beam and comb models	89
5.2	Mapping of points from Z plane to W plane.....	94
5.3	Experimental plan set 1 for simple comb BEA.....	97
5.4	Experimental plan set 2 for simple comb BEA.....	98
5.5	Experimental plan to obtain F_x values for simple comb using BEA	101
5.6	Experimental plan set 1 for differential comb BEA.....	105
5.7	Experimental plan set 2 for differential comb BEA.....	105
5.8	Points to be omitted due to comb finger crashing.....	106
5.9	Summary of BEA runs for simple comb.....	107
6.1	Typical ranges for various physical domains in MEMS	120
6.2	Comparison of five implementations	125
7.1	Symbols used.....	138
7.2	Geometrical parameters of the CMOS-MEMS gyroscope	139
7.3	Functional parameters of the CMOS-MEMS gyroscope	140
7.4	ZRO resulting from linear gradients in beam width	153
7.5	Comparison between analytical calculations and NODAS simulations of ZRO...	179
7.6	Comparison between analytical calculations and NODAS simulations of acceleration sensitivity	180

7.7	Comparison between analytical calculations and NODAS simulations of cross-axis sensitivity	180
A2.1	Comparison of FEA and analytical stiffness (out-of-plane) values for the crab-leg spring.....	204
A2.2	Comparison of FEA and analytical stiffness (out-of-plane) values for the U spring... 204	
A2.3	Comparison of FEA and analytical stiffness (out-of-plane) values for the serpentine spring.....	205
A6.1	Fractional variation of beam widths and gaps used for Monte-Carlo simulations.	240

List of Figures

1.1.	Working principle of a microgyroscope with a sensing accelerometer nested inside a vibrating frame.	2
1.2.	Cross-section of microstructures in a CMOS-MEMS process.....	4
2.1.	Topology comparison of (a) single-layer surface micromachined gyroscopes (e.g., Clark et al. [19]) and (b) CMOS-MEMS nested gyroscope [10]. The dark shaded combs are the drive combs. Shading in the sense combs indicates different potentials.	8
2.2.	(a) SEM of the vertical axis CMOS-MEMS nested-gyroscope [10]. (b) Functionally equivalent structure showing the inner accelerometer, outer rigid frame, inner and outer springs and drive and sense combs.	10
2.3.	Abbreviated process flow for post-CMOS micromachining developed at CMU [7][31][35] (a) CMOS wafer cross-section with circuits and interconnects (soon to be microstructures) (b) Oxide removal step (c) Microstructure release by Silicon removal.....	11
2.4.	(a) Lateral curling seen in beams with deliberately misaligned metal layout (b) Cross-section of the beam. It is seen that the METAL3 is not aligned with the METAL2 and METAL1. (Pictures courtesy Xu Zhu and Hasnain Lakdawala)	12
2.5.	(a) Geometrical offset in a differential comb-drive used in a CMOS accelerometer. One of the gaps is smaller than the other one (b) Laterally curled springs in an accelerometer (Pictures courtesy Vishal Gupta)	13
2.6.	MEMS design hierarchy.....	14
2.7.	The direction along the length (x) and the direction of deflection of a beam (y).....	16
2.8.	(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.....	18
3.1.	Outer frame of a gyroscope driven by a sinusoidal voltage source and a DC source. Motion of point C is shown on the right. (a) With ideal springs oscillations are only along the y direction. (b) However, with non-ideal mismatched springs small amount of motion couples to the x direction.....	21
3.2.	Elements of the stiffness matrix and the in-plane and out-of-plane sub-matrices and . This symmetric matrix has 21 distinct terms. If the shaded elements are zero, the number of distinct non-zero terms reduces to 12.	23

3.3. Design variables for crab-leg-spring, U-spring and serpentine spring with proof-mass. The external forces and moments are applied at C, the centroid of the plate, with only one spring in the analysis so that all the cross-axis terms can be clearly observed.	25
3.4. Forces and moments applied at the centroid of a proof-mass attached to the free end of a crab-leg. Boundary conditions are applied as equality constraints on the three displacements.	26
3.5. Trends in the variation of in-plane spring constants for the U-spring for varying beam lengths (L_{b1}). The design variables are set to: $w=2.0 \mu\text{m}$, $L_t=10.0 \mu\text{m}$, $L_{b2}=200.0 \mu\text{m}$	30
3.6. Trends in the variation of in-plane spring constants for the serpentine-spring for varying beam lengths (a). The design variables are set to: $w=2.0 \mu\text{m}$, $L_b=20.0 \mu\text{m}$, $n=4$	31
3.7. Comparison of analytical model and FEA for crab-leg-spring k_{xy} for varying crab-leg thigh lengths (L_t). The design variables are set to: $w=2.0 \mu\text{m}$, $L_s=50.0 \mu\text{m}$	35
3.8. Comparison of analytical model and FEA for u-spring k_{xy} for varying U-spring beam lengths (L_{b1}). The design variables are set to: $w=2.0 \mu\text{m}$, $L_t=10.0 \mu\text{m}$, $L_{b2}=L_{b1}-30.0 \mu\text{m}$	35
3.9. Comparison of analytical model and FEA for serpentine-spring k_{xy} for varying serpentine-spring beam lengths (a). The design variables are set to: $w=2.0 \mu\text{m}$, $L_b=20.0 \mu\text{m}$, $n=4$	36
3.10. NODAS simulation of cross-axis sensitivity in y-accelerometer. The structure of the accelerometer with four serpentine springs is shown on the side.	37
3.11. Axes of symmetry for the u-spring and serpentine spring	38
3.12. Example of a symmetric spring with the axes of symmetry along the y direction. The load point C, the anchor point A and the end points of beam number 9 are also shown.	40
3.13. Rotation of beam principal axes due to asymmetrical cross-section: asymmetrical side-walls in the single-layer case and misaligned metal layers in the multi-layer case	44
3.14. (a) Side view of a n-layer beam of length L (b) Cross-section of the beam with dots representing the axial forces acting out of plane.....	45

3.15. Cross-section of beams used for comparison of macromodel with FEA. The k_{yz} matches to 2% for 2 sets (50 μm X 3.0 μm and 100 μm X 2.1 μm) of 3 beams each.	49
3.16. In-plane rotation and displacement of the principal axes of elasticity	51
3.17. Signs of off-diagonal elements in the in-plane stiffness matrix	53
3.18. Linear variation in beam-widths across a wafer, mapped onto the springs of a single device with a plate suspended by four springs	54
3.19. (a) Model of a CMOS cantilever beam composed of metal, dielectric and polysilicon layers (b) Cross-section of an asymmetric multi-layer beam with dots representing the axial forces acting out of plane. Since the forces are asymmetrically located there is a resultant lateral bending moment in addition to the vertical bending moment	59
3.20. Norton equivalent of a beam macromodel with thermally induced lumped force and moment sources and an embedded piezoresistor. The beam has three translational pins, three rotational pins and one electrical pin at each port.	62
3.21. (a) Cross-section of beams. Metal3 is 2.1 μm wide and Metal2 and Metal1 layers are 1.8 μm wide and are offset by -0.15 μm from the center of the beam (b) Comparison of behavioral curl model with FEA for beam of length 100 μm . Difference is less than 3% for all temperatures.....	64
3.22. Temperature-induced curling of a simplified accelerometer structure obtained from 3D FEA. The vertical deflections at points (a) and (b) are compared with results from behavioral simulation using the macromodel in Table 3.6.	65
3.23. (a) SEM of the test structure used to characterize beam curling with temperature. It consists of alternating misaligned and symmetric beams (b) Cross-section of the measured beams (misaligned)	66
3.24. Interferometric images of the out-of-plane curl of the test structure at 24°C and 56°C. One fringe length corresponds to 245 nm displacement in the vertical direction.....	67
3.25. Comparison of relative tip deflection from measurements and macromodel showing a match to within 15%. 100 nm and 20 nm error bars are shown for the z and x deflections respectively.	67
4.1. (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.	75
4.2. Functional model generated from the circuit-level schematic of the nested-resonator system shown in Figure 4.1(b)	75
4.3. (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.	76
4.4. Functional schematic generated from the circuit-level schematic of the accelerometer shown in Figure 4.3(b).....	77
4.5. 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.....	78
4.6. Comparison of spring stiffness computation for k_{xx} with FEA.....	78
4.7. Comparison of spring stiffness computation for k_{yy} with FEA.....	78
4.8. Comparison of spring stiffness computation for $k_{\phi_z\phi_z}$ with FEA	78
4.9. Difference in the resonant frequency extracted from the schematic simulations and behavioral simulations (a) drive-mode (b) sense-mode	79
5.1. (a) Top view of a simple comb with three comb fingers (the lesser of the two numbers is taken as the number of fingers in the comb) (b) Cross-section of a comb finger in the CMOS-MEMS process.....	87
5.2. Top view of a differential comb along with the equivalent capacitive divider schematic.....	88
5.3. (a) Cross-section of a comb showing two fixed fingers (F) and one movable finger (M) along with odd and even symmetry planes used in derivation of conformal-mapping based analytical models. (b) Movement in the gap direction breaks the symmetry.....	89
5.4. Simplification of laterally displaced comb cross-section using symmetry (a) Comb section showing 3 fixed fingers and 2 movable fingers displaced in the x direction from the nominal symmetrical position (shown with dotted lines). (b) Simplification by introduction of odd symmetry planes (c) Equivalent configuration with the fixed comb-fingers replaced by odd-symmetry plane placed midway between the rotor and stator fingers	92

5.5. (a) Conformal mapping for a single conductor placed asymmetrically between two ground planes	93
5.6. Comparison of analytical model adapted from [75] and FEA for x displacement of the comb-finger cross-section shown in Figure 5.4.	95
5.7. Boundary-element mesh for a 10 finger vertically curled lateral comb with all CMOS layers.	100
5.8. Histogram of error in % between model and numerical data for 4566 capacitance points. The error is almost within $\pm 3\%$	108
5.9. Histogram of error in % between model and F_x from numerical derivative of capacitance w.r.t x	108
5.10. Histogram of error in % between model and F_y obtained from numerical derivative of capacitance w.r.t. y	108
5.11. Histogram of error in % between model and F_z obtained from numerical derivative of capacitance w.r.t. z	108
5.12. Histogram of capacitance error in % between model and numerical data for the differential comb	109
5.13. Functional diagram of the CMOS-MEMS gyroscope used in simulation [10].....	110
5.14. Interferometry image showing the vertical offset in drive comb	110
5.15. Drive amplitude variation of a gyroscope with change in vertical overlap between the movable and fixed portions of the comb actuator	110
5.16. (a) SEM of capacitance test structure with in-built heaters. The curling can be changed by changing the current passing through the polysilicon wires which pass through the outer frame of the structure. Interferometry images of a quarter of the structure at (b) room temperature and (c) heated are also shown.	111
5.17. Capacitance change measurement schematic shown with heating resistors for each structure. C_p is the parasitic capacitor whose value is obtained using layout extraction.....	112
5.18. Comparison of measured and predicted capacitance change. The two sets of measured data correspond to the voltage applied to each of the two resistors, with 0.1 V applied to the other resistor.	112
6.1. Computation of state variable using Backward Euler integration rule. The areas of the rectangles obtained by integration are shown.	117

6.2.	Circuit interpretation of Backward Euler integration rule for (a) Parallel RLC network (b) Mechanical spring-mass-damper system modeled by use of two additional states to hold v_n and a_n	118
6.3.	In-plane stiffness matrix for a beam [61]. Beam with length = 100 μm , width = 2 μm , thickness = 2 μm and Young's Modulus $E = 165 \text{ GPa}$. A is the cross-section area and I is the moment of inertia. (a) The large span of the diagonal elements of the stiffness matrix is evident. (b) The stiffness matrix after scaling the rotational discipline by has much smaller condition number.	121
6.4.	Squeeze-film damping modeled by equivalent resistors and inductors. Only the first two R-L branches are shown in the figure. More accurate models need more number of branches.	128
6.5.	VerilogA code and equivalent circuit for first implementation of the squeeze-film damping model. Only two of the RL branches are shown. The actual circuit interpretation by the simulator is not exactly known but is probably more complex because it is observed that additional states are implicitly introduced during simulation.	128
6.6.	VerilogA code and equivalent circuit for second implementation of the squeeze-film damping model. The controlled sources inside the damping model are more locally distributed compared to the first implementation. This implementation shows better convergence properties.	129
7.1.	(a) Nested gyroscope design showing the drive and sense combs, the outer and inner springs, the input axis, the direction of driven vibrations and the direction of Coriolis-force induced (sense) vibrations (b) Sense capacitance bridge for movement of inner proof-mass in positive x axis	131
7.2.	(a) SEM of the nested-gyroscope (b) Corresponding NODAS schematic obtained through layout extraction.	134
7.3.	(a) Output spectrum of an ideal gyroscope for an input sinusoidal rotation rate (b) sense schematic showing demodulation of gyroscope capacitance bridge output to yield voltage proportional to input rate. The angle needs to be adjusted to maximize sensitivity and minimize offsets	135
7.4.	Classification of sources of microgyroscope non-idealities.....	137
7.5.	Anti-phase voltages applied to drive the gyroscope into oscillations.	142
7.6.	Nested resonator system and dynamical equations (a) when a force F_{co} is applied to the outer frame and (b) when a force F_{ci} is applied to the inner mass.	145
7.7.	Asymmetrical drive: Actuation voltage applied to only the top linear comb	149

7.8. Example case for beam width variation and equation for coupling, $w_1 = w_2 = w_3 \neq w_4$	154
7.9. Example case for beam width variation, $w_1 = w_2 = w_3 \neq w_4$	157
7.10. Mismatch in the gaps in the drive combs on the top and bottom.....	159
7.11. Mismatch in the gaps in the sense combs on the two sides.....	163
7.12. Cross-section of one set of fingers of a differential sense comb (a) without lateral offset, vertical motion leads to common-mode capacitance change; (b) with lateral offset, vertical motion leads to common-mode and differential capacitance change. .	165
7.13. Spectrum of output voltage of a non-ideal gyroscope when subjected to an external acceleration.....	169
7.14. ZRO Histograms for widths and gaps = 1.8 μm and 2.0 μm from Monte-Carlo simulation.	182
7.15. Acceleration sensitivity histograms for widths and gaps = 1.8 μm and 2.0 μm from Monte-Carlo simulation.	182
7.16. Acceleration-squared sensitivity histograms for widths and gaps = 1.8 μm and 2.0 μm from Monte-Carlo simulation.	183
7.17. Trade-off between gyroscope sensitivity and acceleration sensitivity with varying gap	184
A3.1. Regions defined in the boundary element model for mesh refinement. REFINE0 is the most finely meshed region, followed by REFINE1 and REFINE3 and finally REFINE2 and REFINE4 are the most coarsely meshed regions.	206
A5.1. Rectangular plate suspended by four springs	238

Chapter 1. Introduction

1.1 Introduction

The field of Microelectromechanical Systems (MEMS) has, over the past 20 years, emerged as a technology that promises to have significant impact on everyday living in the near future. MEMS provide inexpensive means to sense and, in a limited way, control physical, chemical and biological interactions with nature. They add a new dimension to the information revolution of the latter half of the twentieth century, by enabling ubiquitous access to sensor data previously limited to industrial, military, and medical applications. MEMS seek to achieve this vision through a variety of manufacturing techniques common among which are surface micromachining, bulk micromachining and LIGA [1]. These integrated circuit (IC)-like techniques are capable of producing micrometer-scale features. However, they lack the precision (i.e., relative accuracy) of traditional mechanical fabrication practices. Being integrated circuit compatible, they derive their power by leveraging well-understood and characterized signal processing capabilities of integrated circuits. As a result, a wide spectrum (literally as well) of applications have been made possible such as inertial sensors, pressure and acoustic transducers, high frequency radios, optical communications, lab-on-a-chip for chemical and biological analysis.

The contributions of this thesis are primarily relevant for micromachined inertial sensors. Accelerometers and gyroscopes are two important members of the inertial sensor family. Accelerometers sense the external acceleration in which they are placed, while gyroscopes measure the rate of rotation or the angular velocity of the object to which they are attached. Multi-axial accelerometers and gyroscopes can be combined to build an Inertial Measurement Unit (IMU), also called an Inertial Navigation System (INS). Traditionally, high precision IMUs have been an indispensable part of ships, aeroplanes, satellites, space shuttles and the like. Surface micromachined inertial sensors, which can be batch-fabricated with low cost have a small sensing proof-mass (\sim micrograms) and consequently lower resolution compared with macro-scale accelerometers or optical gyroscopes [2]. The availability of low cost inertial sensors has opened up a wide range of new applications which do not require the high precision that IMUs demand. The current market for

inertial sensors in automobiles is estimated to be about a billion dollars per year [3]. Air-bag-deployment in automobiles is a well known example of a commercially successful low cost, low resolution application. Surface-micromachined gyroscopes, have applications in dynamic stability control and rollover detection in automobiles, computer mice, pointers, video camera stabilization and a number of robotics and military applications [2][4]. Conventional rotating-wheel gyros and high-precision fiber-optic and ring laser gyros are too expensive and too large to be adopted into the market for micro gyroscopes [2]. While potential markets for inexpensive gyroscopes exist, technical challenges have been impeding the rapid commercial deployment of gyroscopes. In the next section some of the fundamental problems that have been encountered in manufacturing robust micro-machined gyroscopes are examined.

1.2 Motivation

Microgyroscopes are mainly attractive because of their small size ($\sim 1 \text{ mm} \times 1 \text{ mm}$ including sensing circuits) and low cost. Most microgyroscopes consist of a vibrating proof-mass which is driven into oscillation by electrostatic or other means. When placed in a rotational field, the vibrating proof-mass experiences an apparent force called the Coriolis force, which is proportional to the cross-product of the angular velocity of the rotational field and the translational velocity of the oscillating proof-mass (Figure 1.1). The Coriolis force is orthogonal to the direction of the driven oscillation. The displacement induced by the Coriolis force is picked up by a sense accelerometer, which can either

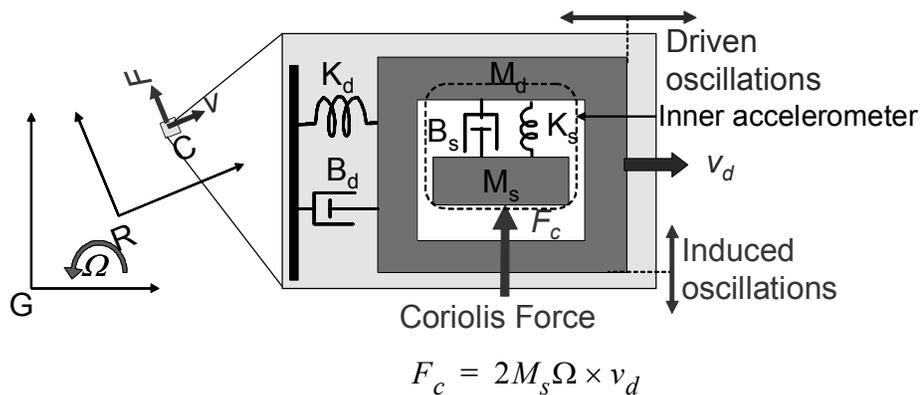


FIGURE 1.1. Working principle of a microgyroscope with a sensing accelerometer nested inside a vibrating frame.

utilize the vibrating proof-mass or have a separate sense proof-mass. Figure 1.1 shows the microgyroscope chip C which is attached to a rotating frame R. The global inertial reference frame G is also shown. The velocity v_d of driven oscillation and Coriolis force F_c are shown in mutually orthogonal directions. The zoomed in view of the gyroscope schematically shows an inner accelerometer nested inside an outer resonator. The outer accelerometer is driven into oscillation. Orthogonal induced oscillations between the two are picked up by a capacitive sensing circuit. In order to identify important design issues, a high-level analysis of typical magnitudes of various microgyroscope quantities is presented below.

In the following analysis typical numbers for microgyroscopes are used in order to bring out the relative magnitudes of displacements and velocities in the driven (oscillation) direction and the induced (Coriolis force) direction. Typical value of the sense mass is about $M_s = 1\mu g$. The angular velocities that can be sensed are of the order of $\Omega = 1^\circ/s$. The oscillations are usually about 10 kHz with an amplitude of about 5 μm . Therefore, the peak oscillation velocity is about $v_d = 0.31 m/s$. The Coriolis force is then given by $F_c = 2M_s\Omega \times v_d = 10 pN$. Assuming a spring stiffness for the sense accelerometer of $1(N/m)$, the sense displacement is about 10 pm. In any real microgyroscope, some part of the driven oscillation couples onto the sense accelerometer, through electrostatic, inertial, viscous and elastic modes. Comparing the magnitudes of the driven oscillation and the displacement produced by the sense accelerometer, it is seen that undesired coupling from the driven oscillation to the sense oscillation should be as small as 2 ppm. While this may be a difficult number to achieve in any low cost system, it is almost impossible to realize such precise dimension-control in IC-based processes which typically control relative fabrication tolerances to only about 1% or 10000 ppm [5]. Furthermore, in capacitive sensors, a displacement of few picometers typically results in a capacitance change of a few zepto farads (10^{-21}). The total sense capacitance and parasitic capacitances are usually of the order of ten to hundred femto farads, leading to a relative capacitance change of 0.1 ppm. Therefore, extremely low noise front ends are required to sense such small relative capacitance changes. Another fundamental issue which limits the resolution of both microaccelerometers and microgyroscopes is mechani-

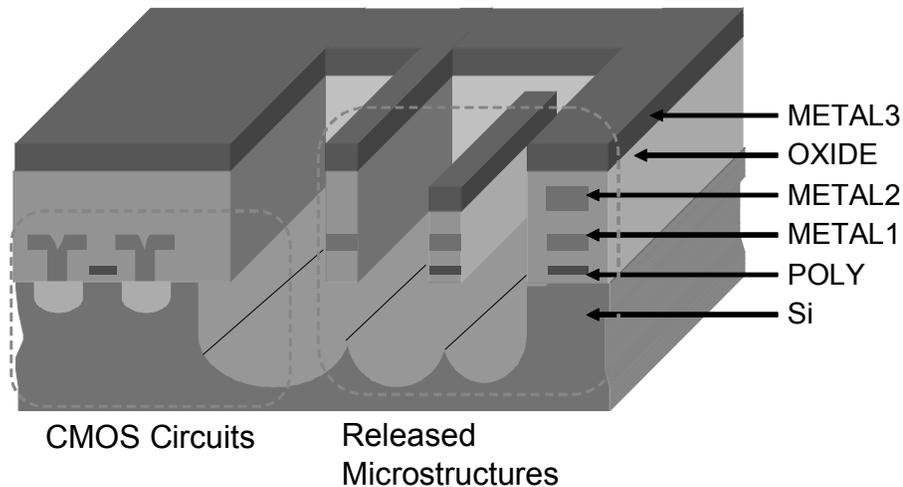


FIGURE 1.2. Cross-section of microstructures in a CMOS-MEMS process

cal thermal noise. Since IC processes and surface micromachining are both basically thin film processes, the resulting proof-masses tend to have a large surface-area to volume ratio. As a result, viscous damping forces are more significant at the micro-scale than at the macro-scale. Therefore, the Brownian motion resulting from viscous loss mechanisms sets a lower limit on the smallest deterministic motion that can be sensed [6].

The solutions to the above fundamental challenges need a multi-pronged approach. The CMOS-MEMS process developed at Carnegie Mellon provides partial answers to some of the challenges [7] (Figure 1.2). Tight integration of MEMS and sensing circuits leads to minimized parasitic capacitances. Large gaps between the MEMS structure and the substrate ($\sim 30 \mu\text{m}$) lead to reduced Couette damping on the underside of the structure. In addition, the CMOS-MEMS process has several other advantages including full compatibility with a standard CMOS process, 0 additional masks for MEMS processing, high aspect ratio MEMS structure and multi-conductor stacks which facilitate complex routing. However, the CMOS-MEMS process also has a few inherent limitations such as vertical curling due to the multi-layer structures, inadequate control of the beam cross-sections and lack of control over mechanical properties of the microstructure.

1.3 Scope of This Thesis

The primary goal of this thesis is to understand the effect of elastic and electrostatic coupling on a CMOS-MEMS microgyroscope. Specifically, the microgyro parameters under consideration are [8][9]:

- **Zero Rate Output (Input offset or Bias):** Input rate required to drive the output voltage to zero.
- **Acceleration and Acceleration-squared Sensitivity:** A linear acceleration applied to the gyroscope may result in an output indistinguishable from that produced by an input rotation. Typically, gyroscopes show a linear as well as quadratic dependence on acceleration.
- **Cross-axis Sensitivity:** Output produced by an angular rotation about an axis orthogonal to the input axis of the gyroscope.

Little or no attention has been paid to the above parameters in public literature and therefore, they are the primary focus of this thesis. Gyro resolution, sensitivity and non-linearity have been analyzed extensively in public literature and therefore, are not covered as part of this thesis.

Good design practice dictates that designers have estimates of expected non-idealities before resorting to simulation tools for more detailed results. The primary goal of this thesis is to provide gyroscope designers with techniques for hand analysis of non-idealities. Behavioral modeling and simulation is used throughout this thesis as a tool to verify hand analysis as well as to provide quantitative data. Development of behavioral models and solution techniques for associated simulation problems comprise a significant portion of this thesis. In order to obtain a quantitative understanding of elastic and electrostatic coupling in a CMOS-MEMS gyroscope, the following issues are discussed en route to the gyroscope:

- General theory of in-plane elastic cross-axis coupling
- Out-of-plane elastic coupling in CMOS-MEMS beams
- Lateral and vertical curling of CMOS-MEMS beams with arbitrary boundary conditions

- Model-order reduction for springs
- 3D modeling of electrostatic combs in the CMOS-MEMS process
- Convergence problems in MEMS behavioral simulations

In the next section, the organization of the topics listed above is described.

1.4 Thesis Organization

The thesis is organized as follows. Chapter 2 briefly surveys the development of micromachined gyroscopes, introduces the CMOS-MEMS process and presents an introduction to the behavioral simulation framework which is used extensively and also contributed to in this thesis. Chapter 3 addresses elastic cross-coupling and thermoelastic analysis for a restricted class of spring suspensions. Chapter 4 discusses reduced-order modeling primarily of suspension beams elastic properties, but also suggests possible extension to include viscous and inertial effects. Chapter 5 describes the electrostatic modeling approach for CMOS-MEMS combs. Convergence problems in MEMS behavioral simulation and guidelines for minimizing them are detailed in Chapter 6. Analysis and simulation of non-idealities in the CMOS-MEMS gyroscope are presented in Chapter 7. Finally, the contributions of the thesis are summarized in Chapter 8 and future directions of work are suggested.

Chapter 2. Background

2.1 Introduction

Much of the work in this thesis falls in the intersection of three complementary research areas: micromachined gyroscopes, CMOS-MEMS and behavioral modeling and simulation. Each of these areas is briefly reviewed in this chapter in order to place the remainder of the thesis in perspective. In the initial section of this chapter, the development of surface-micromachined gyroscopes, and, specifically, vertical-axis gyroscopes, over the last decade is reviewed. Following this, a qualitative comparison between a representative single-layer vertical-axis microgyroscope and a CMOS-MEMS vertical axis gyroscope is made [10]. The CMOS-MEMS process and relevant non-ideal manufacturing effects are then described. In the subsequent part of this chapter, the behavioral simulation framework called Nodal Simulation of Sensors and Actuators (NODAS) [11][12], which has been developed at Carnegie Mellon will be described. NODAS is used for microgyroscope simulation in this thesis. Additionally, models developed as part of this thesis have been incorporated into NODAS.

2.2 Micromachined Gyroscopes

2.2.1 Brief History

Microgyroscopes can be classified by a number of different criteria: by the manufacturing process into surface and bulk micromachined, in terms of the sensing axis as vertical axis and lateral axis or in terms of the intended application range as rate grade, tactical grade and inertial grade [2]. Most of the surface-micromachined gyroscopes reported so far fall in the rate-grade category. The first microgyro reported in 1991 was a surface-micromachined lateral axis gyroscope [13] followed up in [14]. Alternate microgyros built using alternate sensing techniques: piezoresistive [15], tunneling-based [16] and optical sensors [17] have also been reported. The first surface-micromachined vertical (Z) axis gyroscope was made at the University of Michigan in 1994 [18]. This gyro used a vibrating ring suspended by radial springs to sense the Coriolis force. Most of the vertical-axis microgyros developed since then are single-layer structures and use translational drive and

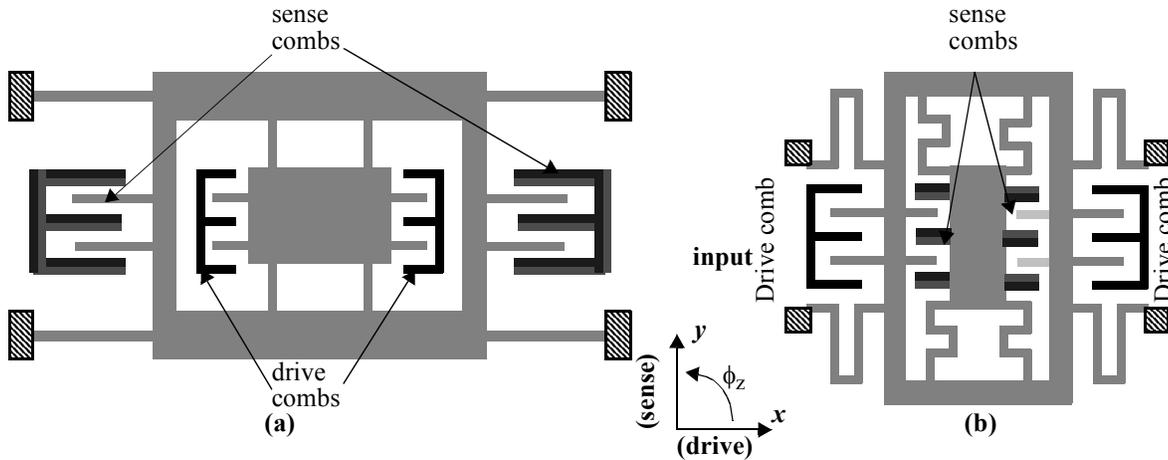


FIGURE 2.1. Topology comparison of (a) single-layer surface micromachined gyroscopes (e.g., Clark et al. [19]) and (b) CMOS-MEMS nested gyroscope [10]. The dark shaded combs are the drive combs. Shading in the sense combs indicates different potentials.

sense modes, in the plane of the structure [19][20][21][22][23][24][25][26][27][28][29]. Lateral [30][31] and vertical axis gyroscopes [10][32] have been built and successfully tested in the multi-layer CMOS-MEMS process. In the next sub-section common features of many single-layer vertical axis microgyroscopes are highlighted and compared with the CMOS-MEMS vertical axis microgyro.

2.2.2 Common Features of Surface-Micromachined Vertical Axis Gyroscopes

The polysilicon microgyroscope developed by Clark et al. [19] at Berkeley is representative of a number of later microgyroscope designs. As shown in Figure 2.1(a), linear combs are used to actuate the inner-mass in the x direction. The inner plate vibrates with large amplitude (few μm) in the x direction. The outer frame, along with the inner plate is, free to oscillate in the y direction. A pair of differential combs on the outside are used to pickup the Coriolis force induced vibrations in the y axis.

The outer frame is suspended by springs which are stiff in the x direction and therefore, has only a small amount of drive motion (few nm). Therefore, the Coriolis force due to the vibration of the outer frame is insignificant. The Coriolis substantially acts only on the central plate, but is transmitted to the rigid frame through the connecting beams which are stiff in the y direction. Thus, only a fraction of the total mass available is being used to sense the Coriolis force. Furthermore, it is seen that the central plate along with the rigid frame is easily displaced in the y direction due to external accelerations. This opens up a

possibility of external accelerations coupling through to the output. There are alternate suspension schemes, which completely decouple the sense and the drive modes [26][32][33][34], but use linear combs for sensing purposes. Linear combs are less sensitive than differential combs for same number of fingers.

Note that since the entire movable structure is at the same potential, the differential sense fingers have to be anchored. There are two possible locations for the sense combs; outside the movable frame as is shown in Figure 2.1(a) or inside the movable frame. The suspension design is such that the drive motion cannot be decoupled from the inner plate. Recall from Chapter 1 that the drive motion is more than 4 orders of magnitude larger than the Coriolis force induced motion. Differential sense combs are typically non-ideal after manufacturing. They can be expected to have a small sensitivity to cross-axis motions, as will be shown in Chapter 7. Placing the differential sense combs inside the movable frame will, therefore, lead to a significant sense signal due to the drive motion coupled to the sense combs. Thus, in case of the single-layer microgyro with the suspension design as shown, the only reasonable alternative is to place the differential combs outside the rigid frame.

In contrast to the single-layer microgyro described above, the CMOS-MEMS nested gyro topology [10] allows use of differential comb for sensing and, at the same time, allows for decoupling of the drive and sense modes. In the next sub-section the vertical axis CMOS-MEMS gyroscope which is used throughout this thesis for simulations is described.

2.2.3 Vertical Axis CMOS-MEMS Gyroscope

The SEM of a nested gyroscope [10] is shown in Figure 2.2(a). This gyroscope is fabricated in the CMOS-MEMS process [7]. It consists of an inner accelerometer nested inside an outer resonator [10] as shown in Figure 2.2(b). The outer resonator is suspended by four springs which are relatively rigid along the sensing direction (x) and compliant along the driven direction (y). The outer resonator is driven at resonance and the inner resonator is forced to move along with the outer resonator because the springs suspending the inner resonator are relatively rigid in y and compliant in x . In the presence of an angular

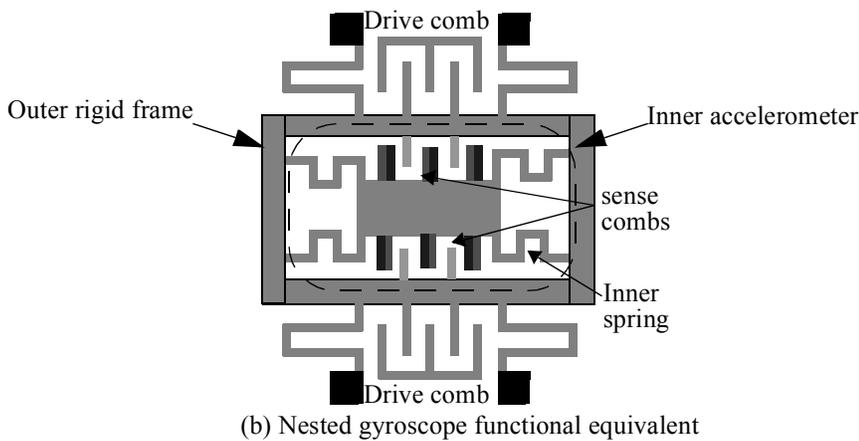
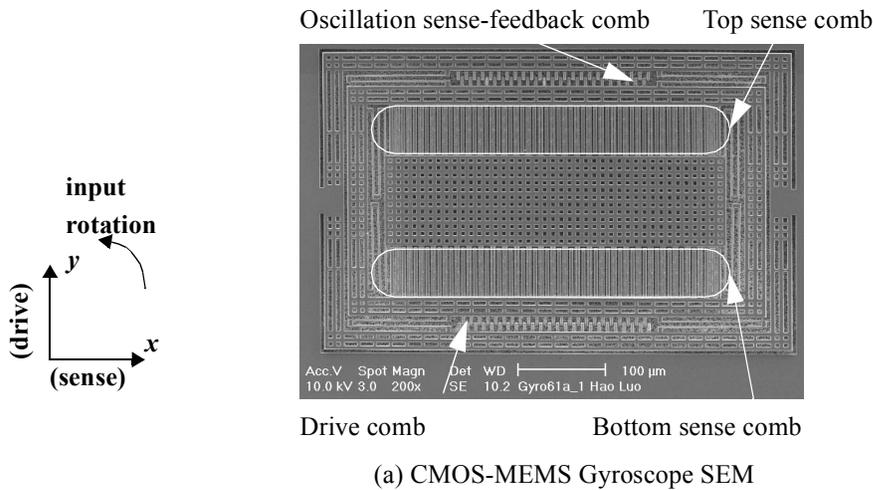


FIGURE 2.2. (a) SEM of the vertical axis CMOS-MEMS nested-gyroscope [10]. (b) Functionally equivalent structure showing the inner accelerometer, outer rigid frame, inner and outer springs and drive and sense combs.

rate Ω_z about the out-of-plane axis, both the resonators experience the Coriolis force in x , however, the inner resonator has a larger displacement. The sensing mode resonant frequency is designed to be larger than the drive frequency. The relative displacement between the two resonators is sensed capacitively using differential combs.

Thus, the CMOS-MEMS gyroscope uses springs to decouple the drive and sense modes and multiple conductors to place the differential sensing combs between the central plate and the rigid frame, both of which are movable.

2.3 CMU CMOS-MEMS Process

In the CMOS-MEMS process developed at CMU [7][31][35], released microstructures are produced by two step post processing of a standard CMOS die. First, an anisot-

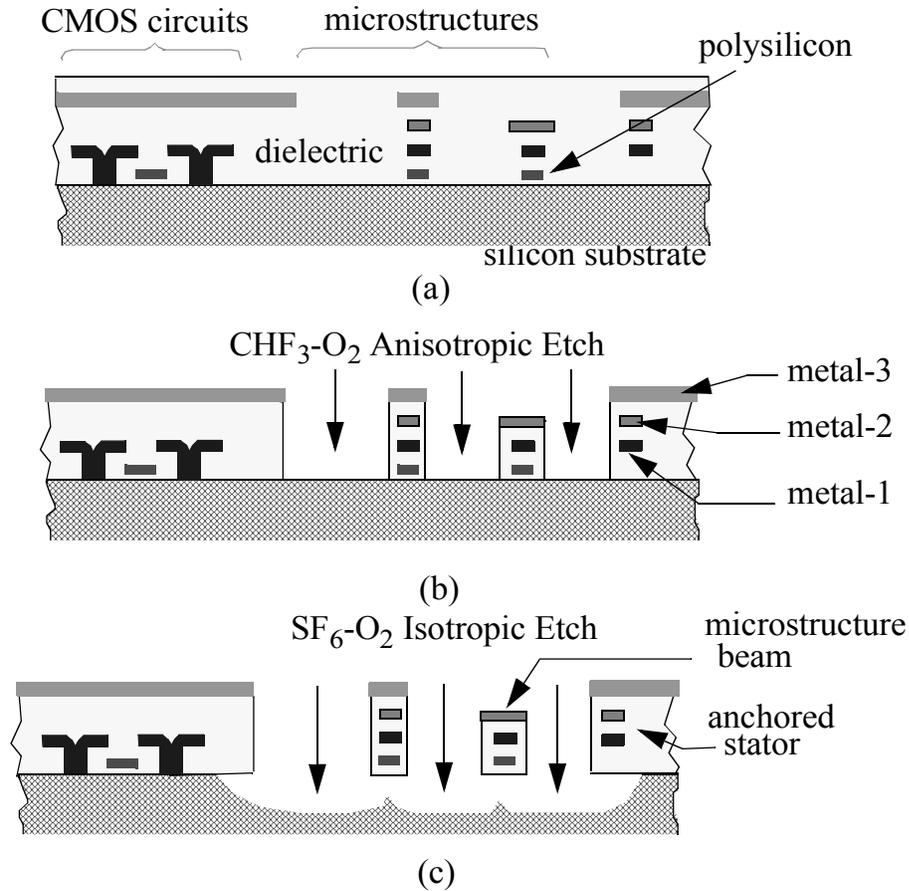


FIGURE 2.3. Abbreviated process flow for post-CMOS micromachining developed at CMU [7][31][35] (a) CMOS wafer cross-section with circuits and interconnects (soon to be microstructures) (b) Oxide removal step (c) Microstructure release by Silicon removal

tropic reactive-ion etch (RIE) of the dielectric not covered by any metal layer is used to define the sidewalls of the microstructures (Figure 2.3(b)). Following this, an isotropic etch of the Silicon substrate leads to release of the suspended microstructures (Figure 2.3(c)).

The suspended microstructures are composed of a sandwich of metal and dielectric layers. Since the materials have different thermal coefficients of expansion, the microstructures behave like thermal multi-morphs. Therefore, after processing, when the wafer temperature is reduced to room temperature, residual stresses appear which tend to curl the microstructure. In sense combs, vertical curling of the fingers leads to reduced sensitivity because of reduced overlap area. The actuation force in case of driving combs is degraded because of reduced change in capacitance with displacement. Furthermore,

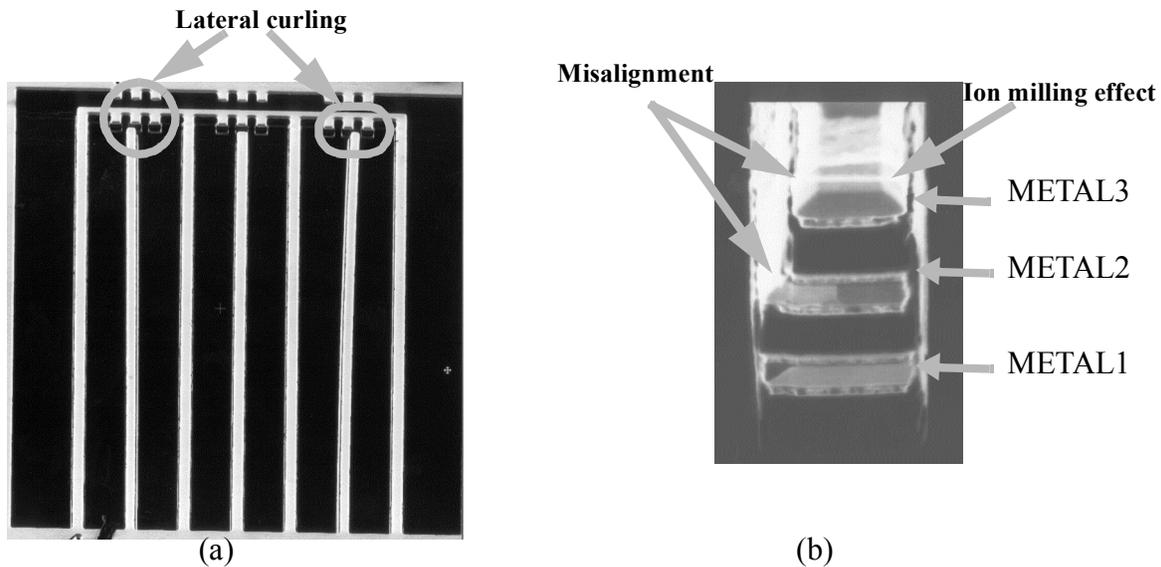


FIGURE 2.4. (a) Lateral curling seen in beams with deliberately misaligned metal layout (b) Cross-section of the beam. It is seen that the METAL3 is not aligned with the METAL2 and METAL1. (Pictures courtesy Xu Zhu and Hasnain Lakdawala)

fringe capacitance due to the edges and corners of the comb-fingers becomes significant at lower overlap areas. Curl-matching frames around the sensor have been proposed in order to reduce the mismatch of the comb-fingers [36]. However, design of these frames for the gyroscope under consideration is more complicated than the accelerometer in [36] and the resultant curl is therefore, not as well-matched. Though vertical curling is seen throughout the gyroscope it is the comb-drive which is affected significantly because it requires maximum overlap of the comb-fingers. Curling of the rest of the structure can be encapsulated into a vertical displacement offset for the comb-drive and can be modeled by considering different vertical positions of the comb-drive.

The CMOS-MEMS beams have embedded metal layers. Misalignment of the metal layer mask during processing [38] can result in the metal layers inside the beam being offset from the center of the beam leading to an asymmetrical beam cross-section. This in turn leads to elastic coupling between the in-plane and the out-of-plane modes and lateral curling of beams and comb-fingers (Figure 2.4(a), (b)). Elastic coupling can lead to an input offset in the microgyroscope due to the drive mode coupling onto the sense mode. Geometrical offsets are caused by lateral curl of the fingers or the beams in the springs (see Figure 2.5). As will be seen later, geometrical offsets give rise to input offsets and cross-axis sensitivity. Other reasons for geometrical offsets include stress gradients along

the chip substrate or along the beam width. However, currently there are no available measurements for stress gradients in the substrate or across the beam width.

Modeling approaches for the non-ideal manufacturing effects described above are presented in the following chapters.

2.4 Structured Design Methodology for MEMS

There are on-going efforts to establish a hierarchy of design levels for MEMS [11][12] similar to that existing in the digital design world. The basis for the hierarchy is decomposition of MEMS devices into MEMS atomic elements such as *plate* masses, *beam* springs, electrostatic *gaps* and *anchors* which are at a similar level as resistors, capacitors and inductors in the electronics design hierarchy. This level is referred to as the atomic level representation. An atomic level schematic representation of MEMS 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 (functional) level, all the springs which connect two rigid elements (for instance, a plate and an anchor) can be lumped together into a single functional *spring* element. The building blocks at this level are “functional” elements such as *mass*, *spring*, *damper*, *electrostatic sensor*, *electrostatic actuator* and *differential sensor*. Each of the functional elements exhibits only one kind of functionality as opposed to the circuit-level atomic elements which incorporate multi-domain physics. At the functional level, the different performance contributions are segregated, requiring the parasitic-physics effects (i.e., mass of beams, damping forces on

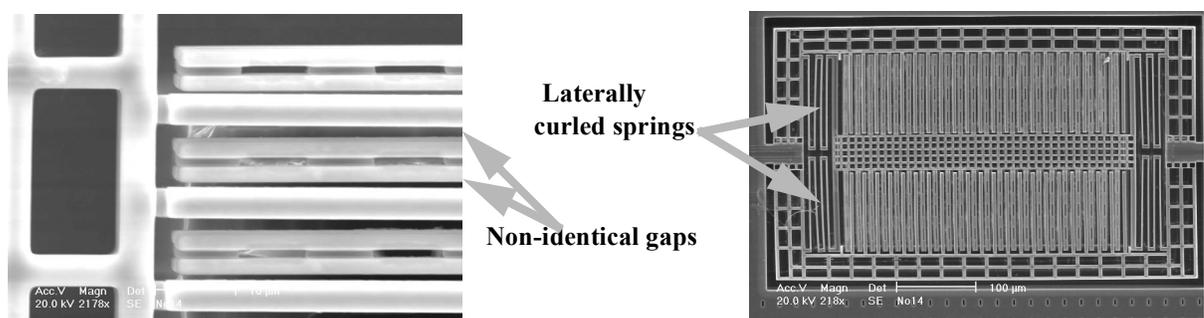


FIGURE 2.5. (a) Geometrical offset in a differential comb-drive used in a CMOS accelerometer. One of the gaps is smaller than the other one (b) Laterally curled springs in an accelerometer (Pictures courtesy Vishal Gupta)

plates, finite stiffness of plates) to be computed and included in the appropriate functional element. The functional level representation cannot be visually correlated to the layout of the device. However, this level closely approximates the spring-mass-damper abstraction of an inertial sensor, which designers use extensively in developing inertial sensors. Abstracting away even the functional composition of the MEMS device, the macromodel level, i.e., simply an equation summarizing the input-output relationship of the device, is obtained.

The MEMS design hierarchy is summarized in Figure 2.6 which shows the layout level, atomic level schematic, the functional schematic and the macromodel representation. A design hierarchy is not of use unless the different levels of the hierarchy can be traversed with ease. Broadly, upward motion through the hierarchy, leading to increasing abstraction, is referred to as *extraction* or *verification*. Downward motion, resulting in increased visibility of finer details, is called *synthesis*. Over the past decade, several research efforts, notably at CMU and other universities as well, have not only developed hierarchical representations of MEMS but also demonstrated automated methodologies for various components of the hierarchy traversal.

The NODAS framework, developed at CMU, implements the hierarchical representation of MEMS described above. Schematics of MEMS sensors are created using param-

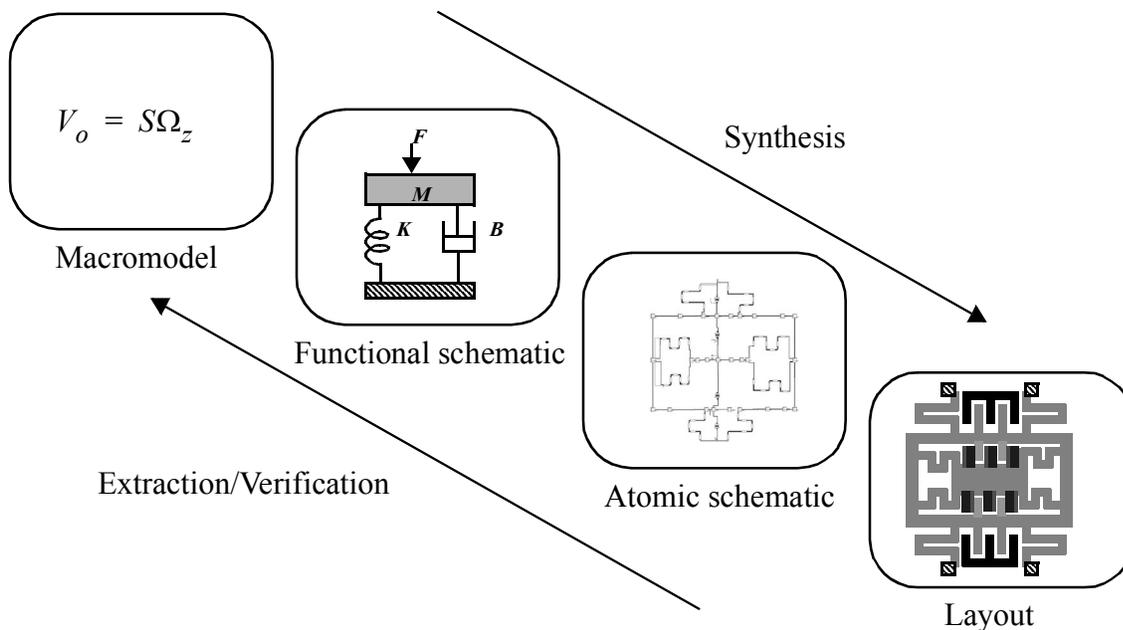


FIGURE 2.6. MEMS design hierarchy

terized elements such as *beams*, *plates*, *anchors* and *combs* and electrical and mechanical independent sources. DC, AC and transient analysis can then be performed on the MEMS sensors. Behavioral simulation of the CMOS-MEMS vertical axis microgyroscope using atomic level elements is used in this thesis to verify hand analysis and to quantify non-ideal effects. The Spectre circuit simulator from Cadence is used as the simulation engine for the behavioral simulations. Modeling approaches for cross-axis coupling in springs, vertical curling and mask misalignment in beams and multi-layer effects in combs are described in the following chapters. The resultant improved models are incorporated into the NODAS library. One of the fundamental tasks which provides the back-bone for such a structured design methodology is building behavioral models. A brief summary of modeling approaches is given in the following sub-section.

2.4.1 Modeling

In the context of the design hierarchy mentioned above, modeling can be viewed as a process of relating parameters at a higher-level of the hierarchy to the parameters at a lower-level. Availability of models which are accurate and can also be evaluated fast enables easy traversal between the different levels possible along both, extraction and synthesis directions. At the level of atomic-elements, modeling involves identification and encoding of significant relationships between geometrical parameters and functional parameters. Examples include derivation of equations for spring stiffness from beam geometrical parameters (width, length, cross-section etc.) and plate mass and moments of inertia from plate length, width and composition. The models implicitly assume a set of manufacturing process-dependent constants for material properties. Those familiar with models in the circuit world can immediately correlate this modeling procedure to the derivation of transistor I-V relationships in terms of geometry and process-dependent doping and material properties. At the same time, those familiar with modeling in the mechanical world can distinguish this process from the building of “solid models” for use in numerical solvers and visualization. Modeling in elastic and electrostatic domains done in this thesis is explained in detail in following chapters. In this section a brief overview of modeling in elastic and electrostatic domains is presented.

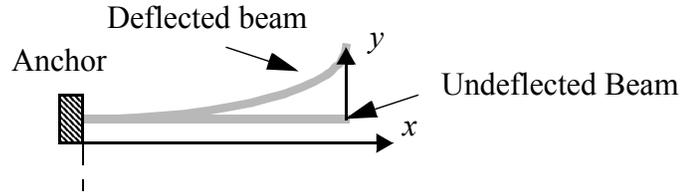


FIGURE 2.7. The direction along the length (x) and the direction of deflection of a beam (y)

2.4.2 Elastic Models

The elastic models in this thesis are based upon linear beam theory [39] wherein forces and bending moments are linearly related to translational and rotational displacements. Shear and non-linear effects are not being considered in this thesis, but are given considerable attention in a parallel work [40]. Linear beam theory is based upon the following fundamental differential equation, which is valid when there are no distributed loads, i.e., forces and moments are applied only at the two end-points of a beam [39]:

$$\frac{d^4 y}{dx^4} = 0 \quad (2.1)$$

where, x is the location along the length of the beam and y is the displacement along one of the two orthogonal directions as shown in Figure 2.7. Energy methods, described in detail in [41], are used to derive equations for spring stiffnesses. A brief introduction to energy methods is given below by way of deriving the compliance matrix for a single beam which is part of a spring.

A number of common spring topologies such as crab-leg, u-shaped and serpentine springs belong to a larger class, in which each spring is a single chain of beams. The analytical advantage in dealing with this class of springs is due to the fact that the forces transmitted through the beams remain invariant from the load point to the anchor point. Figure 2.8 shows a spring composed of 9 beams in a single-chain configuration, attached to a rigid plate at one end and anchored at the other end. The procedure for computing the in-plane compliance matrix for a single beam in the spring 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, assuming linear beam theory, the energy per unit length of the beam is given as:

$$\frac{dU}{dx} = \frac{M_i(x)^2}{2EI_i} \quad (2.2)$$

The total energy of deformation, U , is obtained by integrating over the length of the beam followed by summation over all the beams:

$$U = \sum_{beam\ i=1}^N \int_0^{L_i} \frac{M_i(x)^2}{2EI_i} dx \quad (2.3)$$

where, L_i is the length of the i 'th beam in the flexure, $M_i(x)$ 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 (z axis for in-plane forces and moments about z). The bending moment is a linear function of the forces and moments applied to the end-points of the flexure. Furthermore, it varies linearly with the position along the length of the beam. Therefore, the energy stored in the beam due to displacement is quadratically dependent on the applied forces and moments. In particular, for a single chain of beams (Figure 2.8(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 ζ is given as:

$$\delta\zeta = \frac{\partial U}{\partial F_\zeta} \quad (2.4)$$

where, F_ζ is the force applied in that direction [39]. Similarly, angular displacements can be related to applied moments. The moment $M_i(x)$ being linearly dependent on the applied forces and moments, the displacement is also a linear function of the applied forces, i.e.,

$$\delta\zeta_i = \sum_{\xi} \alpha_{\zeta\xi i} F_{\xi i} \quad (2.5)$$

where $\delta\zeta_i$ is a generalized displacement (includes translation and rotation), F_{ξ_i} is the generalized force in the direction ξ and $\alpha_{\zeta\xi_i}$, the compliance of the i th beam.

The in-plane compliance matrix for a beam, derived in terms of the end-points of the beam, is given as:

$$\begin{bmatrix} \alpha_{xxi} & \alpha_{xyi} & \alpha_{x\phi_zi} \\ \alpha_{yxi} & \alpha_{yyi} & \alpha_{y\phi_zi} \\ \alpha_{\phi_zxi} & \alpha_{\phi_zyi} & \alpha_{\phi_z\phi_zi} \end{bmatrix} \quad (2.6)$$

where,

$$\alpha_{xxi} = \frac{l_i(y_{1i}^2 + y_{2i}^2 + y_{1i}(y_{2i} - 3y_C) - 3y_{2i}y_C - 3y_{2i}^2y_C + 3y_C^2)}{3EI_{zi}}$$

$$\alpha_{yyi} = \frac{l_i(x_{1i}^2 + x_{2i}^2 + x_{1i}(x_{2i} - 3x_C) - 3x_{2i}x_C - 3x_{2i}^2x_C + 3x_C^2)}{3EI_{zi}}$$

$$\alpha_{\phi_z\phi_zi} = \frac{l_i}{EI_{zi}}$$

$$\alpha_{xyi} = \alpha_{yxi} = -\frac{l_i(x_{1i}(2y_{1i} + y_{2i} - 3y_C) + x_{2i}(y_{1i} + 2y_{2i} - 3y_C) - 3x_C(y_{1i} + y_{2i} - 2y_C))}{6EI_{zi}}$$

$$\alpha_{x\phi_zi} = \alpha_{\phi_zxi} = \frac{l_i(y_{1i} + y_{2i} - 2y_C)}{2EI_{zi}}$$

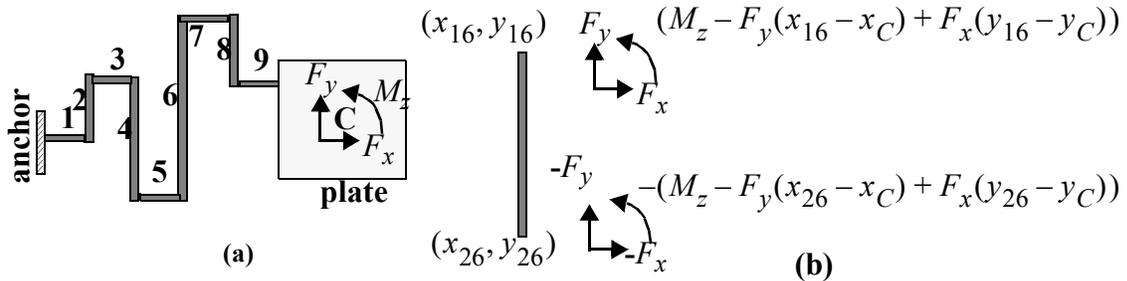


FIGURE 2.8. (a) Spring with single-chain of 9 beams attached to a plate. C is the point of application of force. The other end of the spring is anchored. (b) Free-body diagram of beam 6 and the bending moment along beam 6.

$$\alpha_{y\phi_z i} = \alpha_{\phi_z x i} = -\frac{l_i(x_{1i} + x_{2i} - 2x_C)}{2EI_{zi}}$$

The elements of the in-plane compliance matrix derived above are used in this thesis to derive properties of symmetric springs as well as to compute stiffness values for an entire spring.

2.4.3 Electrostatic Modeling

Electrostatic modeling for gyroscopes mainly involves deriving geometry-based equations for capacitance and force between two or more electrodes in combs, parallel plates and other kinds of sensing or actuation structures. Fundamentally, deriving equations for capacitance involves solving the Laplace equation with appropriate boundary conditions:

$$\nabla^2 V(x, y, z) = 0 \quad (2.7)$$

where, $V(x, y, z)$ is the electrostatic potential which is generally a function of spatial location. Instead of solving the Laplace equation for an entire sensor or actuator, which is seldom practical, usually symmetry considerations are used to break up the sensor into a number of smaller structures which can be solved much more easily. The total energy which is stored in a capacitor with a voltage V applied between the two plates is given as:

$$U = \frac{1}{2}CV^2 \quad (2.8)$$

Once the capacitance has been derived as a function of the relative displacement between the two electrodes the force can be obtained by using the principle of virtual work and differentiating the total energy of the system (if the capacitance is independent of the voltage). For example, the force along the x direction will be given as:

$$F_x = \frac{dU}{dx} = \frac{1}{2} \frac{dC}{dx} V^2 \quad (2.9)$$

It should be noted that the above equation is valid only for *linear* capacitors i.e., where the capacitance is independent of the voltage. For a parallel plate capacitor the capacitance and force are given as:

$$C(x) = \frac{A\epsilon_0}{g+x} \quad (2.10)$$

$$F_x(x) = -\frac{A\epsilon_0}{(g+x)^2} V^2 \quad (2.11)$$

where, A is the area of the electrodes, ϵ_0 is the permittivity of air, g is the designed gap between the two electrodes and x is the displacement along the direction of the gap. For non-parallel-plate structures, significant modeling effort has been spent on computing, analytically or numerically, fringe capacitance terms which are then added onto a parallel plate equation to obtain the total capacitance of the structure. Conformal mapping techniques are frequently used to derive equations for fringe capacitance [42][43]. Detailed modeling of combs in the CMOS-MEMS process is described in Chapter 5.

2.5 Summary

In this chapter a literature survey of microgyroscopes developed over the past decade was presented. Following this, the CMOS-MEMS process was described briefly with particular attention paid to the non-idealities in the process. Having introduced the microgyroscope and the manufacturing process, the rest of the chapter outlines the structured design methodology for MEMS developed at CMU. As part of the outline, the NODAS design framework which implements behavioral simulation of MEMS was summarized. An elementary overview of the modeling procedure for elastic and electrostatic elements in the design hierarchy was also given as a precursor for the more detailed treatment in the following chapters.

Chapter 3. Elastic Modeling

3.1 Introduction

For MEMS gyroscopes to be commercially viable, it is necessary to characterize the non-idealities inherent in them. Non-idealities in microgyroscopes and accelerometers, including offset bias, cross-axis sensitivity and non-linearity, occur due to a combination of undesired mechanical oscillation modes and mismatched sensing capacitances [44]. Undesired oscillation modes are caused by cross-axis coupling in the suspension springs. It has been experimentally observed that real microgyroscopes exhibit elliptical motion [47], depicted in Figure 3.1, as opposed to the expected straight line motion. Coupling from the driven direction to orthogonal directions through non-ideal suspension springs is known to cause the elliptical motion. In order to understand and predict such non-idealities, models for the cross-axis coupling in springs need to be derived. Another important effect, specific to CMOS-MEMS (and other multi-layer) structures, is vertical curling arising from mismatched thermal coefficients of expansion in the different layers. Vertical curling leads to vertical offsets in drive and sense comb structures, resulting in vertical forces and cross-axis coupling. Once again, analytical models are required to estimate the

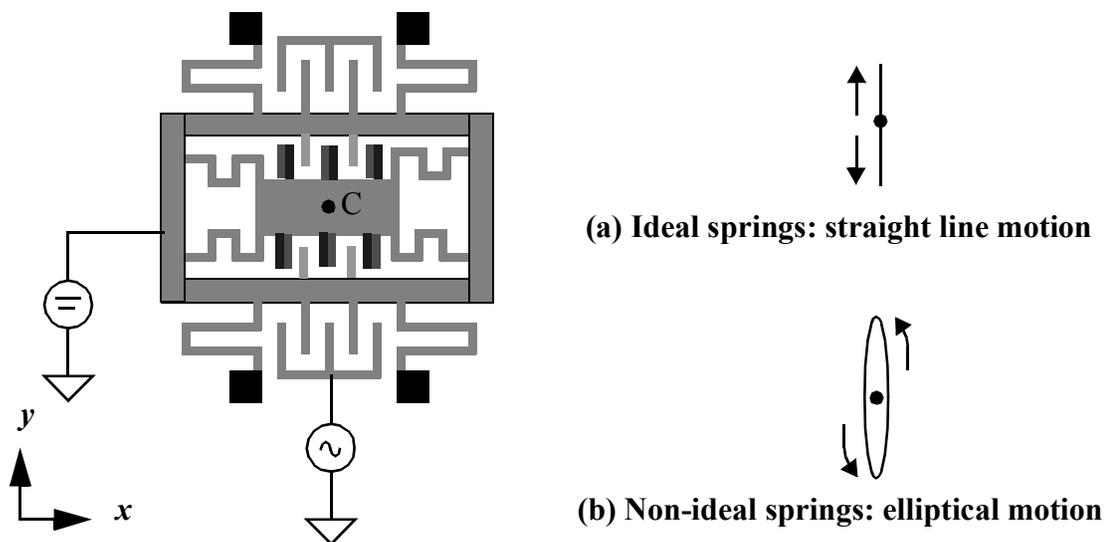


FIGURE 3.1. Outer frame of a gyroscope driven by a sinusoidal voltage source and a DC source. Motion of point C is shown on the right. (a) With ideal springs oscillations are only along the y direction. (b) However, with non-ideal mismatched springs small amount of motion couples to the x direction.

curvature in the design stage either by hand calculations or through behavioral simulations.

Elastic beam theory is an extensively researched area in traditional mechanical and civil engineering. In the context of microgyroscopes, micromechanical springs, and elastic cross-axis coupling, existing beam theory needs to be recast in a form suitable for the MEMS designer. An example of elastic beam theory from a micromechanical viewpoint is a methodology for deriving the lumped-element stiffness models for common micromechanical springs (crab-leg, u-spring, serpentine spring and folded-flexure) as outlined in [41]. In [45], non-linear rod theory has been applied to analyze the vibration modes of a MEMS gyroscope considering the modes to be uncoupled. Coupling among three specific modes of a particular gyroscope structure has been investigated in [46]. None of the above-referred works provide a complete, general and intuitive understanding of elastic cross-axis coupling. The goal of this chapter is to establish a broad understanding of elastic cross-coupling, present general methods for analysis of cross-coupling and consider system-level implications of coupling arising due to individual beams or springs.

In this chapter, the following issues are addressed analytically: in-plane cross-axis coupling, in-plane to out-of-plane coupling due to asymmetric beam cross-sections, vertical and lateral curling of CMOS-MEMS beams. Finite element analyses (FEA) are used throughout this chapter to verify the theory at each stage. The discussion is initiated by formalizing cross-axis coupling through use of a *stiffness matrix* to represent all the stiffness properties of an individual spring, as well as those of a complete system with multiple springs. Energy methods, as introduced in Chapter 2, are used to derive equations for in-plane elastic cross-axis coupling. Cross-axis coupling coefficients are derived for popular spring topologies such as crab-leg, u-spring and serpentine springs. Investigation of options to reduce cross-axis coupling leads to a significant result, which is not revealed here to maintain the readers interest. Following this, equations for out-of-plane coupling in multi-layer beams are derived using Euler-Bernoulli beam theory. The discussion on cross-axis coupling is concluded by examining the impact of individual springs on system level coupling. A geometrical interpretation of the in-plane cross-axis coupling and the relation between the stiffness matrix, the position and orientation of the so-called *principal axes of elasticity* and the observed motion coupling is illustrated through examples. In

the later sections of the chapter, existing thermal multi-morph theory for cantilevers is used as the basis to derive a new macromodel incorporating multi-directional curvature of non-cantilever structures. The new macromodel is developed so as to be completely compatible with existing behavioral models for beams and the NODAS behavioral simulation framework. It is verified by comparison with experimental measurements on test structures which exhibit temperature dependent vertical and lateral curvature.

3.2 Stiffness Matrices

A number of gyroscopes (as well as other inertial sensors) as mentioned in [2] are made up of a mechanical proof-mass suspended by four springs. Each spring can be represented by a 6X6 lumped-element symmetric stiffness matrix (referred to as k in Figure 3.2). The overall system stiffness matrix is obtained as a summation of the individual stiffness matrices. The 6 diagonal terms in the stiffness matrix represent the stiffness of the springs in the translational and rotational directions. The off-diagonal terms represent coupling between different directions. The stiffness matrix k can also be viewed as the combination of 4 3×3 sub-matrices. Two of the sub-matrices are referred to as k_{ip} , the in-plane stiffness matrix and k_{op} , the out-of-plane stiffness matrix. The remaining two sub-matrices are shown as the shaded portion in the k matrix in Figure 3.2. When all the

Shaded elements are zero if there is no out-of-plane coupling

$$k = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} & k_{x\phi_x} & k_{x\phi_y} & k_{x\phi_z} \\ k_{yx} & k_{yy} & k_{yz} & k_{y\phi_x} & k_{y\phi_y} & k_{y\phi_z} \\ k_{zx} & k_{zy} & k_{zz} & k_{z\phi_x} & k_{z\phi_y} & k_{z\phi_z} \\ k_{\phi_x x} & k_{\phi_x y} & k_{\phi_x z} & k_{\phi_x \phi_x} & k_{\phi_x \phi_y} & k_{\phi_x \phi_z} \\ k_{\phi_y x} & k_{\phi_y y} & k_{\phi_y z} & k_{\phi_y \phi_x} & k_{\phi_y \phi_y} & k_{\phi_y \phi_z} \\ k_{\phi_z x} & k_{\phi_z y} & k_{\phi_z z} & k_{\phi_z \phi_x} & k_{\phi_z \phi_y} & k_{\phi_z \phi_z} \end{bmatrix}$$

In-plane stiffness matrix

$$k_{ip} = \begin{bmatrix} k_{xx} & k_{xy} & k_{x\phi_z} \\ k_{yx} & k_{yy} & k_{y\phi_z} \\ k_{\phi_z x} & k_{\phi_z y} & k_{\phi_z \phi_z} \end{bmatrix}$$

Out-of-plane stiffness matrix

$$k_{op} = \begin{bmatrix} k_{zz} & k_{z\phi_x} & k_{z\phi_y} \\ k_{\phi_x z} & k_{\phi_x \phi_x} & k_{\phi_x \phi_y} \\ k_{\phi_y z} & k_{\phi_y \phi_x} & k_{\phi_y \phi_y} \end{bmatrix}$$

FIGURE 3.2. Elements of the stiffness matrix k and the in-plane and out-of-plane sub-matrices k_{ip} and k_{op} . This symmetric matrix has 21 distinct terms. If the shaded elements are zero, the number of distinct non-zero terms reduces to 12.

four spring geometries are identical, and the layout is two-fold symmetric, the system stiffness matrix is diagonal because the cross-axis (off-diagonal) terms of the four springs cancel out exactly after summation. The eigenmodes of the system are perfectly aligned with the reference coordinate frame. In inertial sensors, the sensitive axis is usually aligned with one of the axes of the reference coordinate frame. Due to manufacturing variations, which are inevitable in any IC fabrication process, the four spring dimensions are not perfectly matched. Therefore, there are residual off-diagonal terms in the system stiffness matrix after summation. For example, the widths of the four springs may differ slightly leading to coupling between the three in-plane modes (x , y and ϕ_z) and thus, non-zero off-diagonal elements in the in-plane part of the stiffness matrix k_{ip} . Alternatively, the beams forming the spring may have an asymmetric cross-section resulting in in-plane to out-of-plane coupling i.e., non-zero shaded elements in Figure 3.2. In both cases elliptical motion of the proof-mass [47], instead of the expected straight line motion, results. The elliptical motion can be understood as a displacement (which could be rotational or translational) of the eigenmodes of the system from the reference coordinate frame. Two different approaches are used in the following sections to obtain the off-diagonal terms within the in-plane and out-of-plane sub-matrices, k_{ip} and k_{op} respectively, and to model coupling between the in-plane and out-of-plane directions. The former lend themselves to direct derivation of the terms in the stiffness matrix, while the latter are more easily characterized and understood by a rotation of the principal axes of stiffness. The following section on modeling explains the approach to derive the 6 off-diagonal terms in the in-plane and out-of-plane sub-matrices.

3.3 Modeling

The procedure to derive symbolic linear relationships expressing displacements in terms of the forces and moments applied has been described in detail in [41] and briefly explained in Chapter 2. Energy methods are used to obtain a system of linear equations symbolically expressing the displacements in terms of the forces and moments. The system of symbolic linear relationships are solved to obtain the elements of the stiffness matrices. Symbolic manipulations were done using the Mathematica program [48]. An

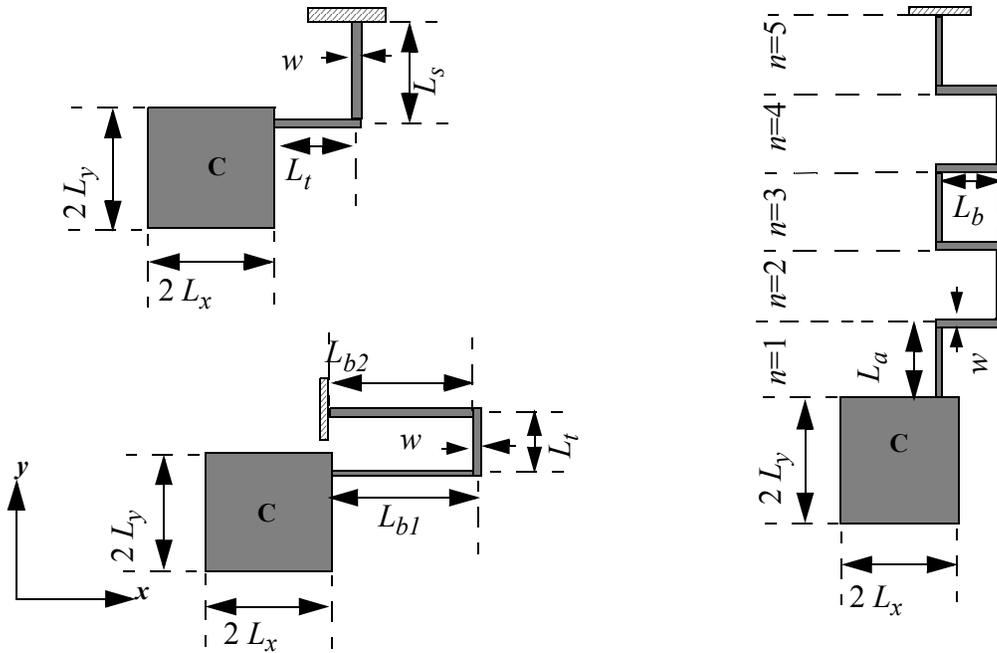


FIGURE 3.3. Design variables for crab-leg-spring, U-spring and serpentine spring with proof-mass. The external forces and moments are applied at C, the centroid of the plate, with only one spring in the analysis so that all the cross-axis terms can be clearly observed.

example Mathematica script to derive the stiffness matrix for the crab-leg spring is given in Appendix A1. The stiffness matrix equations for the three types of springs, crab-leg, u-spring and serpentine spring are described below.

The three types of springs are parameterized as shown in Figure 3.3. The models for diagonal elements (*i.e.*, the translational spring constants) have been derived previously [41]. Here the same technique is applied to derive the off-diagonal stiffness constants. Applying the boundary conditions, as shown in Figure 3.4, a set of linear equations in terms of the applied forces and moments and the unknown displacement is obtained. Solving the set of equations yields a linear relationship between the displacement and applied force for the cross-axis spring constant of interest [41]. The constant of proportionality gives the spring constant as a function of the physical dimensions of the spring. The models for the out-of-plane cross-axis spring constants are similarly derived. The equations for the stiffness constants are presented below. In order to preserve readability, the equations for the off-diagonal elements in the out-of-plane matrix k_{op} , which are considerably more complex, are given in Appendix A2.

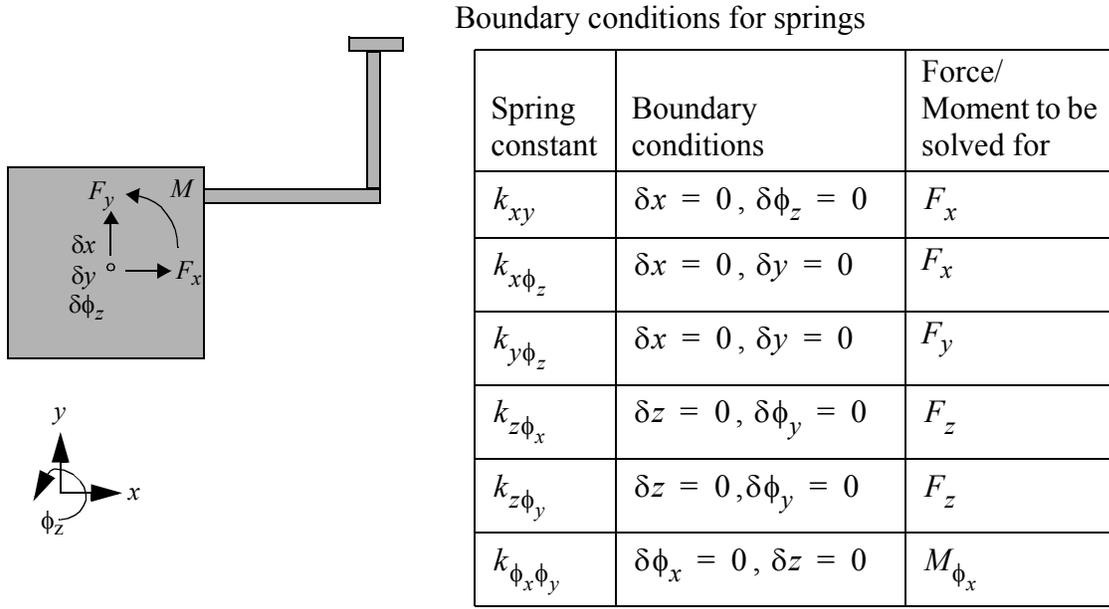


FIGURE 3.4. Forces and moments applied at the centroid of a proof-mass attached to the free end of a crab-leg. Boundary conditions are applied as equality constraints on the three displacements.

For the crab-leg, the analytical models for elastic coupling between x, y and ϕ_z modes for one spring are:

$$k_{xy} = \frac{9EI_{zt}I_{zs}}{L_tL_s(I_{zt}L_s + I_{zs}L_t)} \quad (3.1)$$

$$k_{x\phi_z} = \frac{-3EI_{zs}\left(\left(I_{zs}L_t^2 + 4I_{zt}L_sL_t\right)L_y - 3I_{zt}L_s^2L_x - I_{zt}L_tL_s^2\right)}{L_tL_s^3(I_{zt}L_s + I_{zs}L_t)} \quad (3.2)$$

$$k_{y\phi_z} = \frac{3EI_{zt}\left(-3I_{zs}L_t^2L_y + \left(I_{zt}L_s^2 + 4I_{zs}L_sL_t\right)L_x + I_{zt}L_s^2L_t + 2I_{zs}L_sL_t^2\right)}{L_sL_t^3(I_{zt}L_s + I_{zs}L_t)} \quad (3.3)$$

where, I_{zt} , I_{zs} and I_{yt} , I_{ys} are the moments of inertia of the crab-leg-beams about their individual z axis and y respectively.

For u-springs:

$$k_{xy} = \frac{9EI_{zb}I_{zt}(L_{b1} - L_{b2})(2I_{zt}L_{b1}L_{b2} + I_{zb}(L_{b1} + L_{b2})L_t)}{D_{uip}} \quad (3.4)$$

where, D_{uip} is the common denominator for all the in-plane stiffness constants of the u-spring.

$$D_{uip} = L_t \left(\begin{array}{c} 3I_{zt}^2 L_{b1} L_{b2} (L_{b1}^3 + L_{b2}^3) + I_{zb}^2 (L_{b1}^3 + L_{b2}^3) L_t^2 + \\ I_{zb} I_{zt} (L_{b1}^4 + 4L_{b1}^3 L_{b2} + 3L_{b1}^2 L_{b2}^2 + 4L_{b1} L_{b2}^3 + L_{b2}^4) L_t \end{array} \right) \quad (3.5)$$

Assuming $L_{b1} \sim L_{b2}$ and $L_t \ll L_{b1}$, the derived analytical model is simplified to get

$$k_{xy - uspring} = \frac{3EI_{zt}(L_{b1} - L_{b2})}{L_t L_{b1}^3} \quad (3.6)$$

$$k_{x\phi_z} = \frac{-3EI_{zb} I_{zt} \left(\begin{array}{c} \left(\frac{I_{zt}}{L_t} \left((L_{b1}^2 + L_{b2}^2)^2 + 4L_{b1} L_{b2} (L_{b1} - L_{b2})^2 \right) + 4I_{zb} (L_{b1}^3 + L_{b2}^2 L_{b1}) \right) L_y + \\ (-6I_{zt} L_{b1}^2 L_{b2} + 6I_{zt} L_{b1} L_{b2}^2 - 3I_{zb} L_{b1}^2 L_t + 3I_{zb} L_{b2}^2 L_t) L_x + \\ I_{zt} (L_{b2}^4 + 3L_{b1}^2 L_{b2}^2 - 2L_{b1}^3 L_{b2}) - I_{zb} L_t (-L_{b1}^3 + 3L_{b1} L_{b2}^2 + 2L_{b2}^3) \end{array} \right)}{D_{uip}} \quad (3.7)$$

$$k_{y\phi_z} = \frac{3EI_{zb} \left(\begin{array}{c} I_{zt} (-6I_{zt} L_{b1}^2 L_{b2} + 6I_{zt} L_{b1} L_{b2}^2 - 3I_{zb} L_{b1}^2 L_t + 3I_{zb} L_{b2}^2 L_t) L_y + \\ (I_{zb}^2 L_t^3 + 12I_{zt}^2 L_{b1} L_{b2} L_t + 4I_{zb} I_{zt} L_{b1} L_t^2 + 4I_{zb} I_{zt} L_{b2} L_t^2) L_x + \\ L_t (I_{zb}^2 L_t^2 L_{b1} + I_{zb} I_{zt} L_t (2L_{b1}^2 + 4L_{b1} L_{b2} + L_{b2}^2) + 6I_{zt}^2 L_{b1} L_{b2}) \end{array} \right)}{D_{uip}} \quad (3.8)$$

For a serpentine spring, the k_{xy} for even n is given as:

$$k_{xy} = \frac{36EI_{za} I_{zb}^2}{D_{soip}} \quad (3.9)$$

where, D_{soip} is the common denominator for all the terms in the in-plane stiffness matrix for a serpentine spring with even n ,

$$D_{soip} = L_b(L_b^2 I_{za}^2 (n-2)(n-1)^2 + 3L_a^2 I_{zb}^2 (n^3 - 3n) + 2L_a L_b I_{za} I_{zb} n(2n^3 - 5n^2 + 3n)) \quad (3.10)$$

$$k_{x\phi_z} = \frac{-6EI_{za}I_{zb} \left(\begin{array}{c} (6nI_{zb}L_aL_b + 2(n-1)I_{za}L_b^2)L_y + \\ (-6nI_{zb}L_a^2)L_x + \\ 3n(n-1)I_{zb}L_a^2L_b + n(n-1)I_{za}L_aL_b^2 \end{array} \right)}{(nL_a^2)D_{soip}} \quad (3.11)$$

$$k_{y\phi_z} = \frac{6EI_{za}I_{zb} \left(\begin{array}{c} -6I_{zb}L_bL_y + \\ (2n^2I_{zb}L_a + (2I_{za}L_b(n^2 - 3n + 2)))L_x + \\ (n^2 - 3n)I_{za}L_aL_b \end{array} \right)}{L_b D_{soip}} \quad (3.12)$$

and for odd n

$$k_{xy} = 0 \quad (3.13)$$

$$k_{x\phi_z} = \frac{-6EI_{za}I_{zb}(2L_y + nL_a)}{(nL_a^2)(n^2L_aI_{zb} + (n^2 - 3n + 2)L_bI_{za})} \quad (3.14)$$

$$k_{y\phi_z} = \frac{6EI_{za}I_{zb}((2(n-1)I_{za}L_b + 2nI_{zb}L_a)L_x + (n-1)I_{za}L_b^2 + (n-1)I_{zb}L_aL_b)}{(n-1)L_b^2((n-1)I_{za}^2L_b^2 + 4I_{za}I_{zb}nL_aL_b + 3(n-1)I_{zb}^2L_a^2)} \quad (3.15)$$

Though the above equations for the spring stiffnesses seem to be lengthy and complex polynomials the following sets of observations help to discern their underlying structure and usefulness.

Observations on the terms in the stiffness constants

1. All the stiffness constants are fractions of polynomials in the moments of inertias (I_z) and the lengths of the beams (L).

2. All the stiffness constants for a particular type of spring share a common denominator. This is not surprising because the denominator is simply the determinant of the compliance matrix of that spring. Note that in some cases, like the serpentine spring with odd n , this is not apparent because of canceling of common factors between the numerator and the denominator.
3. As would be obtained from a dimensional analysis, the power of moment of inertia in the denominator is lesser than that in the numerator by 1, while, the power of length in the denominator is greater than that in the numerator by 3.

The next set of observations is focussed on the elastic coupling term between x and y , i.e., the k_{xy} term.

1. k_{xy} for all springs is independent of L_x and L_y i.e., the location of the load point.
2. k_{xy} is never zero for a crab-leg spring and for a serpentine spring with even n .
3. $k_{xy} = 0$ for a serpentine spring with odd n .
4. For a u-spring $k_{xy} = 0$ if and only if $L_{b1} = L_{b2}$ i.e., if the two parallel beams in the u-spring are identical.

The final observations are about the elastic coupling terms between the translational modes x , y and the rotational mode ϕ_z : $k_{x\phi_z}$ and $k_{y\phi_z}$. Comparison of the $k_{x\phi_z}$ and $k_{y\phi_z}$ of the crab-leg, serpentine and the u-spring reveals a common format which is a sum of three terms. Generalized equations for $k_{x\phi_z}$ and $k_{y\phi_z}$ can be written as:

$$k_{x\phi_z} = -k_{xx}L_y + k_{xy}L_x + k_{x\phi_z0} \quad (3.16)$$

$$k_{y\phi_z} = -k_{xy}L_y + k_{yy}L_x + k_{y\phi_z0} \quad (3.17)$$

In the above equations, $k_{x\phi_z0}$ and $k_{y\phi_z0}$ are the elastic coupling constants between the translational and rotational modes when $L_x = 0$ and $L_y = 0$ i.e., when the load is applied directly to the springs. Naturally, $k_{x\phi_z0}$ and $k_{y\phi_z0}$ are only dependent on the spring geometry itself and not on the dimensions of the plate to which the spring is con-

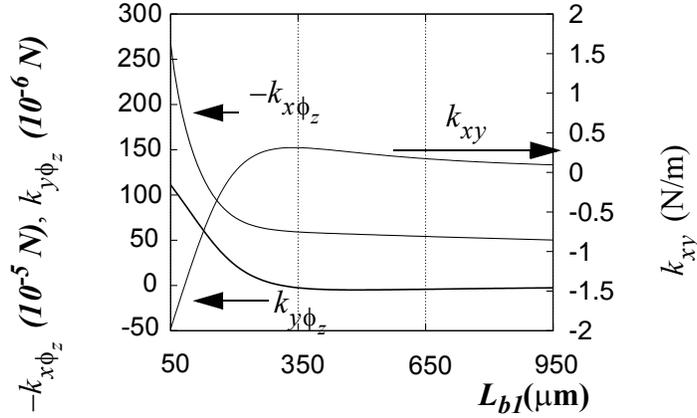


FIGURE 3.5. Trends in the variation of in-plane spring constants for the U-spring for varying beam lengths (L_{b1}). The design variables are set to: $w=2.0 \mu\text{m}$, $L_t=10.0 \mu\text{m}$, $L_{b2}=200.0 \mu\text{m}$

nected. Depending on the type of spring and the distances L_x and L_y , one may suitably approximate $k_{x\phi_z}$ and $k_{y\phi_z}$ by removing one or even two terms from equations (3.16) and (3.17). For example in case of the crab-leg spring, when $L_p, L_s \ll L_x, L_y$, the $k_{x\phi_z0}$ and $k_{y\phi_z0}$ terms can be neglected.

The above models are for a single spring. These models suggest design directions for reducing k_{xy} , $k_{x\phi_z}$ and $k_{y\phi_z}$, and thereby, the device non-idealities. It is possible to eliminate the nominal system K_{xy} by symmetrically placing four springs. However, this will only eliminate K_{xy} in the nominal case in which all four springs are perfectly matched. Manufacturing variations are commonly modeled as functions of wafer position, implying that closely placed beams (as in the same spring) have less width variation than beams which are farther apart (like those on two different springs). In addition to eliminating nominal system K_{xy} , long range width variations can also be nullified by designing the two beams with equal lengths and widths for a U-spring or by choosing n to be odd for a serpentine spring. In Figure 3.5 it is seen that it is possible to design the U-spring such that k_{xy} is very close to zero (near $L_{b1} = 200\mu\text{m}$). Similar trends for the serpentine spring with even n are shown in Figure 3.6. The plots also show that when k_{xy} is very close to

zero, $k_{x\phi_z}$ and $k_{y\phi_z}$ can have significant values indicating that the springs have to be designed keeping in mind the particular requirement. In microgyroscopes it is usually more important to eliminate k_{xy} than to eliminate $k_{x\phi_z}$ and $k_{y\phi_z}$ because of several reasons which will be explained in detail in Chapter 7. One of the important fundamental reasons for this is the fact that the physical operating principle of the microgyroscope is based upon coupling between x and the y modes.

In the next section the validity of the equations derived is verified by comparison with finite element analyses.

3.4 Model Verification

The models derived above are verified by comparison to FEA results. Assuming that the widths of all the beams in a spring are equal, there are three remaining design variables for the crab-leg, four for the u-spring and four for the serpentine spring. The distances of the spring attachment point from the load point, L_x and L_y are held constant at $100\mu m$. The Abaqus solver was used for FEA [49]. Convergence analysis was done to determine the granularity of the finite element mesh that was required. Consequently, each beam was split into 40 divisions along the length and 10 divisions along the width. FEA with 3D quadratic brick elements was done on 8 crab-leg designs, 16 u-spring designs and 8 ser-

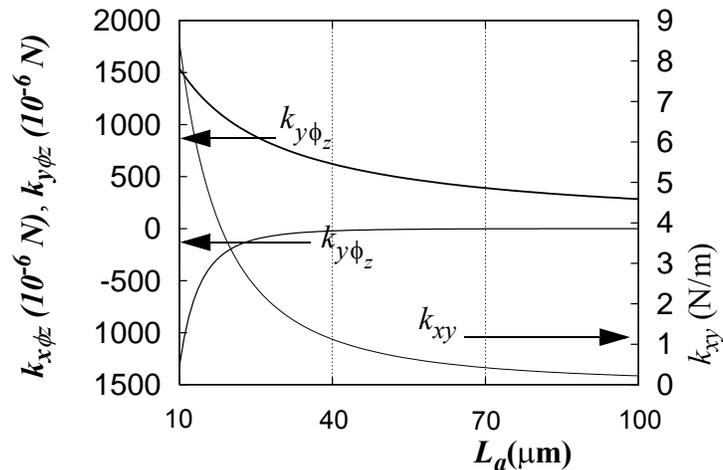


FIGURE 3.6. Trends in the variation of in-plane spring constants for the serpentine-spring for varying beam lengths (a). The design variables are set to: $w=2.0 \mu m$, $L_b=20.0 \mu m$, $n=4$.

pentine spring designs. The widths of the crab-leg and u-springs were set to $2.0 \mu m$ and $5.0 \mu m$ to span the entire range of typically used designs. The length of the crab-leg and u-spring beams was set to $40 \mu m$ and $400 \mu m$. Beams longer than $400 \mu m$ are generally not manufacturable. At the lower limit, beams shorter than $40 \mu m$ are seldom used since they lead to very stiff suspensions. For the serpentine spring a different set of metrics were used to decide the range of spring dimensions. Serpentine springs are normally used when more number of turns are required, either because the spring is too stiff otherwise, or, because the maximum length of the beams is constrained by manufacturing or other considerations. Therefore, the maximum spring width used in serpentine spring designs was only $4.0 \mu m$, and the maximum length of beams used was $100 \mu m$. It should be noted that the results of spring stiffness, span nearly 4 decades, because the spring stiffness are cubic in terms of the beam width and length. The results are summarized in Table 3.1 for the crab-

Table 3.1 Comparison of FEA and analytical stiffness (in-plane) values for the crab-leg spring

w	L_t	L_s	k_{xy} (N/m)			$k_{x\phi_z}$ ($\times 10^{-6}$ N) (N)			$k_{y\phi_z}$ ($\times 10^{-6}$ N)		
			μm	μm	μm	A	S	Error (%)	A	S	Error (%)
2	40	40	14.4	14.8	-2.71	-19.2	-10.5	83.2	1226	1237	-0.89
2	400	40	0.273	0.273	-0.18	-427	-426	0.42	-0.893	-0.848	5.42
2	40	400	0.273	0.273	-0.18	15.9	15.9	0.06	1030	1027	0.29
2	400	400	0.0154	0.0155	-0.90	1.82	1.85	-1.25	6.84	6.87	-0.57
5	40	40	202	207	-2.70	336	475	-29.3	1.78e4	1.78e4	0.00
5	400	40	4.07	3.94	3.17	-5733	-5478	4.65	-1.811	3.73	-148
5	40	400	4.07	4.00	1.60	239	226	5.48	1.47e4	1.41e4	4.61
5	400	400	0.237	0.240	-1.25	28.9	29.3	-1.60	106	107	-0.56

leg, Table 3.2 for the u-spring and Table 3.3 for the serpentine spring respectively. For the

Table 3.2 Comparison of FEA and analytical stiffness (in-plane) values for the u-spring

w	L_{b1}	L_{b2}	L_t	k_{xy} (N/m)			$k_{x\phi_z}$ ($\times 10^{-6}$ N) (N)			$k_{y\phi_z}$ ($\times 10^{-6}$ N)		
				μm	μm	μm	A	S	Error (%)	A	S	Error (%)
2	40	40	10	0	0.701	NA	-3482	-3407	2.20	1013	968.7	4.57
2	40	400	10	-0.623	-0.605	3.01	-2518	-2466	2.11	30	29.05	3.27
2	400	40	10	0.623	0.612	1.81	-1948	-1866	4.39	-16.78	-16.27	3.13
2	400	400	10	0	7.7e-4	NA	-371.5	-368.7	0.76	4.969	4.954	0.30
2	40	40	50	0	0.0509	-NA	-226.1	-223.1	1.34	694.3	692.1	0.32
2	40	400	50	-0.145	-0.145	-0.28	-185.2	-186.8	-0.86	10.04	10.07	-0.30
2	400	40	50	0.144	0.146	-0.69	-70.04	-69.28	1.10	3.184	3.207	-0.72
2	400	400	50	0	1.0e-4	NA	-27.41	-27.44	-0.11	4.547	4.566	-0.42
5	40	40	10	0	15	NA	-3.33e4	-3.11e4	6.94	1.38e4	1.25e4	10.8
5	40	400	10	-7.71	-6.88	12.0	-2.46e4	-2.24e4	10.2	370.8	327.3	13.3
5	400	40	10	7.71	7.07	9.00	-1.78e4	-1.53e4	16.0	-165.5	-140.4	17.9
5	400	400	10	0	0.0203	NA	-3697	-3572	3.50	76.44	75.52	1.22
5	40	40	50	0	1.968	-NA	-3035	-2876	5.53	9817	9530	3.01
5	40	400	50	-2.10	-2.08	1.01	-2530	-2555	-0.98	146.8	145.4	0.96
5	400	40	50	2.10	2.11	-0.57	-878	-832	5.53	57.79	58.54	-1.28
5	400	400	50	0	0.0056	NA	-380.9	-379.7	0.32	70.09	69.98	0.16

Table 3.3 Comparison of FEA and analytical stiffness (in-plane) values for the serpentine spring

w	L_a	L_b	k_{xy} (N/m)			$k_{x\phi_z}$ ($\times 10^{-6}$ N)			$k_{y\phi_z}$ ($\times 10^{-6}$ N)		
			μm	μm	μm	A	S	Error (%)	A	S	Error (%)
2	10	10	26.9	29.5	-8.71	-1064	-1033	3.00	7597	8090	-6.09
2	10	100	0.251	0.258	-2.83	-591	-595	-0.54	41.56	42.72	-2.72
2	100	10	0.471	0.471	0.00	10.5	10.5	0.67	1133	1128	0.44
2	100	100	0.0269	0.0273	-1.32	-5.91	-5.93	-0.30	11.64	11.74	-0.85
4	10	10	216	252	-14.4	-8297	-8111	2.29	6.24e+04	6.58e+04	-5.27
4	10	100	2.01	2.12	-5.28	-4730	-4716	0.30	337.5	343.5	-1.75

Table 3.3 Comparison of FEA and analytical stiffness (in-plane) values for the serpentine spring

w	L_a	L_b	$k_{xy}(\text{N/m})$			$k_{x\phi_z}(\times 10^{-6}\text{N})$			$k_{y\phi_z}(\times 10^{-6}\text{N})$		
			A	S	Error (%)	A	S	Error (%)	A	S	Error (%)
μm	μm	μm									
4	100	10	3.77	3.65	3.29	88.1	74.6	18.11	9273	8503	9.06
4	100	100	0.216	0.220	-2.22	-47.1	-47.7	-1.32	94.68	92.76	2.07

crab-leg spring and the U-spring it is seen from the FEA results that, for beam widths of 2 μm , when the beam lengths are at least 0.75 times the plate dimensions, all the models match the FEA values to within 10%. For smaller beam lengths the plate ceases to be rigid. Plate deformations are not considered in this spring modeling exercise. A brief explanation of the procedure used to obtain k_{xy} through finite elements is in order here. Geometrical boundary conditions are applied on the finite element model such that the anchor points have zero displacement in all degrees of freedom and the load point is applied a fixed (1 nm) displacement in only the x direction, the y and z directions being kept at 0 displacements. Then, the reaction force at the load point in the y direction is used to compute the k_{xy} . Note that for suspensions which are very stiff in the x direction and which do not have a significantly large k_{xy} , the ratio of the reaction force in the y direction to that in the x direction can be much smaller than 1 (~ 0.01 or even smaller). Finite numerical precision sets a lower limit on the smallest reaction force that can be computed, thereby limiting the smallest value of k_{xy} that can be precisely obtained from finite element analysis. For the serpentine spring it was not possible to obtain accurate values of k_{xy} from FEA when the values were low due to this reason. All the models other than $k_{y\phi_z}$ match FEA results to within 10% for beam widths of 2 μm . The $k_{y\phi_z}$ also matches within 10% except when L_b is much greater than L_a .

As noted previously, k_{xy} is of more significance than $k_{y\phi_z}$ and $k_{x\phi_z}$, and is, therefore, analyzed further. Keeping all other design variables constant, the variation of k_{xy} with the beam length was studied. The beam widths are kept constant at 2.0 μm . In the crab-leg

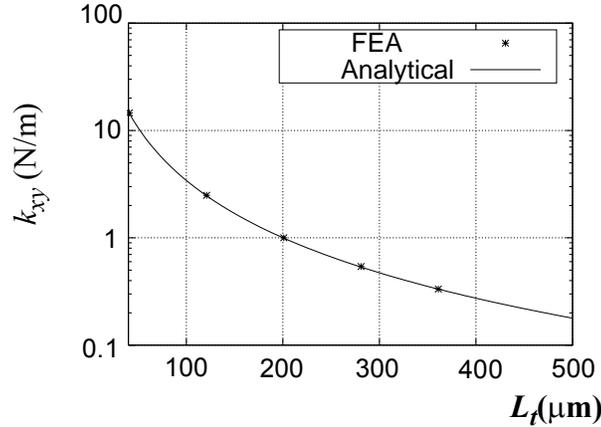


FIGURE 3.7. Comparison of analytical model and FEA for crab-leg-spring k_{xy} for varying crab-leg thigh lengths (L_t). The design variables are set to: $w=2.0 \mu\text{m}$, $L_s=50.0 \mu\text{m}$

plots, the shin length $L_s = 50 \mu\text{m}$. In the u-spring plots, since equal beam lengths lead to zero k_{xy} , a constant difference $L_{b2} - L_{b1} = 30 \mu\text{m}$ was maintained. For the serpentine springs, As seen in Figure 3.7, Figure 3.8 and Figure 3.9 respectively the analytical models match the FEA values to within 2% for the crab-leg, 5% for the U-spring and 9% for the serpentine-spring. Particularly evident is the large range of values over which the models match the FEA values.

Although it appears as if the springs exhibit less cross-axis coupling as the lengths of the beams increase, it should be kept in mind that the stiffness coefficients in the principal directions k_{xx} and k_{yy} also reduce as the lengths of the beams increase. Therefore, while

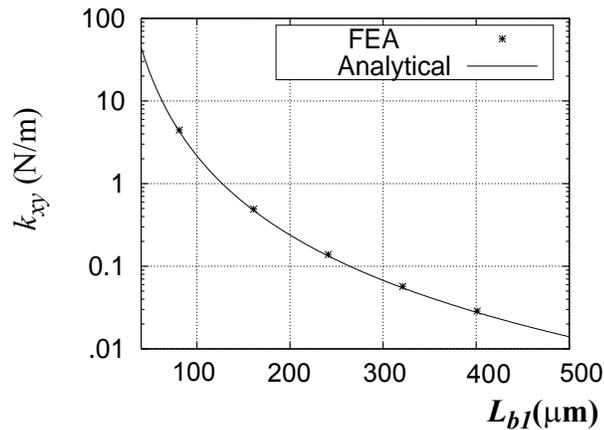


FIGURE 3.8. Comparison of analytical model and FEA for u-spring k_{xy} for varying U-spring beam lengths (L_{b1}). The design variables are set to: $w=2.0 \mu\text{m}$, $L_t=10.0 \mu\text{m}$, $L_{b2}=L_{b1}-30.0 \mu\text{m}$

the coefficient k_{xy} is useful to calculate actual values of cross-axis coupling, it does not facilitate instant comparison of cross-axis coupling in different spring designs. For this purpose, the correlation between the cross-axis coupling coefficient and the principal axes of elasticity is more useful and is explained in Section 3.8. In the following section, an accelerometer simulation is used to illustrate the practical significance of the cross-axis coupling coefficients.

3.5 Accelerometer Simulation

A macromodel for the serpentine spring was incorporated in NODAS [50]. AC analysis of a proof-mass suspended by four serpentine springs was done using the spring macromodel as well as individual beam elements. The macromodel-based simulation (with $n=4$) was about 5 times faster than the individual beam element-based simulation. For higher n , the speedup will be greater.

The serpentine spring, proof-mass structure described above was employed as a y -accelerometer. Input accelerations were applied in both the x and the y directions. Mode coupling is observed in FEA when diagonal springs are identical and one pair of diagonal springs is wider than the other pair. The widths of the springs are indicated on the side in microns. This configuration was simulated using the serpentine spring macromodel in NODAS. Input accelerations, which are out of phase to easily distinguish their effects, are applied in y and x directions. As expected, a significant cross-axis sensitivity (resulting

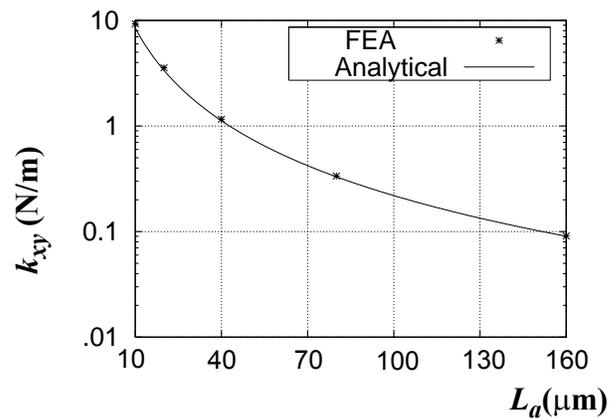


FIGURE 3.9. Comparison of analytical model and FEA for serpentine-spring k_{xy} for varying serpentine-spring beam lengths (a). The design variables are set to: $w=2.0 \mu\text{m}$, $L_b=20.0 \mu\text{m}$, $n=4$

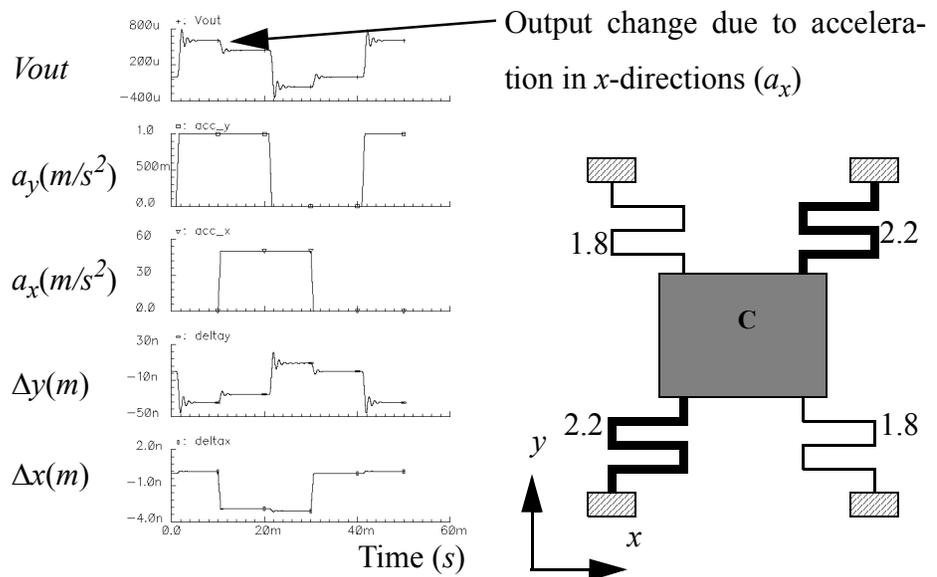


FIGURE 3.10. NODAS simulation of cross-axis sensitivity in y -accelerometer. The structure of the accelerometer with four serpentine springs is shown on the side.

from mode coupling) is seen in the V_{out} waveform in Figure 3.10. In this example the numbers chosen for the spring widths are deliberately biased so that the effect of cross-axis coupling is exaggerated and, therefore, visually discernible. In real accelerometers, the same phenomenon causes cross-axis couplings which are about one to two order of magnitudes smaller than the main output.

Having emphasized the significance of analyzing the cross-axis coupling coefficients, it is natural to examine ways of reducing or even completely eliminating cross-axis coupling. In the next section, special properties of symmetric springs, with respect to cross-axis coupling, are stated and proved.

3.6 Symmetric springs

Recalling that the u-spring with equal beam lengths and the serpentine spring with odd n have $k_{xy} = 0$ leads one to wonder if the two springs have anything in common. The answer is that both the springs have an axis of symmetry as shown in Figure 3.11. This common property can be generalized for all springs which are in the form of a single chain of beams as follows:

“The cross-axis coupling between the two in-plane translational directions is zero for any spring which has an axes of symmetry”

In other words, any single-chain spring that has an axis of symmetry has $k_{xy} = 0$. This important property is mathematically proven below.

Consider a single-chain spring consisting of $2n$ beams. Note that a symmetric spring with odd number of beams can always be converted to even number of beams by splitting the central beam. The compliance matrix, described in Chapter 2, is used here to compute the k_{xy} of the entire spring. Let the positions of the end-points of the i th beam in the spring be x_{1i}, y_{1i} and x_{2i}, y_{2i} . Let l_i, w_i, t and I_{zi} be the length, width, thickness and the moment of inertia of the beam respectively. Let the load point be x_C, y_C . Then the 6 distinct terms of the compliance matrix are given as:

$$\alpha_{xxi} = \frac{l_i(y_{1i}^2 + y_{2i}^2 + y_{1i}(y_{2i} - 3y_C) - 3y_{2i}y_C - 3y_{2i}y_C + 3y_C^2)}{3EI_{zi}}$$

$$\alpha_{yyi} = \frac{l_i(x_{1i}^2 + x_{2i}^2 + x_{1i}(x_{2i} - 3x_C) - 3x_{2i}x_C - 3x_{2i}x_C + 3x_C^2)}{3EI_{zi}}$$

$$\alpha_{\phi_z\phi_zi} = \frac{l_i}{EI_{zi}}$$

$$\alpha_{xyi} = \alpha_{yxi} = -\frac{l_i(x_{1i}(2y_{1i} + y_{2i} - 3y_C) + x_{2i}(y_{1i} + 2y_{2i} - 3y_C) - 3x_C(y_{1i} + y_{2i} - 2y_C))}{6EI_{zi}}$$

$$\alpha_{x\phi_zi} = \alpha_{\phi_zxi} = \frac{l_i(y_{1i} + y_{2i} - 2y_C)}{2EI_{zi}}$$

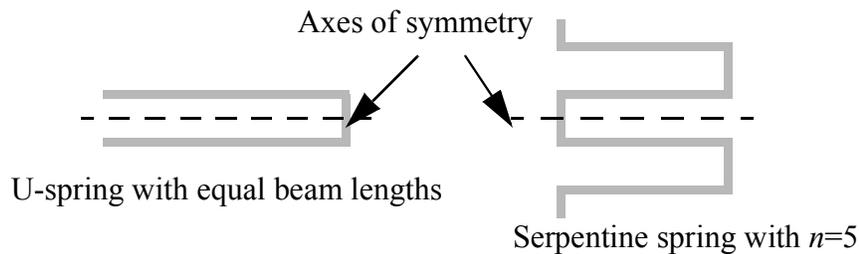


FIGURE 3.11. Axes of symmetry for the u-spring and serpentine spring

$$\alpha_{y\phi_{zi}} = \alpha_{\phi_{zi}x_i} = -\frac{l_i(x_{1i} + x_{2i} - 2x_C)}{2EI_{zi}}$$

For a beam whose length is along the x direction (i.e., $y_{1i} = y_{2i}$) the above equations reduce to:

$$\alpha_{xxi} = 0 \quad (3.18)$$

$$\alpha_{yyi} = \frac{l_i(x_{1i}^2 + x_{2i}^2 + x_{1i}(x_{2i} - 3x_C) - 3x_{2i}x_C + 3x_C^2)}{3EI_{zi}} = \frac{l_i(l_i^2 + 3x_{1i}x_{2i} - 3x_{1i}x_C - 3x_{2i}x_C + 3x_C^2)}{3EI_{zi}} \quad (3.19)$$

$$\alpha_{\phi_z\phi_{zi}} = \frac{l_i}{EI_{zi}} \quad (3.20)$$

$$\alpha_{xyi} = \alpha_{yx_i} = -\frac{l_i(x_{1i} + x_{2i} - 2x_C)(y_{1i} - y_C)}{2EI_{zi}} = -\frac{l_i(x_{mi} - x_C)(y_{1i} - y_C)}{EI_{zi}} \quad (3.21)$$

$$\alpha_{x\phi_{zi}} = \alpha_{\phi_{zi}x_i} = \frac{l_i(y_{1i} - y_C)}{EI_{zi}} \quad (3.22)$$

$$\alpha_{y\phi_{zi}} = \alpha_{\phi_{zi}y_i} = -\frac{l_i(x_{mi} - x_C)}{EI_{zi}} \quad (3.23)$$

where,

$$x_{mi} = \frac{x_{1i} + x_{2i}}{2} \quad (3.24)$$

is the mid-point of the beam along the x direction. For this analysis, without loss of generality, the axis of symmetry is assumed to be in the y direction. With the symmetry assumption, it is implied that the two beams which are symmetrically located on either side of the axis of symmetry (for example, beams numbers 7 and 4 in Figure 3.12) are identical, i.e., have the same width. Adding the compliance matrix of the two x beams that are symmetrically placed about the axes of symmetry:

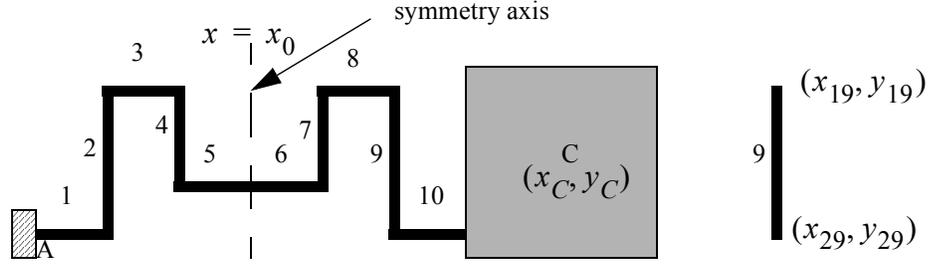


FIGURE 3.12. Example of a symmetric spring with the axes of symmetry along the y direction. The load point C , the anchor point A and the end points of beam number 9 are also shown.

$$\alpha_{\phi_z \phi_z i} + \alpha_{\phi_z \phi_z (2n-i)} = \frac{2l_i}{EI_{zi}} \quad (3.25)$$

$$\alpha_{xyi} + \alpha_{xy(2n-i)} = \alpha_{yxi} + \alpha_{yx(2n-i)} = -\frac{2l_i(x_0 - x_C)(y_{1i} - y_C)}{EI_{zi}} \quad (3.26)$$

$$\alpha_{x\phi_z i} + \alpha_{x\phi_z(2n-i)} = \alpha_{\phi_z xi} + \alpha_{\phi_z x(2n-i)} = \frac{2l_i(y_{1i} - y_C)}{EI_{zi}} \quad (3.27)$$

$$\alpha_{y\phi_z i} + \alpha_{y\phi_z(2n-i)} = \alpha_{\phi_z yi} + \alpha_{\phi_z y(2n-i)} = -\frac{2l_i(x_0 - x_C)}{EI_{zi}} \quad (3.28)$$

Similarly, simplifying the compliance equations for a y -beam (numbered j):

$$\alpha_{xxj} + \alpha_{xx(2n-j)} = \frac{l_j(y_{1j}^2 + y_{2j}^2 + y_{1j}(y_{2j} - 3y_C) - 3y_{j2}y_C + 3y_C^2)}{3EI_{zj}} \quad (3.29)$$

$$\alpha_{yyj} + \alpha_{yy(2n-j)} = 0 \quad (3.30)$$

$$\alpha_{\phi_z \phi_z j} + \alpha_{\phi_z \phi_z (2n-j)} = \frac{l_j}{EI_{zj}} \quad (3.31)$$

$$\alpha_{xyj} = \alpha_{yxj} = -\frac{l_j(x_{1j} - x_C)(y_{mi} - y_C)}{EI_{zj}} \quad (3.32)$$

$$\alpha_{x\phi_z j} = \alpha_{\phi_z xj} = \frac{l_j(y_{mj} - y_C)}{EI_{zj}} \quad (3.33)$$

$$\alpha_{y\phi_zj} = \alpha_{\phi_zyj} = -\frac{l_j(x_{1j} - x_C)}{EI_{zj}} \quad (3.34)$$

where,

$$y_{mj} = \frac{y_1 + y_2}{2} \quad (3.35)$$

is the mid-point of the j th beam, which is parallel to the y direction. Adding the compliance of two y beams which are placed symmetrically:

$$\alpha_{\phi_z\phi_zj} + \alpha_{\phi_z\phi_z(2n-j)} = \frac{2l_j}{EI_{zj}} \quad (3.36)$$

$$\alpha_{xyj} + \alpha_{xy(2n-j)} = \alpha_{yxj} + \alpha_{yx(2n-j)} = -\frac{2l_j((x_0 - x_C)(y_{mj} - y_C))}{EI_{zj}} \quad (3.37)$$

$$\alpha_{x\phi_zj} + \alpha_{y\phi_z(2n-j)} = \alpha_{\phi_zxj} + \alpha_{\phi_zx(2n-j)} = \frac{2l_j(y_{mj} - y_C)}{EI_{zj}} \quad (3.38)$$

$$\alpha_{y\phi_zj} + \alpha_{y\phi_z(2n-j)} = \alpha_{\phi_zyj} + \alpha_{\phi_zy(2n-j)} = -\frac{2l_j(x_0 - x_C)}{EI_{zj}} \quad (3.39)$$

The stiffness matrix being the inverse of the compliance matrix i.e.,

$$\begin{bmatrix} k_{xx} & k_{xy} & k_{x\phi_z} \\ k_{yx} & k_{yy} & k_{y\phi_z} \\ k_{\phi_zx} & k_{\phi_zy} & k_{\phi_z\phi_z} \end{bmatrix} = \left(\sum_{k=1}^{2n} \begin{bmatrix} \alpha_{xx} & \alpha_{xy} & \alpha_{x\phi_z} \\ \alpha_{yx} & \alpha_{yy} & \alpha_{y\phi_z} \\ \alpha_{\phi_zx} & \alpha_{\phi_zy} & \alpha_{\phi_z\phi_z} \end{bmatrix}_k \right)^{-1}$$

we can write:

$$k_{xy} = -\frac{1}{[\alpha]} \left(\left(\sum_{k=1}^{2n} \alpha_{yxk} \right) \left(\sum_{k=1}^{2n} \alpha_{\phi_z\phi_zk} \right) - \left(\sum_{k=1}^{2n} \alpha_{y\phi_zk} \right) \left(\sum_{k=1}^{2n} \alpha_{x\phi_zk} \right) \right) \quad (3.40)$$

where each k corresponds to a pair of symmetrically located beams.

Now each of the summation terms is expanded and contributions from the beams along x and the beams along y are separated. Using (3.26) and (3.37)

$$\sum_{k=1}^{2n} \alpha_{y_x k} = \sum_{xbeams} \frac{2l_i(x_0 - x_C)(y_{1i} - y_C)}{EI_{zi}} + \sum_{ybeams} \frac{2l_j(x_0 - x_C)(y_{mj} - y_C)}{EI_{zj}} \quad (3.41)$$

$$\sum_{k=1}^{2n} \alpha_{y_x k} = -(x_0 - x_C) \left(\sum_{xbeams} \frac{2l_i(y_{1i} - y_C)}{EI_{zi}} + \sum_{ybeams} \frac{2l_j(y_{mj} - y_C)}{EI_{zj}} \right)$$

Using (3.25) and (3.36)

$$\sum_{k=1}^{2n} \alpha_{\phi_z \phi_z k} = \sum_{xbeams} \frac{2l_i}{EI_{zi}} + \sum_{ybeams} \frac{2l_j}{EI_{zj}} = \sum_{k=1}^{2n} \frac{2l_k}{EI_{zk}} \quad (3.42)$$

Using (3.28) and (3.39)

$$\sum_{k=1}^{2n} \alpha_{y \phi_z k} = \sum_{xbeams} \frac{2l_i(x_0 - x_C)}{EI_{zi}} + \sum_{ybeams} \frac{2l_j(x_0 - x_C)}{EI_{zj}} \quad (3.43)$$

$$\sum_{k=1}^{2n} \alpha_{y \phi_z k} = -2(x_0 - x_C) \sum_{k=1}^{2n} \frac{l_k}{EI_{zk}} \quad (3.44)$$

Using (3.27) and (3.38)

$$\sum_{k=1}^{2n} \alpha_{x \phi_z k} = \sum_{xbeams} \frac{2l_i(y_{1i} - y_C)}{EI_{zi}} + \sum_{ybeams} \frac{2l_j(y_{mj} - y_C)}{EI_{zj}} \quad (3.45)$$

Evaluating the two terms of the cofactor in (3.40):

$$\left(\sum_{k=1}^{2n} \alpha_{y_x k} \right) \left(\sum_{k=1}^{2n} \alpha_{\phi_z \phi_z k} \right) = -(x_0 - x_C) \left(\sum_{xbeams} \frac{2l_i(y_{1i} - y_C)}{EI_{zi}} + \sum_{ybeams} \frac{2l_j(y_{mj} - y_C)}{EI_{zj}} \right) \left(\sum_{k=1}^{2n} \frac{2l_k}{EI_{zk}} \right) \quad (3.46)$$

$$\left(\sum_{k=1}^{2n} \alpha_{y \phi_z k} \right) \left(\sum_{k=1}^{2n} \alpha_{x \phi_z k} \right) = -(x_0 - x_C) \left(\sum_{k=1}^{2n} \frac{2l_k}{EI_{zk}} \right) \left(\sum_{xbeams} \frac{2l_i(y_{1i} - y_C)}{EI_{zi}} + \sum_{ybeams} \frac{2l_j(y_{mj} - y_C)}{EI_{zj}} \right)$$

(3.47)

Observing that the two cofactor terms are identical it is concluded that their difference is zero and therefore:

$$k_{xy} = 0 \quad (3.48)$$

Thus, it is shown that for any spring which has an axis of symmetry the in-plane elastic coupling is zero. It is important to note that there is no constraint on the widths of the beams in the above derivation (except that the beams which are symmetrically placed have identical widths, which is implied by the symmetry condition). It is also evident that placing a system composed of four such symmetric springs will have a net $K_{xy} = 0$. The above proof concludes the analysis of in-plane elastic cross-axis coupling. Referring to Figure 3.2, the in-plane stiffness matrix k_{ip} has been analyzed in detail so far. The out-of-plane stiffness matrix k_{op} provides much less scope for intuitive understanding. Therefore, the models and finite element verification for the stiffness matrix k_{op} are presented in Appendix A2 without extensive analysis. In the subsequent section, the discussion on elastic cross-axis coupling is extended to the remaining off-diagonal (shaded) elements of the stiffness matrix in Figure 3.2 by looking at coupling between the in-plane and the out-of-plane modes.

3.7 In-plane to Out-of-plane Elastic Cross-axis coupling

The in-plane to out-of-plane elastic coupling coefficients in Figure 3.2 (k_{xz} , $k_{x\phi_x}$, $k_{x\phi_y}$, k_{yz} , $k_{y\phi_x}$, $k_{y\phi_y}$, $k_{z\phi_z}$, $k_{\phi_x\phi_z}$ and $k_{\phi_y\phi_z}$ are the distinct elements assuming a symmetrical matrix) do not lend themselves to easy symbolic manipulation unlike the elements of the in-plane and out-of-plane stiffness matrices. They are, however, modeled in a much simpler manner for a single beam by considering the rotation of the *principle axes of elasticity* as is shown below. It has been shown earlier [39][51] that asymmetrical sidewall angles in single-layer microstructures lead to rotation of the principal axes of the

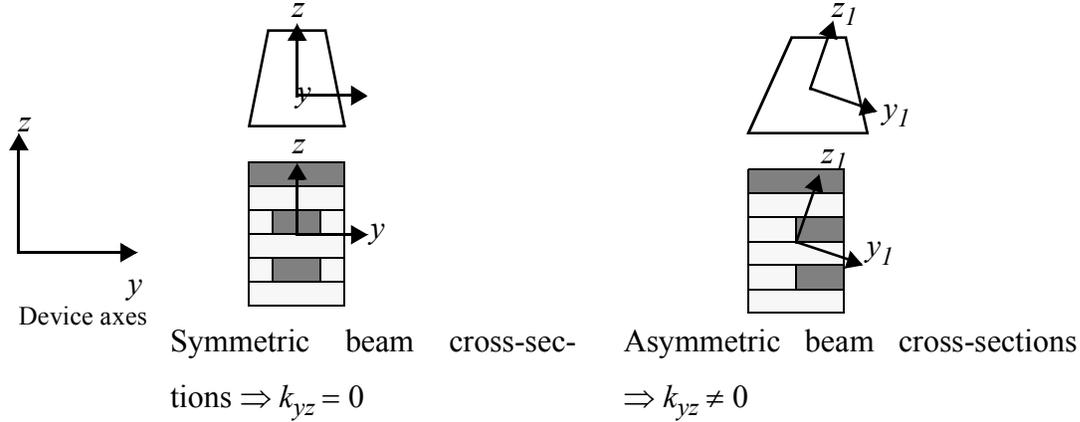


FIGURE 3.13. Rotation of beam principal axes due to asymmetrical cross-section: asymmetrical side-walls in the single-layer case and misaligned metal layers in the multi-layer case

beam cross-section and, therefore, causes x - z (in-plane to out-of-plane) coupling. The angle by which the axes are rotated is given by:

$$\phi = \frac{1}{2} \operatorname{atan} \left(\frac{2I_{yz}}{I_y - I_z} \right) \quad (3.49)$$

The above expression is only valid for homogeneous cross-section beams. In this section, a more general expression that is also valid for a multi-layer beam with misaligned layers is derived. This is followed by an approximate analysis for out-of-plane cross-axis coupling for springs with multiple beams.

3.7.1 Rotation of Principal Axes in Multi-layer Beam

The rotation of principal axes is illustrated in Figure 3.13(a) for the single-layer and multi-layer cases. FEA of CMOS-MEMS beams showed that x - z coupling can arise due to misaligned metal layers in a beam leading to an asymmetrical beam cross-section. In contrast with single-layer beams where the geometrical asymmetry causes cross-axis coupling, a material asymmetry, as shown in Figure 3.13, can also cause cross-axis coupling in CMOS-MEMS beams. The derivation of the rotation of principal axes in a multi-layer beam due to asymmetric cross-section is done in two steps: derivation of the location of the neutral axis i.e., the origin of the principal axes, followed by calculation of the orientation of the principal axes.

A multilayer cantilever structure with asymmetric cross-section is shown in Figure 3.14. Axial forces acting at the centroid of each layer lead to vertical and lateral bending moments. Each layer, i , has a thickness t_i , width w_i , area $A_i = w_i t_i$, coefficient of thermal expansion, α_i , and an effective Young's Modulus E_i . Let the coordinates of the center of the i th layer be (y_i, z_i) . The material properties for each layer are assumed to be uniform throughout the layer and independent of temperature.

In order to compute the location of the neutral axis (in this analysis the terms “neutral axis” and “centroid” are used interchangeably), a force F_x is applied at a point $X(y_o, z_o)$ along the axis of the beam and moments M_y and M_z about the y and z axes respectively so that a uniform axial strain ε is produced in the beam (i.e., the beam cross-section is displaced by a distance εL in the x direction, where L is the length of the beam). The total reaction force and the reaction moments produced about the point X are now calculated. The reaction force is given by the summation of the force over the entire beam cross-section as:

$$F_x = \sum_{i=1}^n A_i (E_i \varepsilon) = \varepsilon \left(\sum_{i=1}^n E_i A_i \right) \quad (3.50)$$

The reaction moment about the point X about the y and z axes are respectively given by:

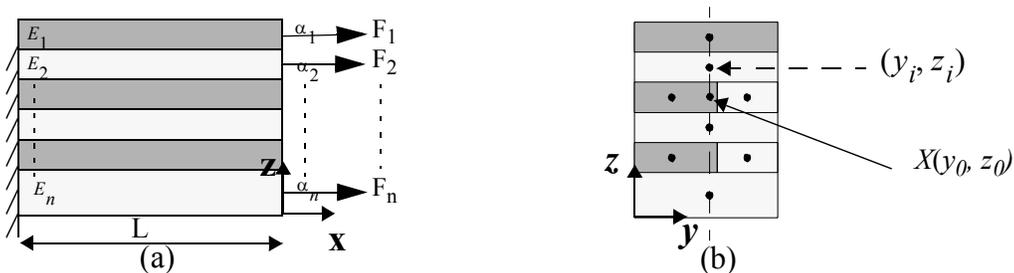


FIGURE 3.14. (a) Side view of a n -layer beam of length L (b) Cross-section of the beam with dots representing the axial forces acting out of plane

$$\begin{aligned}
M_y &= \sum_{i=1}^n F_i(z_i - z_o) = \varepsilon \sum_{i=1}^n E_i A_i (z_i - z_o) \text{ and} \\
M_z &= \sum_{i=1}^n F_i(y_i - y_o) = \varepsilon \sum_{i=1}^n E_i A_i (y_i - y_o)
\end{aligned} \tag{3.51}$$

The neutral axis (centroid) is defined as the point about which if forces and moments are applied to produce a uniform strain, then the moments are equal to zero. In other words, it is sufficient to apply only a force at the centroid in order to produce a purely uniform strain. Therefore, to compute the position of the neutral axis the moments computed above are equated to zero.

$$\sum_{i=1}^n E_i A_i (z_i - z_c) = 0 \text{ and } \sum_{i=1}^n E_i A_i (y_i - y_c) = 0 \tag{3.52}$$

where, (y_c, z_c) is the location of the neutral axis. Therefore,

$$\begin{aligned}
z_c &= \frac{\sum_{i=1}^n E_i A_i z_i}{\sum_{i=1}^n E_i A_i} \text{ and } y_c = \frac{\sum_{i=1}^n E_i A_i y_i}{\sum_{i=1}^n E_i A_i}
\end{aligned} \tag{3.53}$$

Now the case of pure bending about the y axis passing through the centroid is considered and the moments that are required to maintain this state are computed. Note that pure bending implies that there are no external forces acting on the beam. Let ρ_y be the radius of curvature of the beam. Using Euler-Bernoulli beam theory [39], the strain ε_i at the centroid of a layer located at (y_{ic}, z_{ic}) measured from the neutral axis, is given as:

$$\varepsilon_i = \frac{z_{ic}}{\rho_y} \tag{3.54}$$

The moment about the y axis needed to maintain beam i in this state along with a curvature ρ_y is:

$$M_{yi} = \frac{E_i I_{yi}}{\rho_y} + E_i A_i \varepsilon_i z_{ic} = \frac{E_i}{\rho_y} (I_{yi} + A_i z_{ic}^2) \quad (3.55)$$

where, $A_i = w_i t_i$ is the cross-section area of the beam and $I_{yi} = \frac{w_i t_i^3}{12}$ is the moment of inertia of the i th beam about its own centroid. Using the parallel-axes theorem, the above equation can be rewritten as:

$$M_{yi} = \frac{E_i I_{yic}}{\rho_y} \quad (3.56)$$

where, $I_{yic} = I_{yi} + A_i z_{ic}^2$ is the moment of inertia of the i th beam about the neutral axis of the composite beam. The bending moment about the z axis is given as:

$$M_{zi} = E_i A_i \varepsilon_i y_{ic} = \frac{E_i}{\rho_y} (A_i z_{ic} y_{ic}) \quad (3.57)$$

Note the absence of the $\frac{1}{\rho_z}$ term in the above equation since there is no bending about the z axis. Also, note that the term $A_i z_{ic} y_{ic}$ is simply the product of inertia I_{yzic} of the cross-section of the i th beam about the neutral axis of the composite beam. Therefore, the above equation can be rewritten as:

$$M_{zi} = \frac{E_i I_{yzic}}{\rho_y} \quad (3.58)$$

Summing up (3.56) and (3.58) over all the layers in the beam, the total moments required to produce pure bending about the y axis are obtained as:

$$M_{yc} = \sum_{i=1}^n \frac{E_i I_{yic}}{\rho_y} = \frac{\chi_y}{\rho_y} \quad (3.59)$$

$$M_{zc} = \sum_{i=1}^n \frac{E_i I_{yzic}}{\rho_y} = \frac{\chi_{yz}}{\rho_y} \quad (3.60)$$

Similarly, to produce pure bending with radius of curvature ρ_z about the z axis we need moments:

$$M_{yc} = \sum_{i=1}^n \frac{E_i I_{yzic}}{\rho_z} = \frac{\chi_{yz}}{\rho_z} \quad (3.61)$$

$$M_{zc} = \sum_{i=1}^n \frac{E_i I_{zic}}{\rho_z} = \frac{\chi_z}{\rho_z} \quad (3.62)$$

where, $\chi_y = \sum_{i=1}^n E_i I_{yic}$, $\chi_z = \sum_{i=1}^n E_i I_{zic}$ and $\chi_{yz} = \sum_{i=1}^n E_i I_{yzic}$. Note that the χ_{yz} term,

which is responsible for producing cross-moments, will be zero if the layers in the beam are in perfect symmetry about either the y or z axes. The set of axes about which bending does not produce a moment about an orthogonal axis are called principal axes. Let us consider bending with radius of curvature ρ about an axis y_1 which is rotated by an angle ϕ w.r.t. the y axis. The curvature can be split into orthogonal components about the y and z axes as:

$$\frac{1}{\rho_y} = \frac{1}{\rho} \cos(\phi) \quad \text{and} \quad \frac{1}{\rho_z} = \frac{1}{\rho} \sin(\phi) \quad (3.63)$$

The moments about the rotated axes y_1, z_1 can be written as:

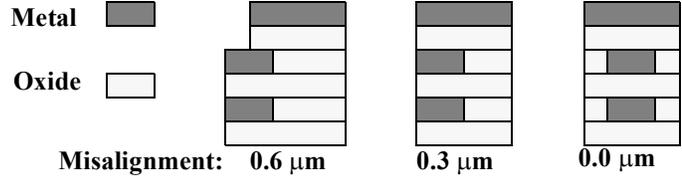
$$M_{y_1c} = M_{yc} \cos(-\phi) + M_{zc} \sin(-\phi) = \left(\frac{\chi_y}{\rho_y} + \frac{\chi_{yz}}{\rho_z} \right) \cos(\phi) - \left(\frac{\chi_{yz}}{\rho_y} + \frac{\chi_z}{\rho_z} \right) \sin(\phi) \quad (3.64)$$

$$M_{z_1c} = M_{yc} \sin(-\phi) + M_{zc} \cos(-\phi) = -\left(\frac{\chi_y}{\rho_y} + \frac{\chi_{yz}}{\rho_z} \right) \sin(\phi) + \left(\frac{\chi_{yz}}{\rho_y} + \frac{\chi_z}{\rho_z} \right) \cos(\phi) \quad (3.65)$$

Simplifying the above equations using (3.63):

$$M_{y_1c} = \frac{1}{\rho} (\chi_y \cos(\phi)^2 - \chi_z \sin(\phi)^2) \quad (3.66)$$

FIGURE 3.15. Cross-section of beams used for comparison of macromodel with FEA. The k_{yz} matches to 2% for 2 sets (50 μm X 3.0 μm and 100 μm X 2.1 μm) of 3 beams each.



$$M_{z_1 c} = \frac{1}{\rho} \left(\left(\frac{\chi_z - \chi_y}{2} \right) \sin(2\phi) + \chi_{yz} \cos(2\phi) \right) \quad (3.67)$$

If y_1, z_1 are principal axes of the beam, then for pure bending about y_1 the moment about z_1 will be zero, i.e., $M_{z_1 c} = 0$

$$\left(\frac{\chi_z - \chi_y}{2} \right) \sin(2\phi) + \chi_{yz} \cos(2\phi) = 0 \quad (3.68)$$

Therefore:

$$\tan(2\phi) = \frac{2\chi_{yz}}{\chi_y - \chi_z} \text{ i.e., } \phi = \frac{1}{2} \text{atan} \left(\frac{2\chi_{yz}}{\chi_y - \chi_z} \right) \quad (3.69)$$

It is seen that (3.69) is similar to (3.49) except that the flexural rigidities χ_{yz} , χ_y and χ_z are used in place of the moments of inertias I_{yz} , I_y and I_z respectively. Note that the stiffness matrix coefficients in the global reference frame can be calculated for a single beam by rotating the diagonal stiffness matrix.

A behavioral model of the multi-layer beam incorporating rotation of principal axes was implemented in the NODAS. k_{yz} was obtained from NODAS simulations and 3D finite element analysis (FEA) using Coventorware [54] by applying a displacement in z , and observing the reaction force in y . Two sets of three beams each, having length and width 100 X 2 μm and 50 X 3 μm respectively, were used in the simulations. The beams contained three metal layers and inter-metal oxide (Figure 3.15). The comparison between

k_{yz} values obtained from NODAS and FEA is shown in Table 3.4. For both the sets of

Table 3.4 Comparison of k_{yz} from NODAS and FEA

Beam dimension (l μm X w μm)	Misalignment (μm)	k_{yz} (NODAS) (N/m)	k_{yz} (FEA) (N/m)	Error (%)
100 X 2.1	0	0	0	0
100 X 2.1	0.3	0.0108	0.0108	-0.18
100 X 2.1	0.6	0.650	0.650	0.03
50 X 3	0	0	0	0
50 X 3	0.3	0.138	0.135	2.2
50 X 3	0.6	7.39	7.24	2.03

beams, for all values of misalignment the model matches the FEA results to about 2%.

In this section, in-plane to out-of-plane cross-axis coupling for a single beam was modeled by considering the rotation of the principal axes of stiffness. In the following section the concept of principal axes of stiffness is discussed further in the context of in-plane cross-axis coupling.

3.8 Geometrical Interpretation of Cross-axis Coupling

In this following discussion, a correlation between the elements of the in-plane stiffness matrix and the position and orientation of the principal axes of elasticity is established. Such a correlation allows us to compare two different spring designs with regard to their cross-axis coupling properties, independent of their absolute stiffness values. The benefits of such a comparison will become more apparent towards the end of this section.

Consider a proof-mass suspended by 4 springs as shown in Figure 3.16(a). The *center of elasticity* and the *principal axes of elasticity* of the system are shown respectively, as the point O_k and a set of orthogonal axes (x_k, y_k) such that: if O_k is displaced from its nominal position along either of the principal axes of elasticity, a reaction force is produced only along that axis. At the same time, no reaction moments are produced. In this section the in-plane system stiffness matrix is quantitatively linked to the origin and orientation of the principal axes of elasticity. There are two main cases of interest: (1) rotation and (2) displacement of the principal axes of elasticity. Figure 3.16 shows the nominal case and the

two other cases for a plate suspended by four springs. Nominally, when the 4 springs are identical, the center of elasticity is at the center of the plate and the principal axes of elasticity are aligned with the sides of the plate. As shown in Figure 3.16(b) and (c) respectively, rotation of the principal axes corresponds to a non-zero K_{xy} of the overall system and displacement of the principal axes corresponds to a non-zero $K_{\phi_z x}$ and $K_{\phi_z y}$. In terms of the system stiffness matrix, the two cases can be written as:

$$\begin{bmatrix} F_x \\ F_y \\ M_{\phi_z} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_{\phi_z} \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \phi_z \end{bmatrix} \quad (3.70)$$

where θ is the angle by which the principal axes have rotated

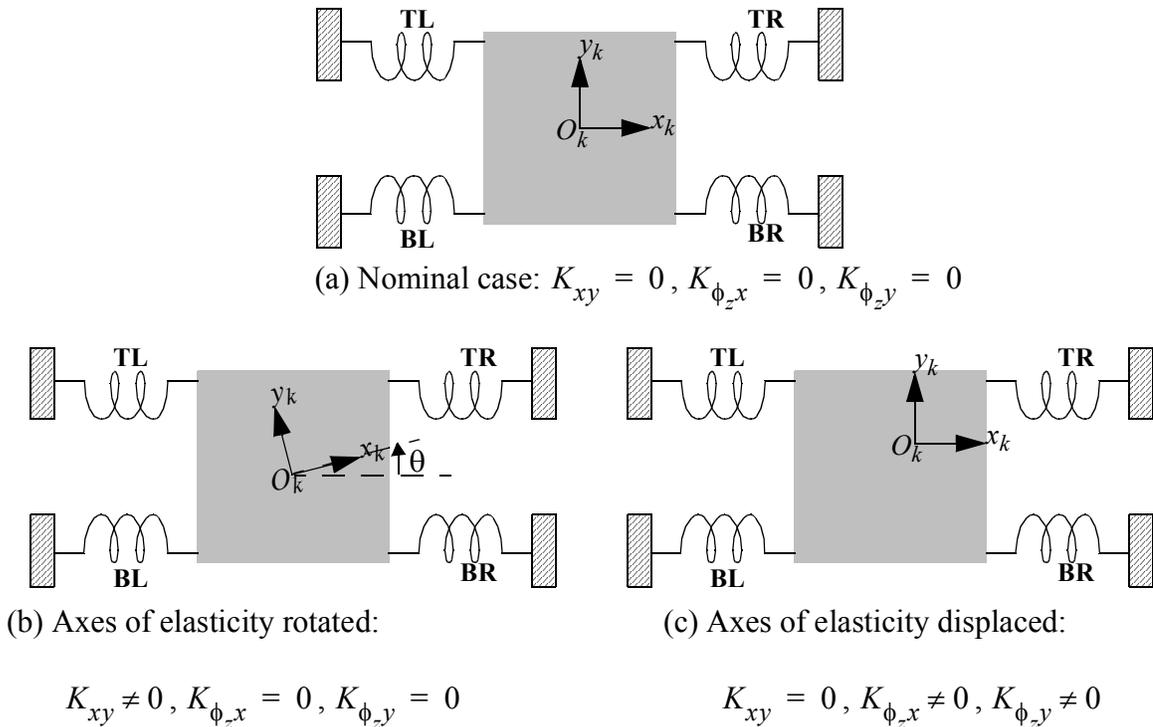


FIGURE 3.16. In-plane rotation and displacement of the principal axes of elasticity

$$\begin{bmatrix} F_x \\ F_y \\ M_{\phi_z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -Y_0 & X_0 & 1 \end{bmatrix} \begin{bmatrix} K_x & 0 & 0 \\ 0 & K_y & 0 \\ 0 & 0 & K_{\phi_z} \end{bmatrix} \begin{bmatrix} 1 & 0 & -Y_0 \\ 0 & 1 & X_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \phi_z \end{bmatrix} \quad (3.71)$$

where, (X_0, Y_0) is the position of the displaced set of principal axes. With an aim of correlating θ , and (X_0, Y_0) to the off-diagonal elements of the system stiffness matrix, the above equations are expanded to yield:

$$\begin{bmatrix} F_x \\ F_y \\ M_{\phi_z} \end{bmatrix} = \begin{bmatrix} K_x \cos(\theta)^2 + K_y \sin(\theta)^2 & \left(\frac{K_x - K_y}{2}\right) \sin(2\theta) & 0 \\ \left(\frac{K_x - K_y}{2}\right) \sin(2\theta) & K_y \cos(\theta)^2 + K_x \sin(\theta)^2 & 0 \\ 0 & 0 & K_{\phi_z} \end{bmatrix} \begin{bmatrix} x \\ y \\ \phi_z \end{bmatrix} \quad (3.72)$$

$$\begin{bmatrix} F_x \\ F_y \\ M_{\phi_z} \end{bmatrix} = \begin{bmatrix} K_x & 0 & -K_x Y_0 \\ 0 & K_y & K_y X_0 \\ -K_x Y_0 & K_y X_0 & K_{\phi_z} + K_x Y_0^2 + K_y X_0^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ \phi_z \end{bmatrix} \quad (3.73)$$

Note that the positions of the zero elements in the above stiffness matrices are consistent with the assertions in Figure 3.16.

Having shown that the rotation and displacement of the principal axes of elasticity correspond to system stiffness matrix elements K_{xy} and $K_{x\phi_z}, K_{y\phi_z}$ respectively, the system stiffness matrix is now derived in terms of the individual spring stiffness matrix which were derived in Section 3.3. Figure 3.17 shows how the signs of the off-diagonal elements change as the spring (crab-leg shown only to specify orientation, can be any spring topology in general) is mirrored in the x and y directions. The signs can be accounted for by following the simple rule: If a spring is mirrored about the x (y) direction, then the elements in the corresponding row and column of the stiffness matrix are negated. Note that, while implementing this rule, the diagonal elements change are negated twice and, therefore,

they do not change sign at all and remain positive. The overall stiffness matrix of a plate suspended by the 4 springs shown in Figure 3.17 can be written as:

$$K = k_{TL} + k_{TR} + k_{BR} + k_{BL} = \begin{bmatrix} 4k_{xx} & 0 & 0 \\ 0 & 4k_{yy} & 0 \\ 0 & 0 & 4k_{\phi_z\phi_z} \end{bmatrix} \quad (3.74)$$

Note that the off-diagonal terms cancel out and the diagonal terms add up. Following this discussion on the principal axes of stiffness, the practical implications of the insight into elastic coupling are explored in the next section.

3.9 Manufacturing Variations and Elastic Cross-axis Coupling

Manufacturing variations are inevitable in any fabrication process. In particular, mismatch between the widths of beams in springs can lead to elastic cross-axis coupling i.e., rotation or translation of the principal axes of stiffness as discussed in the previous section. In IC fabrication processes, geometrical as well as material properties are known to vary by as much as 10% across a wafer. This variation is generally composed of two parts: a gradual trend across the wafer and a localized random component. When we zoom into a single device, the gradual trends appear to be almost linear across the area of the device. While a linear trend may suffice to model the statistical mean of a parameter across a device, random variations may lead to significant deviation from the statistical mean in individual springs. Therefore, in the following analysis, both, linear variations as well as localized random variation of beam widths of individual springs are considered.

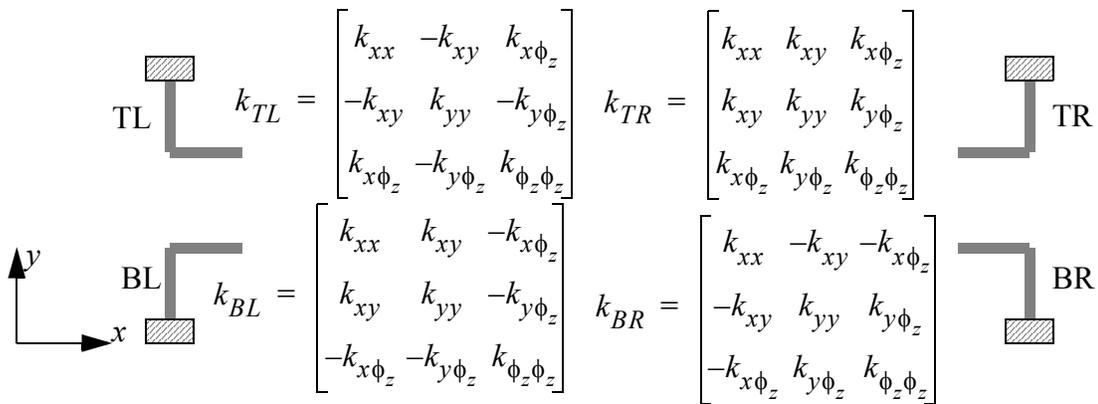


FIGURE 3.17. Signs of off-diagonal elements in the in-plane stiffness matrix

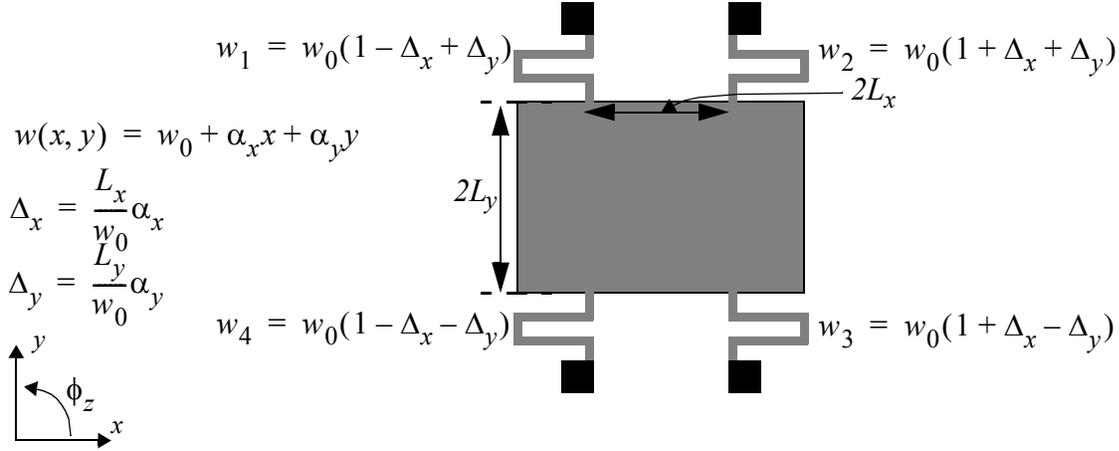


FIGURE 3.18. Linear variation in beam-widths across a wafer, mapped onto the springs of a single device with a plate suspended by four springs

A linear variation of beam widths across a device is depicted in Figure 3.18. The width of a hypothetical beam that would have been located at the center of the device is assumed to be w_0 . The gradients along the x and y directions are assumed to be α_x and α_y respectively. Then, the beam width variations are as shown in Figure 3.18. It is assumed that the distance between the beams forming a single spring is small enough that the beams of the same spring are assumed to have the same width. Since the elements of the stiffness matrix are proportional to w^3 (w being the width of the beams) to the first order, for a fractional change Δ in the width, the corresponding fractional change in the stiffness is 3Δ . The in-plane portion of the stiffness matrix can be written using (3.74) and Figure 3.17 as:

$$K = 3 \begin{bmatrix} (4/3)k_{xx} & k_{xy}(\Delta_1 + \Delta_3 - \Delta_2 - \Delta_4) & k_{x\phi_z}(\Delta_1 + \Delta_2 - \Delta_3 - \Delta_4) \\ k_{xy}(\Delta_1 + \Delta_3 - \Delta_2 - \Delta_4) & (4/3)k_{yy} & k_{y\phi_z}(\Delta_1 + \Delta_4 - \Delta_2 - \Delta_3) \\ k_{x\phi_z}(\Delta_1 + \Delta_2 - \Delta_3 - \Delta_4) & k_{y\phi_z}(\Delta_1 + \Delta_4 - \Delta_2 - \Delta_3) & (4/3)k_{\phi_z\phi_z} \end{bmatrix} \quad (3.75)$$

Substituting the values for Δ_1 , Δ_2 , Δ_3 and Δ_4 from Figure 3.18,

$$K = \begin{bmatrix} 4k_{xx} & 0 & k_{x\phi_z}(12\Delta_y) \\ 0 & 4k_{yy} & k_{y\phi_z}(12\Delta_x) \\ k_{x\phi_z}(12\Delta_y) & k_{y\phi_z}(12\Delta_x) & 4k_{\phi_z\phi_z} \end{bmatrix} \quad (3.76)$$

Note that a linear gradient in the beam widths along any direction, will not result in a non-zero K_{xy} in the system matrix. This implies that there is no first order elastic coupling between the x and the y modes arising due to a linear gradient in the beam widths. There will, however, be a second order elastic coupling between the x and the y modes through the $K_{x\phi_z}$ and the $K_{y\phi_z}$ in (3.76).

Three cases of beam width variation are now examined and the corresponding changes in the stiffness matrix as well as the location and orientation of the principal axes are observed. The system stiffness matrix can, in general, be written as:

$$K = k_{TL}(1 - 3\Delta_1) + k_{TR}(1 - 3\Delta_2) + k_{BR}(1 + 3\Delta_3) + k_{BL}(1 + 3\Delta_4) \quad (3.77)$$

The first case is when the $\alpha_x = 0$, i.e., widths of the top two springs i.e., springs TL and TR in Figure 3.17, are enhanced by a fraction Δ_y and the widths of the bottom two springs, BL and BR are diminished by a fraction Δ_y . The system stiffness matrix obtained using (3.77) is:

$$K = \begin{bmatrix} 4k_{xx} & 0 & k_{x\phi_z}(12\Delta_y) \\ 0 & 4k_{yy} & 0 \\ k_{x\phi_z}(12\Delta_y) & 0 & 4k_{\phi_z\phi_z} \end{bmatrix} \quad (3.78)$$

Comparing the above stiffness matrix with (3.74) it is seen that they differ only in the $K_{x\phi_z}$ and the K_{ϕ_zx} locations. Geometrically, this is equivalent to movement of the principal axes along the y axis. This can be explained by the fact that the top springs TL and TR become “weaker” and the bottom springs BL and BR become “stronger” thereby pushing the principal axes downwards. Note also that the K_{xy} terms are zero in (3.78) and this cor-

responds to the principal axes not rotating. In terms of the displacement of the suspended plate, motion along the x direction couples to rotation about the z axis and vice-versa, but there is no coupling between x and y motion.

The second case under consideration is when $\alpha_y = 0$, i.e., the widths of the two springs on the left i.e, springs TL and BL in Figure 3.17 are diminished by a fraction Δ_x and the widths of the two springs on the right, TR and BR are enhanced by a fraction Δ_x . The overall stiffness matrix can be written using (3.77) as:

$$K = \begin{bmatrix} 4k_{xx} & 0 & 0 \\ 0 & 4k_{yy} & k_{y\phi_z}(12\Delta_x) \\ 0 & k_{y\phi_z}(12\Delta_x) & 4k_{\phi_z\phi_z} \end{bmatrix} \quad (3.79)$$

Comparing the above stiffness matrix with (3.74) the difference is only in the $K_{y\phi_z}$ and the $K_{\phi_z y}$ locations. This is similar to the first case considered above, except that now movement of the principal axes is along the x axis. The left springs TL and BL become “weaker” and the right springs TR and BR become “stronger”, pushing the principal axes to the right. Again, note that the K_{xy} terms are zero in (3.79) and this corresponds to the principal axes not rotating. Motion of the suspended plate in the y couple to rotation about the z axis and vice-versa but there is no coupling between y and x .

The third case is when the widths of the springs on the 135° diagonal i.e, springs TL and BR in Figure 3.17 are enhanced by a fraction Δ and the widths of the two springs on the 45° diagonal, TR and BL are diminished by a fraction Δ . Note that such a variation will not result from a linear gradient. The overall stiffness matrix can be written using (3.77) as:

$$K = \begin{bmatrix} 4k_{xx} & k_{xy}(12\Delta) & 0 \\ k_{xy}(12\Delta) & 4k_{yy} & 0 \\ 0 & 0 & 4k_{\phi_z\phi_z} \end{bmatrix} \quad (3.80)$$

Comparing the above stiffness matrix with (3.74) the difference is seen to be only in the K_{xy} and the K_{yx} locations. This is in contrast to the previous two cases where the K_{xy} terms were zero. In this case there is no movement of the principal axes but there is a rotation. The direction of rotation of the principal axes depends on which of the two orthogonal spring constants i.e., $4k_{xx}$ and $4k_{yy}$ is larger. In this case there is direct motion coupling between the x and y directions, but there is no coupling to the rotational direction.

There is an interesting duality between the position and orientation of the principal axes and the actual displacements of the suspended plate. For example, when the principal axes are not displaced but only rotated, there is coupling only between the translational modes x and y . Similarly, when the principal axes are displaced but not rotated, there is coupling between a translational mode (x or y) mode and the rotational mode, but no direct coupling between the translational modes themselves. Note, however, that there is a second-order coupling between the translational motions, if the principal axes are displaced in an arbitrary direction which is neither along x nor y .

In general, in a real device, one would expect a small uncertainty in the position and orientation of the principal axes of elasticity due to manufacturing variations. To complete the discussion on the in-plane cross-axis coupling, simple expressions are now derived for motion coupling between the translational modes for two main cases of interest from the stiffness matrix point of view:

1. Rotation of principal axes i.e., $K_{xy} \neq 0$, $K_{x\phi_z} = K_{\phi_z x} = 0$ and $K_{y\phi_z} = K_{\phi_z y} = 0$:

$$\frac{y}{x} = \frac{K_{xy}}{K_{yy}} \text{ and } \frac{x}{y} = \frac{K_{xy}}{K_{xx}} \quad (3.81)$$

2. Displacement of the principal axes, but no rotation i.e., $K_{xy} = 0$, $(K_{x\phi_z} = K_{\phi_z x}) \neq 0$ and $(K_{y\phi_z} = K_{\phi_z y}) \neq 0$:

$$\frac{y}{x} = \frac{K_{x\phi_z} K_{y\phi_z}}{K_{yy} K_{\phi_z \phi_z}} \text{ and } \frac{x}{y} = \frac{K_{x\phi_z} K_{y\phi_z}}{K_{xx} K_{\phi_z \phi_z}} \quad (3.82)$$

Note that in (3.81) the coupling depends only on one off-diagonal term K_{xy} and is, therefore, directly proportional to the fractional width mismatch Δ . However, in (3.82), the coupling depends on two off-diagonal terms $K_{x\phi_z}$ and $K_{y\phi_z}$ and is therefore proportional to the Δ^2 . Δ can be considered either as the normalized standard deviation of the widths of the four springs for a statistical analysis or, if the width of only one spring is changed, as the fractional width mismatch of the single spring. Comparing (3.81) and (3.82) with (3.72) and (3.73) respectively, it is seen that the coupling between the two translational modes is directly proportional to the angle by which the axes are rotated or the product of the displacements of the principal axes respectively. The high-level parameters (θ , X_0 and Y_0) defining the position and orientation of the principal axes can thus be effectively used by designers to set manufacturing tolerances for system design which are independent of the spring topology being used.

The above analysis connects the system stiffness matrix and the displacement and rotation of the principal axes of stiffness of rotation to the actual values of mode-coupling. In this section, the system stiffness matrix, in particular the off-diagonal elements, were expressed in terms of geometrical asymmetries arising due to manufacturing variations. In the previous sections, in-plane, out-of-plane and in-plane to out-of-plane cross-axis coupling were analyzed in detail. The analysis lends more insight into spring design issues and also highlights the differences in the types of cross-axis coupling. This section concludes the entire discussion on modeling of elastic cross-axis coupling.

The next section focuses on another important practical elastic modeling issue: temperature dependent curvature of multi-layer beams. The derivation of the neutral axis of the multi-layer beams as presented in Section 3.7 is an integral part of the models for vertical and lateral curvature.

3.10 Curl Modeling

Vertical stress gradients in a cantilever beam arising due to the multi-layer nature of the CMOS microstructures have been analyzed previously using thermal multimorph theory [56][57][58]. The technique outlined in [58] is useful for calculating the vertical cur-

vature of multi-layer cantilever beams. Building upon the basic thermal multimorph theory, models for the following additional effects are developed in this section:

1. Arbitrary boundary conditions applied to beam ends
2. Lateral curling due to misaligned metal layers
3. Curling of a number of interconnected beams

The model formulated is compatible with existing linear behavioral models for beams in NODAS [7]. Using this macromodel the internal stresses in the different layers in the CMOS microstructure can be calculated. The stress in the polysilicon layer can be used to incorporate piezoresistive effects. The first sub-section details the model development, the second presents verification results using FEA and the final sub-section describes the measured results.

3.10.1 Extension of Multimorph Analysis

A multilayer cantilever structure is shown in Figure 3.14. Axial forces acting at the centroid of each layer lead to vertical and lateral bending moments. Each layer, i , has a thickness t_i , width w_i , area $A_i = w_i t_i$, coefficient of thermal expansion, α_i , and an effective Young's Modulus E_i . The out-of-plane curling due to residual stress gradient in the beam produces a tip deflection δ . The material properties for each layer are assumed to be uniform throughout the layer and independent of temperature.

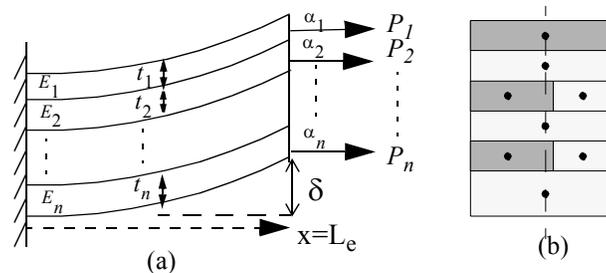


FIGURE 3.19. (a) Model of a CMOS cantilever beam composed of metal, dielectric and polysilicon layers (b) Cross-section of an asymmetric multi-layer beam with dots representing the axial forces acting out of plane. Since the forces are asymmetrically located there is a resultant lateral bending moment in addition to the vertical bending moment

Let P_i represent the force in the i th layer due to the interfacial forces between adjacent layers. Since the forces P_i are produced by action-reaction pairs, they sum to zero. Let M_{yi} represent the moment about the y -axis in the i th layer produced by the interfacial forces.

$$\sum_{i=1}^n P_i = 0; \left(\sum_{i=1}^n M_{yi} \right) = -(Z^T P) \quad (3.83)$$

where, P denotes the force column vector and Z is the moment arm vector measured from the neutral axis of the composite beam.

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_n \end{bmatrix}, \quad Z = \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_n \end{bmatrix} = \begin{bmatrix} \frac{t_1}{2} - z_0 \\ t_1 + \frac{t_2}{2} - z_0 \\ \dots \\ \sum_{i=1}^{n-1} t_i + \frac{t_n}{2} - z_0 \end{bmatrix} \quad (3.84)$$

where, z_0 is the vertical distance of the neutral axis of bending from the top of the beam and z_i is the distance of the centroid of each layer from the neutral axis. Thickness of the beam is assumed to be much less than the radius of curvature (ρ), and the radius of curvature can be assumed to be the same for each layer.

$$\frac{1}{\rho} = \frac{M_{yi}}{E_i I_{yi}} \text{ or } M_{yi} = \frac{E_i I_{yi}}{\rho} \text{ where, } I_{yi} = \frac{w_i t_i^3}{12} \quad (3.85)$$

where I_{yi} is the moment of inertia of the i th layer having width w_i taken about the principal axis of the layer parallel to the y -axis. Let T_0 represent the temperature of the beam when it is flat [58]. Equating axial strains at the interfaces between layers due to temperature change, $\Delta T = T - T_0$

$$\frac{P_{i+1}}{E_{i+1}A_{i+1}} - \frac{P_i}{E_iA_i} + \Delta T(\alpha_{i+1} - \alpha_i) - \frac{z_i - z_{i+1}}{\rho} = 0 \quad (3.86)$$

Observing the uniformity of the above equation in the subscript i :

$$\frac{P_i}{E_iA_i} + \Delta T\alpha_i - \frac{z_i}{\rho} = C \quad (3.87)$$

where C is a uniform axial strain for all layers. Multiplying throughout by $E_iA_iz_i$,

$$P_iz_i + E_iA_iz_i\Delta T\alpha_i - \frac{E_iA_iz_i^2}{\rho} = C(E_iA_iz_i) \quad (3.88)$$

Summing up over all layers and using (3.83) and (3.85)

$$-\sum_i \frac{E_i}{\rho}(I_{yi} + A_iz_i^2) + \sum_i E_iA_iz_i\Delta T\alpha_i = C\sum_i E_iA_iz_i \quad (3.89)$$

Noting that the first term on the left hand side contains the parallel axis theorem for computing moments and that the right hand side reduces to zero, total bending moment acting on the composite beam is obtained as:

$$M_y = \sum_i (z_i w_i t_i E_i \Delta T \alpha_i) \quad (3.90)$$

A similar analysis for the lateral moment yields:

$$M_z = \sum_i (y_i w_i t_i E_i \Delta T \alpha_i) \quad (3.91)$$

The total axial thermally induced force is given by:

$$F_x = \sum_i (w_i t_i E_i \Delta T \alpha_i) \quad (3.92)$$

It should be noted that the thermally induced forces and moments do not depend on the length of the beam and can be computed by using only the beam cross-section. This further implies that the curvature of a beam is independent of the length of the beam. The tip

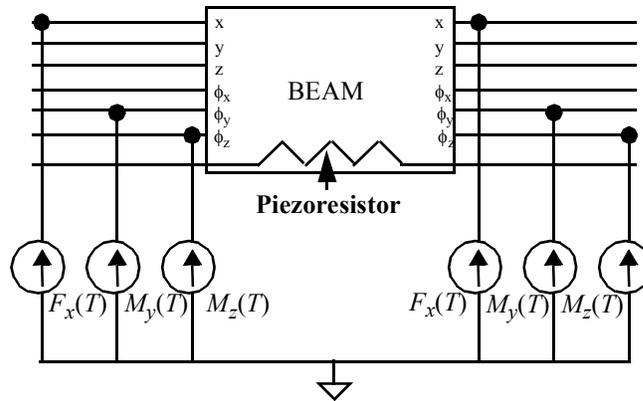


FIGURE 3.20. Norton equivalent of a beam macromodel with thermally induced lumped force and moment sources and an embedded piezoresistor. The beam has three translational pins, three rotational pins and one electrical pin at each port.

deflection, however, is quadratically dependent on the length of the beam. Since linear beam theory has been assumed, by superposition, any additional force or moments can be applied at the ends of the beam. In other words, independent arbitrary boundary conditions can be applied in addition to the thermally induced forces and moments. Applying the force from (3.92) and the moments from (3.90) and (3.91), at the ends of the beam will result in exactly the same displacement and rotation of the composite beam as that produced in each individual beam by the forces and moments produced by the interfacial forces between layers and the thermal stresses. Those familiar with Thevenin and Norton equivalent models in electrical circuits can identify with the analogy of a Norton equivalent model. The Norton equivalent model of the macromodel is shown in Figure 3.20. The beam itself behaves analogous to a resistive element, the force (current) transmitted through the beam (resistor) producing a linearly dependent displacement (voltage) across the two ends of the beam (resistor). The thermally induced forces and moments are simply added as sources to ground, as if current sources are placed at both ends of the resistor.

In order to model the piezoresistive effect the strain in the polysilicon layer along the length of the beam needs to be calculated. Using Euler-Bernoulli beam theory, it can be shown that the total change in resistance depends only on the average strain along the length of the beam, which in turn is dependent only on the axial strain of the beam and the curvature at the center of the beam. The average longitudinal strain in a layer which is at distance of (c_y, c_z) from the centroid of the composite beam is:

$$\varepsilon_l = \left(\frac{(x_1 - x_2)}{L} + \frac{(\phi_{y1} - \phi_{y2})c_z}{L} + \frac{(\phi_{z1} - \phi_{z2})c_y}{L} \right) \quad (3.93)$$

where, $(x_1, \phi_{y1}, \phi_{z1})$ and $(x_2, \phi_{y2}, \phi_{z2})$ are the axial positions, vertical rotation and lateral rotation at the two terminals of the beam and L is the length of the beam. The resistivity change is related to the longitudinal strain in the layer through the piezoresistive coefficient, π_l as:

$$\frac{\Delta\rho}{\rho} = E_l \pi_l \varepsilon_l \quad (3.94)$$

Transverse piezoresistance of polysilicon is much lower than longitudinal and has been omitted from the above equation.

The piezoresistance model illustrates the usefulness and flexibility of the proposed macromodel for curvature multi-layer structures. In the next section, the macromodel is verified by comparison with FEA.

3.10.2 FEA Verification

In order to verify the three claims of the macromodel two different finite element analyses were run. First FEA is done on single cantilever beams in order to verify the accuracy of the macromodel in both vertical and lateral curvature. Second, a suspended plate structure with four U-springs is numerically analyzed to verify that the macromodel accurately captures arbitrary boundary conditions as well as a number of interconnected beams.

As part of the first set of analyses, thermomechanical FEA was done on a CMOS beam with 3 metal layers. The width of the Metal3 layer is set to a commonly used value of $2.1\mu m$. Metal2 and Metal1 layers widths are set to $1.8\mu m$. Figure 3.21(b) shows the comparison of tip deflections predicted by the macromodel and the FEA in the lateral (y) and vertical (z) directions. The bottom two metal layers were deliberately misaligned as shown in Figure 3.21 (a) in order to produce lateral curling with temperature change. As predicted by the theory, the deflection is linear with temperature and the difference between the macromodel and the FEA is less than 3% for all temperatures.

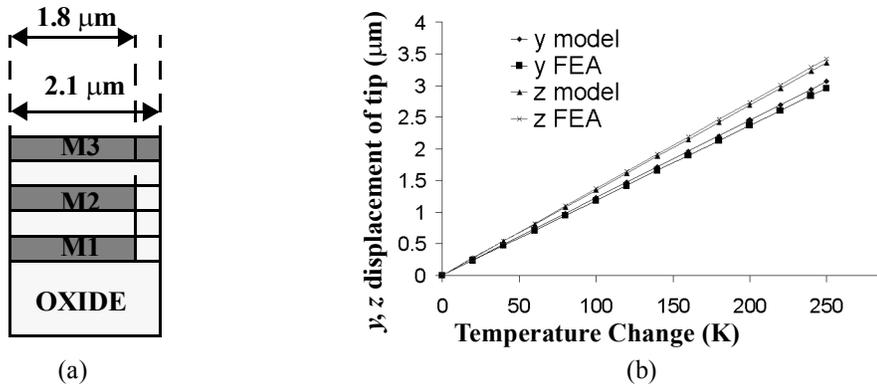


FIGURE 3.21. (a) Cross-section of beams. Metal3 is 2.1 μm wide and Metal2 and Metal1 layers are 1.8 μm wide and are offset by -0.15 μm from the center of the beam (b) Comparison of behavioral curl model with FEA for beam of length 100 μm . Difference is less than 3% for all temperatures.

In order to verify the generality of the model, seven different beam compositions were analyzed. Table 3.5 summarizes the comparison. The type of the beam refers to the composition with numbers denoting presence of the corresponding metal layer and P indicating presence of polysilicon layer. The beams were 100 μm long, the metal layers are 2.1 μm and the polysilicon is 1.2 μm wide. Since a linear relationship between temperature change and tip deflection was expected, the temperature change was set to only one value: 100 K. The tip deflection values are given in μm . The model values match the FEA very

Table 3.5 Vertical deflection (in μm) of tip in 100 μm : Macromodel vs. FEA

Type	321P	32P	31P	3P	21P	2P	1P
FEA	1.37	2.00	0.84	1.47	2.86	3.07	8.19
Model	1.36	1.98	0.82	1.44	2.81	3.03	8.06

closely, probably limited by numerical precision of the finite element analyses or minor second-order non-linear effects.

In order to verify the capability of the model to handle arbitrary boundary conditions, 3D thermo-mechanical FEA of a simplified accelerometer structure was done using Coventorware. The structure consists of a single plate suspended by four serpentine springs. The serpentine spring was deliberately chosen to see the effects of a large number of beams. Since there is no existing thermal curling macromodel for a plate, the plate itself is

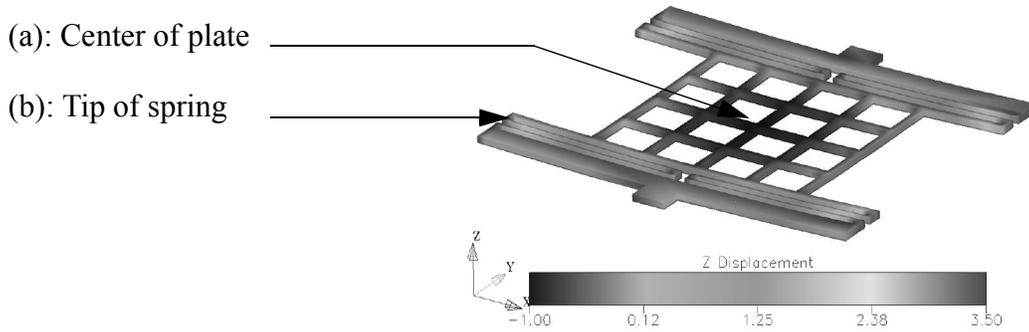


FIGURE 3.22. Temperature-induced curling of a simplified accelerometer structure obtained from 3D FEA. The vertical deflections at points (a) and (b) are compared with results from behavioral simulation using the macromodel in Table 3.6.

composed of a number of beam elements. The curled shape of the accelerometer structure due to temperature change is seen in Figure 3.22. The comparison of vertical deflection from FEA with the macromodel is shown in Table 3.6. The two deflections are in good

Table 3.6 Vertical deflection (in μm) for simplified accelerometer

Location	Macromodel	FEA
(a) Center of plate	-0.9	-0.95
(b) Tip of spring	2.9	3.0

agreement for both the points. Thus, it is seen that the macromodel with force and moment sources applied at the two ends provides a convenient technique to simulate temperature dependent curvature in complex suspended structures with a number of beams, which are not necessarily cantilevers.

3.10.3 Measurements

Measurements to characterize vertical and lateral curl in CMOS beams were made on beams with integrated heaters. An SEM of the test structure is shown in Figure 3.23. The structure has 3 main parts. A heated base with an integrated polysilicon heater embedded in the structure. A meandering spring thermally isolates the heated base from the substrate. The test beams are attached to the heated base. Ohmic heating is used to increase the temperature of the base. The temperature of the base and the beams is the same as thermal losses to the substrate are small due to the small device area. The device temperature

was extracted from the resistance change in the polysilicon heater. The temperature characteristic of the polysilicon heater was characterized separately by measuring the resistance of the polysilicon heater while the device was placed in a temperature-controlled oven. The device temperature was measured using the temperature characteristic of threshold voltage of the N-well-substrate diode. Measurements with an infrared microscope were made to confirm the uniform temperature distribution.

A 10 Hz triangular heating pulse was applied to the heater. Beam deflections in the lateral direction and the beam curvature were measured using the MIT microvision system [59]. The temperature distribution is expected to reach its equilibrium value at every measurement as the thermal time constant of the structure is 6 ms. The beam out-of-plane curl measurements were confirmed by static interferometry images of the structure. The change in device shape with temperature is shown in Figure 3.24.

Comparison of the measured deflection vs. temperature with the macromodel is shown in Figure 3.25. A measured $0.15 \mu\text{m}$ overetch was incorporated in the deflection computation. The measured and the modeled deflections match to within 15% for large deflections. Possible sources of error include temperature calibration and microvision resolution.

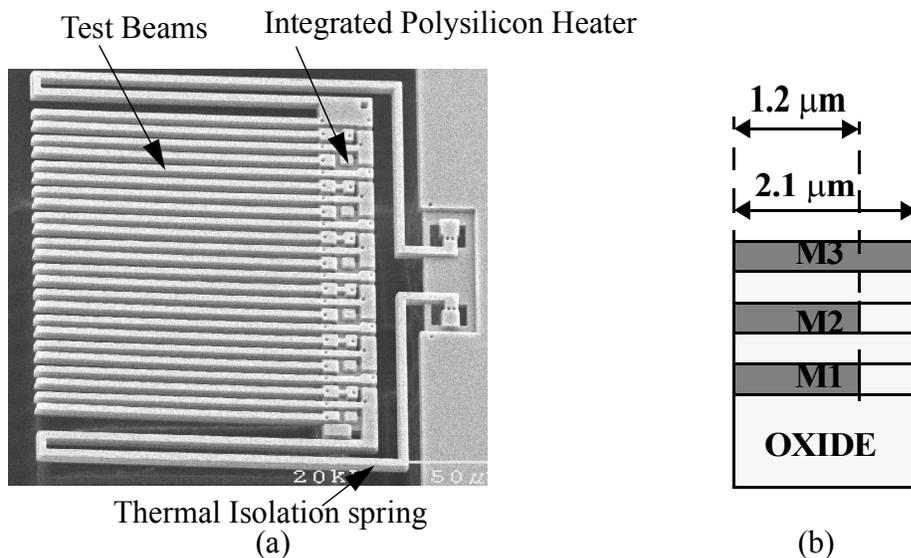


FIGURE 3.23. (a) SEM of the test structure used to characterize beam curling with temperature. It consists of alternating misaligned and symmetric beams (b) Cross-section of the measured beams (misaligned)

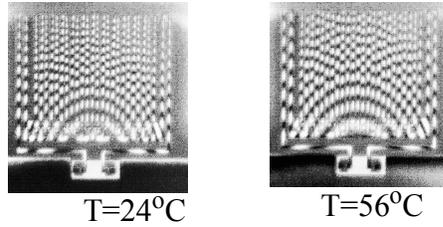


FIGURE 3.24. Interferometric images of the out-of-plane curl of the test structure at 24°C and 56°C. One fringe length corresponds to 245 nm displacement in the vertical direction

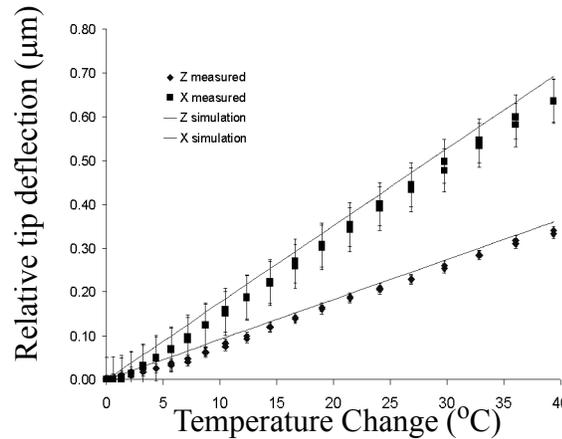


FIGURE 3.25. Comparison of relative tip deflection from measurements and macromodel showing a match to within 15%. 100 nm and 20 nm error bars are shown for the z and x deflections respectively.

Thus, the proposed macromodel for vertical and lateral curling of multi-layer beams has been verified through FEA as well as measurement results.

3.11 Summary

A comprehensive analysis of elastic cross-axis coupling was presented at the individual spring level as well as the system level with multiple springs. Equations for cross-axis stiffness constants were derived for crab-leg, u-shaped and serpentine springs. It was proved that first order cross-axis coupling between the two in-plane translational modes can be eliminated by using symmetric springs. Beam cross-section asymmetry leading to cross-axis coupling between the in-plane and the out-of-plane modes was also modeled. A geometrical interpretation of cross-axis coupling relating the stiffness constants to translational and rotational movement of the principal axes of elasticity was presented. This

interpretation allows us to specify cross-axis coupling independent of the absolute spring stiffnesses and can also be used to represent inertial and viscous coupling.

A macromodel suitable for schematic-based simulation of thermally induced lateral and vertical curling in multi-layer CMOS sensors was derived using thermal multimorph theory. By simulating a schematic with multiple beams, the curling characteristic of CMOS sensors can be estimated at the design stage instead of post-fabrication characterization. Furthermore, effects of manufacturing variation such as mask misalignment on sensor performance can now be studied in an integrated manner.

In this chapter elastic cross-axis coupling models have been derived for specific spring topologies. In the next chapter a more general method for computing the stiffness matrix, which can be applied to any single-chain configuration of beams, is presented.

Chapter 4. Reduced Order Models: From Beams to Springs

4.1 Introduction

In a typical accelerometer schematic consisting of a plate suspended by four U-springs, more than 50% of the nodes in the simulation matrix can be traced to the *beam* elements used. In such an accelerometer, as well as other inertial sensors, the *beam* elements operate in a completely linear regime (because of the extremely small i.e., < 10 nm displacements). Exceptions to this linear behavior are the *beams* associated with the large driven mode (few μm) in a microgyroscope. The large number of spring beams in an inertial sensor schematic makes springs prime candidates for reduced order models. This also has implications for simulation-based design synthesis from high-level performance specifications, which has been demonstrated for accelerometers [60]. During synthesis, the accelerometer schematic is simulated several thousand times with varying geometrical parameters to find an optimal design. The main output of the simulations is the overall performance of a design. The displacements at the terminals of a particular beam element are not of any interest during the synthesis process. Similarly, a single mass element bearing equivalent inertial properties as the overall *plate* system in the inertial sensor, is more efficient in terms of simulation times. Therefore, higher-level behavioral simulations using functional elements such as *spring*, *mass*, *damper* and electrostatic *sensors* and *actuators* are better suited for embedding within the synthesis loop.

The previous chapter described in detail the models for cross-axis coupling in particular spring topologies such as crab-leg, serpentine and U-springs. In this chapter, the focus is shifted to more general spring topologies and to extract functional parameters from geometrical parameters for entire systems. The main goal of this chapter is to develop a methodology for rapid translation from a circuit-level *beam-plate* schematic to a functional-level *spring-mass* schematic. This goal is achieved by combining an efficient spring stiffness computation procedure with an algorithm to automatically generate a *spring-mass* netlist from a *beam-plate* netlist. As mentioned in Section 2.4.1 the functional *spring* ele-

ment implemented is pin-compatible with lower-level *beam* elements in NODAS. Combining the spring stiffness computation with the translator, spring-mass behavioral models of inertial sensors can be generated automatically. This work 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.

The first sub-section describes the computational procedure to obtain the spring stiffness coefficients for an extended family of springs. 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 functional level is then demonstrated by design-space exploration of two inertial sensors. Finally, discussions and conclusions from the results obtained are presented. Future directions of work to include effects of inertia and damping and extension for completely arbitrary spring topologies are outlined in the final sub-section of chapter.

4.2 Spring Stiffness Computation

4.2.1 Background

Springs are a very important part of the inertial sensor design process. Using the energy methods outlined in Chapter 2 and described in [41][61], stiffness matrices (which are composed of analytical models for each stiffness constant) have been derived previously for beams, crab-leg, u-shaped, serpentine and folded-flexure springs. Extension to new topologies involves detailed analysis of the spring using free-body diagrams of individual beam elements. There are a wide variety of spring topologies and it is practically impossible to pre-derive the stiffness matrix for each of them. Numerical alternatives such as building behavioral models through FEA have been proposed previously [62]. The commercial tool Coventorware also has an in-built methodology for generating macro-models of springs [54].

The previous chapter described in detail models for cross-axis coupling in specific spring topologies such as crab-leg, u-spring and serpentine springs. However, for other spring topologies, designers have to resort to finite element analysis or nodal simulation

using tools such as NODAS, with *beam* elements. The computational procedure presented in this section eliminates the need for finite element analyses to obtain linear spring coefficients. Furthermore, it also speeds up nodal analysis by replacing a number of beams with a single spring.

The automated procedure reported in [63] for computing the spring stiffness matrix for any single-chain configuration of beams is developed based on the modeling techniques described in Section 2.4.2 and [41]. This procedure only requires the designer to specify the spring as a layout or as a NODAS schematic composed of *beam* elements. In the next sub-section, the computational procedure is described in detail.

4.2.2 Stiffness Computation Procedure

The procedure essentially consists of first obtaining the compliance matrix of the spring and then computing the inverse of the compliance matrix. For single-chain of beams springs, the compliance matrix is simply the sum of the compliance matrix of each individual beam because there is no “branching” of forces at any point in the spring.

For any spring which is configured as a single chain of beams the forces and moments transmitted through a beam in the spring can be computed independent of all the other beams in the spring. In other words, the forces and moments passing through a beam only depend on the position of the beam with respect to the point of application of force. Based on this fact, the method for deriving the in-plane compliance matrix for a beam in a spring was described in Section 2.4.2. Using the same method, the out-of-plane compliance matrix for a beam can also be derived. For the *i*th beam, the compliance matrix is given as:

$$\begin{bmatrix} \alpha_{zzi} & \alpha_{z\phi_x i} & \alpha_{z\phi_y i} \\ \alpha_{\phi_x z i} & \alpha_{\phi_x \phi_x i} & \alpha_{\phi_x \phi_y i} \\ \alpha_{\phi_y z i} & \alpha_{\phi_y \phi_x i} & \alpha_{\phi_y \phi_y i} \end{bmatrix} \quad (4.1)$$

where,

$$\alpha_{zzi} = \frac{1}{3EI_{yi}l_i} \left(\frac{1}{3GJ_{xi}l_i} (-x_C y_{1i} - x_{1i} y_{2i} + x_C y_{2i} + x_{2i}(y_{1i} - y_C) + x_{1i} y_C)^2 + \right. \\ \left. \begin{aligned} & x_{1i}^4 + x_{2i}^4 - 3x_{2i}^3 x_C + 3x_{2i}^2 x_C^2 - x_{1i}^3 (x_{2i} + 3x_C) - x_{2i}^2 y_{1i}^2 + 3x_{2i} x_C y_{1i}^2 + \\ & y_{1i}^4 - x_{2i}^2 y_{1i} y_{2i} - y_{1i}^3 y_{2i} + 2x_{2i}^2 y_{2i}^2 - 3x_{2i} x_C y_{2i}^2 - y_{1i} y_{2i}^3 + y_{2i}^4 + \\ & 3x_{2i}^2 y_{1i} y_C - 6x_{2i} x_C y_{1i} y_C - 3y_{1i}^3 y_C - 3x_{2i}^2 y_{2i}^2 y_C + 6x_{2i} x_C y_{2i} y_C + \\ & 3y_{1i}^2 y_{2i} y_C + 3y_{1i} y_{2i}^2 y_C - 3y_{2i}^3 y_C + 3y_{1i}^2 y_C^2 - 6y_{1i} y_{2i} y_C^2 + 3y_{2i}^2 y_C^2 + \\ & x_{1i}^2 (3x_{2i} x_C + 3x_C^2 + 2y_{1i}^2 - y_{1i} y_{2i} - y_{2i}^2 - 3y_{1i} y_C + 3y_{2i} y_C) - \\ & x_{1i} \left(\begin{aligned} & x_{2i}^3 - 3x_{2i}^2 x_C + 6x_{2i} x_C^2 + x_{2i} y_{1i}^2 + 3x_C y_{1i}^2 - \\ & 2x_{2i} y_{1i} y_{2i} + x_{2i} y_{2i}^2 - 3x_C y_{2i}^2 - 6x_C y_{1i} y_C + 6x_C y_{2i} y_C + \end{aligned} \right) \end{aligned} \right)$$

$$\alpha_{\phi_x \phi_x i} = \frac{EI_{yi}(x_{1i} - x_{2i})^2 + GJ_{xi}(y_{1i} - y_{2i})^2}{EGI_{yi}J_{xi}l_i}$$

$$\alpha_{\phi_y \phi_y i} = \frac{GJ_{yi}(x_{1i} - x_{2i})^2 + EI_{xi}(y_{1i} - y_{2i})^2}{EGI_{yi}J_{xi}l_i}$$

$$\alpha_{z\phi_x i} = \alpha_{\phi_x z i} = - \left(\begin{aligned} & \frac{(y_{1i} - y_{2i})}{2EI_{yi}l_i} (x_{1i}^2 - x_{2i}^2 - 2x_{1i}x_C + 2x_{2i}x_C + y_{1i}^2 - y_{2i}^2 - 2y_{1i}y_C + 2y_{2i}y_C) + \\ & \frac{(x_{1i} - x_{2i})}{GJ_{xi}l_i} (x_C y_{1i} + x_{1i} y_{2i} - x_C y_{2i} - x_{1i} y_C + x_{2i}(-y_{1i} + y_C)) \end{aligned} \right)$$

$$\alpha_{\phi_y z i} = \alpha_{z\phi_y i} = \left(\begin{aligned} & \frac{(x_{1i} - x_{2i})}{2EI_{yi}l_i} (x_{1i}^2 - x_{2i}^2 - 2x_{1i}x_C + 2x_{2i}x_C + y_{1i}^2 - y_{2i}^2 - 2y_{1i}y_C + 2y_{2i}y_C) + \\ & \frac{(-y_{1i} + y_{2i})}{GJ_{xi}l_i} (x_C y_{1i} + x_{1i} y_{2i} - x_C y_{2i} - x_{1i} y_C + x_{2i}(-y_{1i} + y_C)) \end{aligned} \right)$$

$$\alpha_{\phi_x \phi_y i} = \alpha_{\phi_y \phi_x i} = \frac{(EI_{yi} - GJ_{xi})(x_{1i} - x_{2i})(y_{1i} - y_{2i})}{EGI_{yi}J_{xi}l_i}$$

$(x_{1i}y_{1i})(x_{2i}y_{2i})$ are the coordinates of the beam end-points

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

l_i is the length of the beam

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

G is the Shear Modulus of the material of the beam

I_{zi} is the moment of inertia of the i th beam cross-section about the z (thickness) axis

I_{yi} is the moment of inertia of the beam cross-section about the y (width) axis

J_{xi} is the torsion constant of the beam cross-section about the x (length) axis

The above compliance terms are obtained from the strain energy of bending and torsion. There are additional in-plane compliance terms which arise due to the strain energy of axial deformation. These terms are given as:

$$\alpha_{xxi} = \frac{(x_{1i} - x_{2i})^2}{E A I_i}$$

$$\alpha_{yyi} = \frac{(y_{1i} - y_{2i})^2}{E A I_i}$$

$$\alpha_{\phi_z \phi_z i} = 0$$

$$\alpha_{xyi} = \alpha_{yxi} = \frac{(x_{1i} - x_{2i})(y_{1i} - y_{2i})}{E A I_i}$$

$$\alpha_{x\phi_z i} = \alpha_{\phi_z x i} = 0$$

$$\alpha_{y\phi_z i} = \alpha_{\phi_z y i} = 0$$

The compliance matrix of the entire spring is obtained by summing the compliance matrices of the individual beam elements.

The inverse of the spring compliance matrix yields the spring stiffness matrix. This procedure is equivalent in the electrical domain to summing up the series resistances and inverting to obtain the equivalent conductance of a number of series connected resistors. In the next section, the second component of reduced order modeling strategy, i.e., translation of the circuit-level *beam-plate* schematic to the functional *spring-mass* schematic, is described.

4.3 Translation from Circuit-level to Functional Schematic

The translator implements the following tasks during the conversion to the functional schematic.

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.

The algorithm to convert a *beam-plate* schematic to a *spring-mass* schematic is illustrated with the help of two examples: the nested-gyroscope introduced in Figure 2.2 and a z-axis accelerometer. In order to provide more complete picture of the usefulness of this algorithm, the starting point for this illustration is the layout. Referring to the hierarchy of MEMS design levels as shown in Figure 2.6, traversal of the hierarchy from the layout level to the functional level will now be demonstrated.

4.3.1 Nested gyroscope

The layout of a nested gyroscope system is shown in Figure 4.1(a). A similar topology has also been employed in a gyroscope designed earlier [32]. 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 [64]. The layout extractor recognizes the different components in the layout and generates the NODAS schematic representation of the nested-resonator system shown in Figure 4.1(b). The actuation and sense combs seen in the layout are removed from the schematic since they are not relevant to the current discussion. 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

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. As shown in Figure 4.2, the functional model comprises only of 2 *spring* and 2 *mass* elements, while the circuit-level schematic has 5 *plate* and 76 *beam* elements. The complexity reduction is evident even by visual comparison between the circuit-level schematic and the functional schematic.

4.3.2 Z-axis accelerometer

Figure 4.3(a) shows the layout of a 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 4.3(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 functional schematic which is shown in Figure 4.4. The circuit-level schematic with all the four springs has more than 200 *beams* in it.

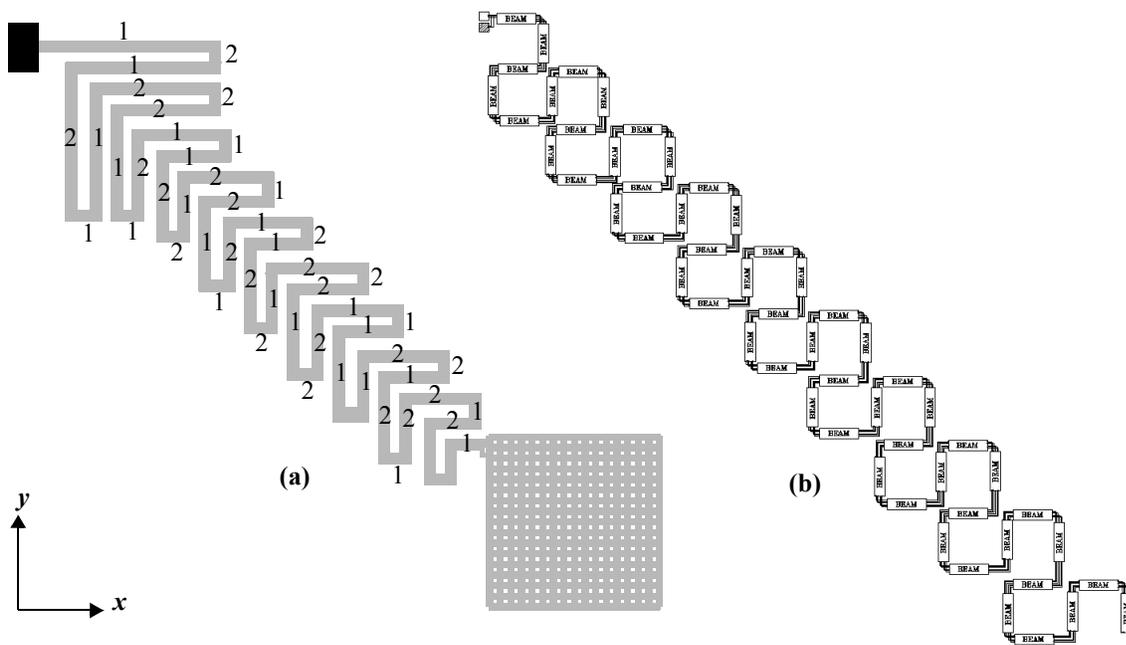


FIGURE 4.3. (a) Layout of a spring with about 50 beams connected to a proof-mass at one end and anchored at the other end (b) Corresponding NODAS schematic of the spring obtained through layout extraction. Beams marked with “1” and “2” have widths w_1 and w_2 respectively.

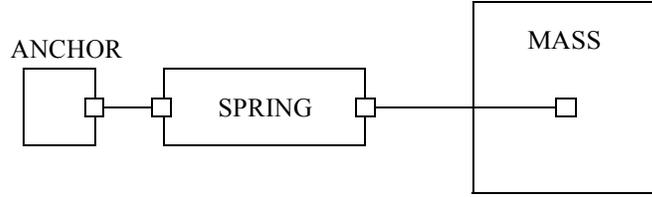


FIGURE 4.4. Functional schematic generated from the circuit-level schematic of the accelerometer shown in Figure 4.3(b)

In both the above examples, the nested-gyroscope and the z-axis accelerometer, the functional schematic, naturally, has much fewer elements compared to the circuit-level schematic. Therefore, it is expected that the simulation using the functional schematic will be considerably faster than that using the circuit-level schematic. However, the functional schematic is capable of capturing only the lowest resonant modes of the structure, whereas the circuit-level schematic is capable of revealing many more modes.

4.4 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 circuit level and at the higher functional level and the results are compared with respect to the accuracy and the simulation speed.

4.4.1 Verification of Spring Stiffness Computation

The layout of the spring used for verification is shown in Figure 4.5. The length and the width (measured from centers of the adjacent beams) of the vertical beams are varied while the horizontal beams are left unchanged. The comparisons of the spring stiffness computations with the FEA results are shown in Figure 4.6, Figure 4.7 and Figure 4.8 for k_{xx} , k_{yy} and $k_{\phi_z\phi_z}$ 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} , $k_{y\phi_z}$ and $k_{x\phi_z}$ 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

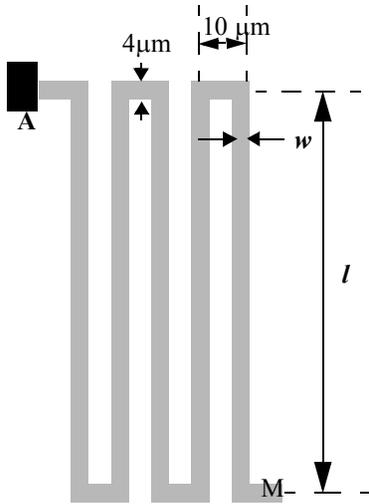


FIGURE 4.5. Layout of the spring used for FEA. A is the anchored point. M is the point to which the mass is attached. The length l and the width w of the vertical beam are varied over a range of values.

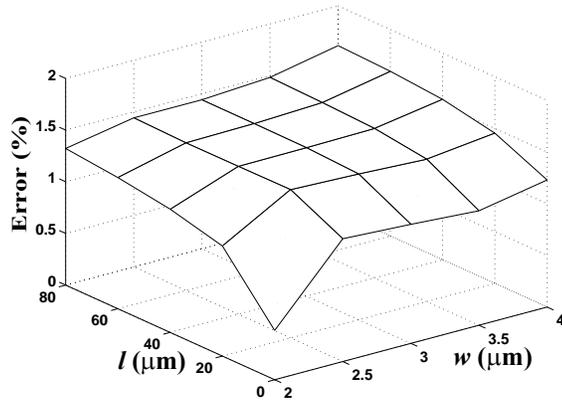


FIGURE 4.6. Comparison of spring stiffness computation for k_{xx} with FEA

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 ($4 \mu\text{m}$) is a significant fraction of the length ($10 \mu\text{m}$). For the comparisons shown the beam length was measured from the edge of the adjacent beams and a correction of:

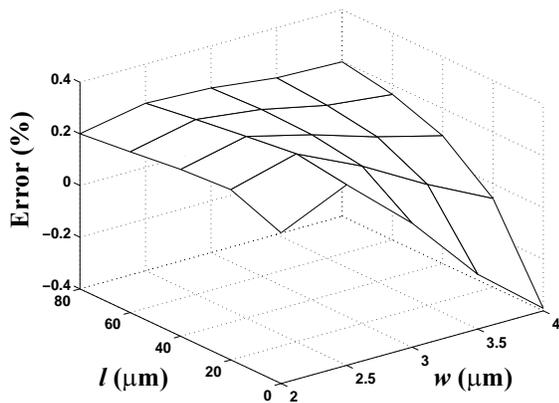


FIGURE 4.7. Comparison of spring stiffness computation for k_{yy} with FEA

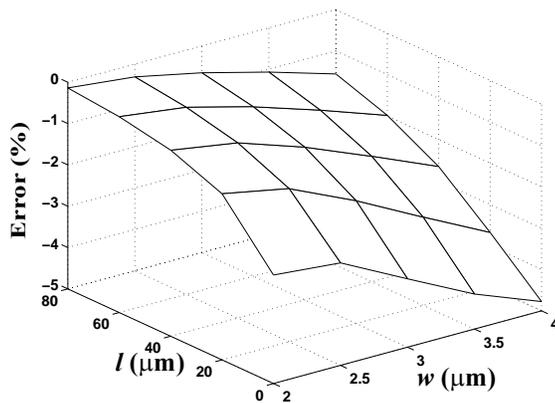


FIGURE 4.8. Comparison of spring stiffness computation for $k_{\phi_z\phi_z}$ with FEA

$$0.25 \left(\frac{w_{adjacent}}{w_{beam}} \right) (w_{adjacent}) \quad (4.2)$$

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.

4.5 Comparison of Circuit-level Schematic with Functional Schematic Simulation

The nested-gyroscope and the z -axis accelerometer are simulated at both the circuit-level and the functional level and the results are compared with respect to the accuracy and simulation time.

4.5.1 Example 1: Nested-Gyroscope Design Space Exploration

One of the design issues in a nested-resonator gyroscope 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

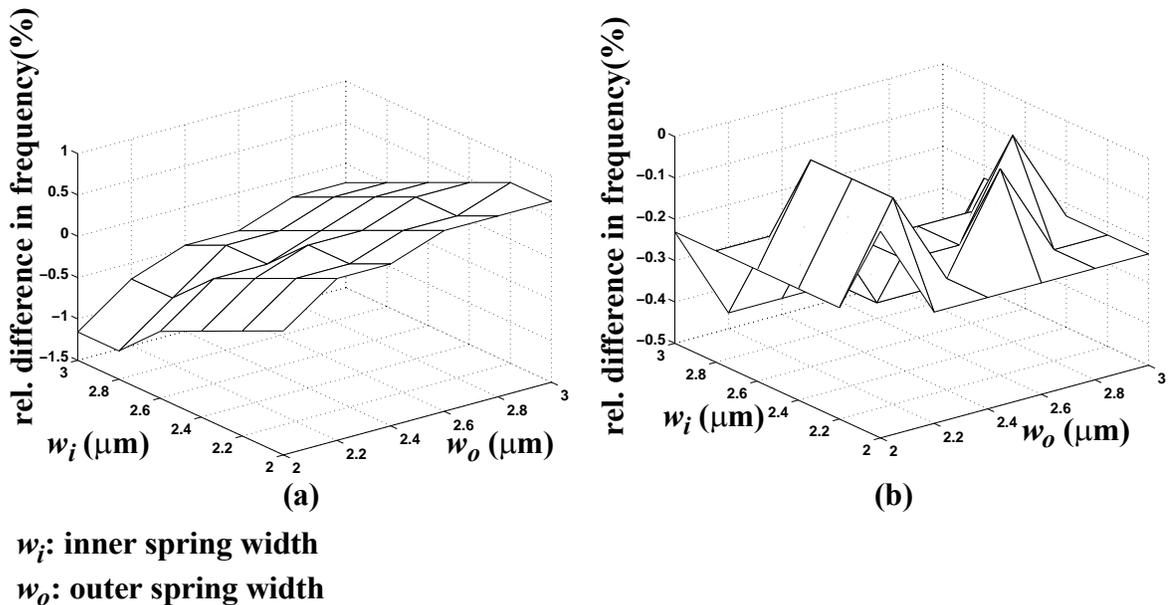


FIGURE 4.9. Difference in the resonant frequency extracted from the schematic simulations and behavioral simulations (a) drive-mode (b) sense-mode

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.

A design space exploration is done on the nested-gyroscope using circuit-level and functional simulations. The widths of the beams in the outer and inner springs were chosen as the two design variables. AC analysis of the schematic shown in Figure 4.1(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 functional schematic of the nested gyroscope (shown in Figure 4.2) was done. 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 4.9(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.

4.5.2 Example 2: Resonance Frequency Analysis of Z-axis Accelerometer

The sensitivity of an accelerometer is inversely proportional to square of 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 4.3(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 4.3(a). For different settings of these design variables AC analysis was done on both the circuit-level schematic as well as the functional schematic combined with the spring computation. The functional 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.

Thus, in both cases, a significant improvement in simulation time is obtained at the expense of a small loss in accuracy, which can be attributed to the fact that inertial effects have been neglected. This small loss in accuracy is usually acceptable for applications such as simulation-based synthesis. However, the improvement in simulation time will, in turn, lead to significant reduction of synthesis time.

4.6 Inertial and Viscous Effects and Extension to Arbitrary Spring Topologies

In this section, extensions to the stiffness computation procedure to include inertial and viscous effects are presented. The extensions are proposed for arbitrary spring topologies. The goal is to reduce a multiple terminal network of interconnected beams to a 2-port spring, retaining the ports at which the network of beams originally connected to other elements in the schematic while preserving the significant modes of the original network.

4.6.1 Model Formulation

A mechanical second-order system with n degrees of freedom can be written as:

$$[M] \cdot [\ddot{X}] + [B] \cdot [\dot{X}] + [K] \cdot [X] = [F] \quad (4.3)$$

where the X vector contains the position of the n nodes and the F vector contains the external forces acting at the n nodes, i.e.,

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \text{ and } F = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{bmatrix}$$

The M and the B matrices are assumed to be diagonal. The K matrix may have cross-axis coupling terms and therefore is not diagonal, but is a symmetric matrix. Introducing additional state variables $V = \dot{X}$, the above equations can be rewritten as:

$$\begin{bmatrix} B & M \\ -M & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dots \\ \dot{v}_1 \\ \dot{v}_2 \\ \dots \end{bmatrix} + \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ v_1 \\ v_2 \\ \dots \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.4)$$

Note that in the above equation the first matrix is skew-symmetric while the second one is symmetric. These properties may come in handy later on to preserve the passivity of the reduced-order models [65].

Let there be p nodes where the network of beams connects to other elements (such as a *plate* or an *anchor*). These p nodes can have external forces applied to them. This means that the force vector on the right hand side can have non-zero entries only at these p nodes. The original X vector is partitioned into two sub-vectors X_1 of size p and X_2 of size $n - p$. The reduced-order model should have only p nodes. Rewriting (4.4),

$$\begin{bmatrix} B & M \\ -M & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{V}_1 \\ \dot{V}_2 \end{bmatrix} + \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.5)$$

The X_1 and the X_2 vectors are not linearly independent i.e., the X_2 vector can be expressed in terms of the X_1 and V_1 vectors. Similarly the V_2 vector can be expressed in terms of the V_1 vector. If reduced-order matrices B_p for the visco-inertial properties and K_p for the elastic properties then:

$$\begin{bmatrix} B_p \end{bmatrix} \cdot \begin{bmatrix} \dot{X}_1 \\ \dot{V}_1 \end{bmatrix} + \begin{bmatrix} K_p \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ V_1 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \end{bmatrix} \quad (4.6)$$

In order to convert (4.5) into (4.6) a transformation Z with $2n$ rows and $2p$ columns is required such that:

$$Z_{2n \times 2p} \cdot \begin{bmatrix} X_1 \\ V_1 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ V_1 \\ V_2 \end{bmatrix} \quad (4.7)$$

Observing the form of the above equation and recalling that $V_1 = \dot{X}_1$ and $V_2 = \dot{X}_2$, Z can be written as:

$$Z_{2n \times 2p} = \begin{bmatrix} I_{p \times p} & 0 \\ Q_{(n-p) \times p} & 0 \\ 0 & I_{p \times p} \\ 0 & Q_{(n-p) \times p} \end{bmatrix} \quad (4.8)$$

where, Q is a matrix whose computation is described in the following section. Substituting (4.7) in (4.5):

$$\begin{bmatrix} B & M \\ -M & 0 \end{bmatrix} \cdot Z \cdot \begin{bmatrix} \dot{X}_1 \\ \dot{V}_1 \end{bmatrix} + \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} \cdot Z \cdot \begin{bmatrix} X_1 \\ V_1 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.9)$$

Pre-multiplying the equation with the transpose of Z , Z^T :

$$\left(Z^T \cdot \begin{bmatrix} B & M \\ -M & 0 \end{bmatrix} \cdot Z \right) \cdot \begin{bmatrix} \dot{X}_1 \\ \dot{V}_1 \end{bmatrix} + \left(Z^T \cdot \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix} \cdot Z \right) \cdot \begin{bmatrix} X_1 \\ V_1 \end{bmatrix} = Z^T \cdot \begin{bmatrix} F_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.10)$$

Comparing (4.10) with (4.6) and using (4.8) (to verify that the right hand sides are identical) we can compute the reduced-order B and K matrices, if the Z transformation matrix is known. In the next section we will describe the computation of the Z matrix.

4.6.2 Model-order Reduction

This section describes the model-order reduction process which is basically the process of expressing the state of n displacements in the X vector and the n velocities in the V vector in terms of only the p displacements and the p velocities in the reduced-order displacement and velocity vectors, X_1 and V_1 respectively. The transformation matrix Z brings about the reduction of the model order.

Expanding and rewriting (4.5):

$$\begin{bmatrix} B_{11} & 0 & M_{11} & 0 \\ 0 & B_{22} & 0 & M_{22} \\ -M_{11} & 0 & 0 & 0 \\ 0 & -M_{22} & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{V}_1 \\ \dot{V}_2 \end{bmatrix} + \begin{bmatrix} K_{11} & K_{12} & 0 & 0 \\ K_{21} & K_{22} & 0 & 0 \\ 0 & 0 & M_{11} & 0 \\ 0 & 0 & 0 & M_{22} \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.11)$$

Since the derivatives are zero at steady state:

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \cdot \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \end{bmatrix} \quad (4.12)$$

i.e.,

$$[K_{21}X_1 + K_{22}X_2] = 0 \text{ i.e., } [X_2] = [-(K_{22})^{-1}K_{21} X_1] \quad (4.13)$$

and,

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} I \\ -(K_{22})^{-1}K_{21} \end{bmatrix} X_1 \quad (4.14)$$

Comparing (4.14) with (4.7) and (4.8) we have:

$$Z_{2n \times 2p} = \begin{bmatrix} I & 0 \\ -(K_{22})^{-1}K_{21} & 0 \\ 0 & I \\ 0 & -(K_{22})^{-1}K_{21} \end{bmatrix} \quad (4.15)$$

Now the reduced-order system matrices can be written as:

$$\tilde{K} = \begin{bmatrix} K_{11} + Q^T K_{12} & 0 \\ 0 & M_{11} + Q^T M_{22} Q \end{bmatrix} \text{ and}$$

$$\tilde{B} = \begin{bmatrix} B_{11} + Q^T B_{22} Q & M_{11} + Q^T M_{22} Q \\ -(M_{11} + Q^T M_{22} Q) & 0 \end{bmatrix}$$

where,

$$Q = \begin{bmatrix} -(K_{22})^{-1}K_{21} \end{bmatrix} \quad (4.16)$$

Thus, by computing the Q matrix the reduced order model can be obtained.

4.6.3 General Reduced-order Modeling

The reduced-order model derived in Section 4.6.2 has the same zeroth order moments (steady state solution) as the original model. In order to match higher order moments also, additional nodes need to be introduced into the reduced-order model. The transformation matrix Z will then look like:

$$Z_{2n \times 2(p+r)} = \begin{bmatrix} I_{p \times p} & E_1 & 0 & 0 \\ Q_{(n-p) \times p} & E_2 & 0 & 0 \\ 0 & 0 & I_{p \times p} & E_1 \\ 0 & 0 & Q_{(n-p) \times p} & E_2 \end{bmatrix} \quad (4.17)$$

where, the vectors $E_{n \times r} = \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$ are orthonormal within themselves and to $\begin{bmatrix} I_{p \times p} \\ Q_{(n-p) \times p} \end{bmatrix}$

and they jointly form a basis for the first $p + r$ moments of the original system. The additional vectors E , can be obtained using Arnoldi methods as shown in [65][66]. However, it is not clear whether the first $p + r$ moments of the transformed system matrices will match those of the original system. If the basis Z were orthonormal this has been proved earlier [65]. But the first p columns of the basis Z are not orthonormal. Therefore, additional work is required to prove or disprove the moment matching properties of the transformed system matrices.

4.7 Summary

This chapter addressed the functional level of the MEMS design hierarchy and extraction from the circuit-level to the functional level. The extraction was accomplished by combining a simple method for computing stiffness matrices for springs with an algorithm to combine multiple beams to a *spring* element and multiple connected plates to a single *mass* element. This stiffness computing procedure can handle any single-chain configuration of beams and is accurate to within 5% as long as all beams are at least as long as they are wide. Using the flow from layout to circuit level schematic to functional schematic, faster design techniques for inertial sensors were demonstrated. For a complex spring with about 50 beams, the simulations using the functional schematic coupled with the spring computation procedure were up to 100 times faster than nodal simulation using *beam* and *plate* atomic elements. Extensions of the stiffness computation procedure to arbitrary spring topologies and to include inertial and viscous effects have also been proposed to overcome current accuracy limitations.

Chapter 5. Electrostatic Modeling of CMOS-MEMS

Comb

5.1 Introduction

In the previous two chapters, elastic modeling was described in extensive detail particularly with regard to elastic cross-axis coupling. In this chapter, the domain of interest is electrostatics. Electrostatic combs are used extensively in MEMS for sensing and actuation. As described in Section 2.4.3, two main quantities of interest in the electrostatic domain are the capacitance and force. A modeling methodology for capacitance and force of complex multi-conductor structures is developed in this chapter. Measurements on fabricated variable capacitance test structures are used to verify the models.

Linear combs with dominant motion along the length of the combs [67] (Figure 5.1(a)) are used in microgyroscopes for actuation since they produce constant force over large amplitudes [10]. Differential combs, shown in Figure 5.2, are used for sensing the induced oscillations in gyroscopes. Since microgyroscopes are highly sensitive to spurious forces, it is important to estimate the multi-directional actuation forces produced by the simple comb. Furthermore, since temperature-dependent microstructure curling in CMOS-MEMS [7] gyroscopes can lead to drive amplitude variation with temperature,

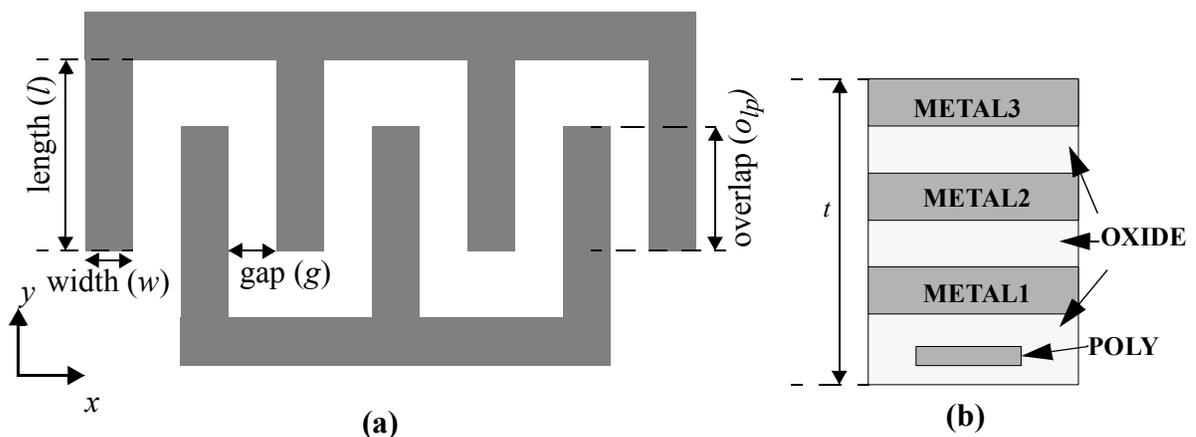


FIGURE 5.1. (a) Top view of a simple comb with three comb fingers (the lesser of the two numbers is taken as the number of fingers in the comb) (b) Cross-section of a comb finger in the CMOS-MEMS process

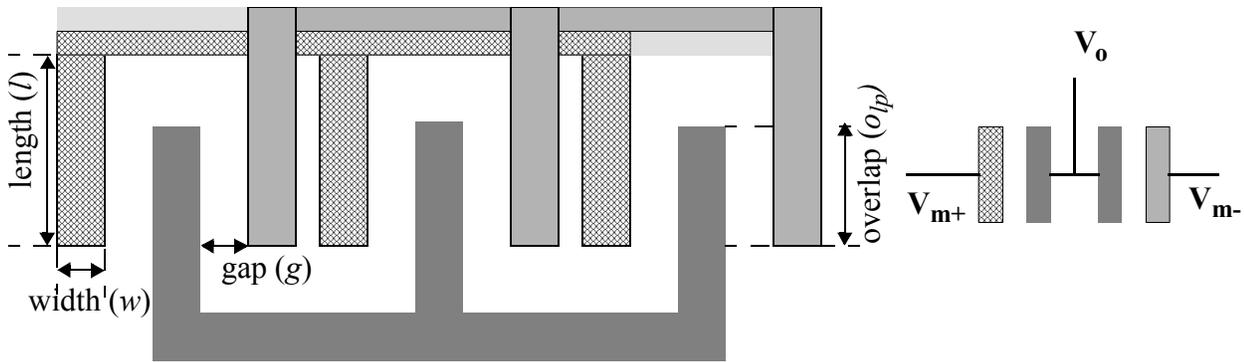


FIGURE 5.2. Top view of a differential comb along with the equivalent capacitive divider schematic.

robust design requires curvature-inclusive comb actuator models. Simulation of manufacturing variations requires models which are valid over a range of geometrical parameters. When used for motion sensing, differential combs require an accurate capacitance model. The modeling problem is further complicated in a multi-layered CMOS-MEMS comb (Figure 5.1(b)) which has multiple edges and vertices on each finger. In addition to providing accurate values for the capacitance and force, a behavioral model should also conserve energy [68]. Behavioral simulation can be used effectively to aid in gyroscope design only if the comb models provide reasonable estimates of capacitance and force.

In Chapter 3 and Chapter 4 the elastic properties of beam elements were analyzed. There are important fundamental differences between the *beam* and the *comb* models. Unlike elastic beam differential equations, which have a commonly valid and accurate closed-form general solution, the Laplace equation defining the electrostatic behavior of combs has closed form solutions only for a limited number of symmetrical boundary conditions, which are often restricted to two dimensions. Furthermore, while *beam* elements are most commonly used in their linear region of operation in inertial sensors, *comb* capacitance and force are inherently non-linear. Usually the number of *comb* elements (2 to 4) in an inertial sensor is much smaller compared to the number of beam elements (>

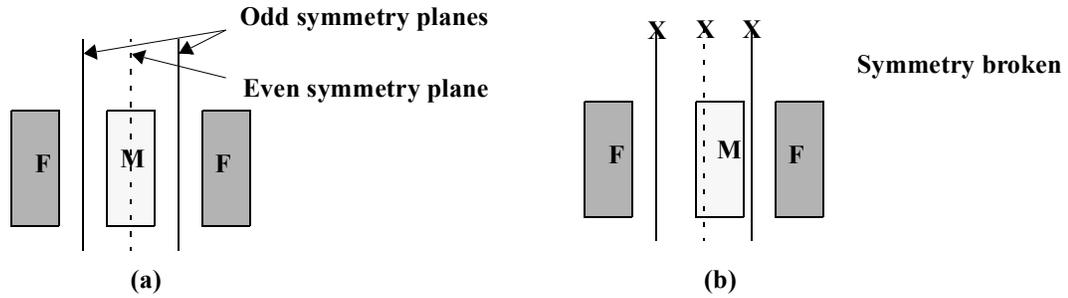


FIGURE 5.3. (a) Cross-section of a comb showing two fixed fingers (F) and one movable finger (M) along with odd and even symmetry planes used in derivation of conformal-mapping based analytical models. (b) Movement in the gap direction breaks the symmetry

15). These differences are summarized in Table 5.1. These differences suggest that differ-

Table 5.1 Comparison of *beam* and *comb* models

Element type	<i>beam</i>	<i>comb</i>
Solution of physics	accurate, 3D, analytical	approximate, 2D, analytical
Non-linearity	only for large displacements	even for small displacements
Number in schematic	44	4

ent modeling strategies be used for the *beam* and the *comb* models. Accordingly, an alternate modeling methodology for electrostatic elements is presented in this chapter. Before delving into the modeling methodology, a brief review of existing public literature is given in the following section.

5.2 Background

2D analyses of the comb-drive cross-section usually use conformal mapping [42][51] techniques. These analyses are primarily aimed at the simple comb with the dominant motion being in the direction of the length of the comb-fingers. Capacitance change due to vertical or lateral motion has been modeled using approximate analytical equations for single-layer comb fingers [51] or by numerical simulation of fixed-geometry 2D multi-layer cross-sections [69][70][71]. Angled comb-finger side-walls (observed in polysilicon microstructures) have been modeled by superposing curve-fits onto nominal analytical equations [72].

The conformal-mapping based analytical models for fringe capacitance are derived using symmetry assumptions. However, movement in the gap direction breaks the symmetry as shown in Figure 5.3 and therefore, the analytical models are not strictly valid. None

of the above-referred models take into account the fringe capacitance at the tips of the fingers, and charge concentration effects at finger corners. While the numerical models [69][70][71] take into account the multi-layer nature of CMOS comb-drives (Figure 5.1(b)), they are all valid only for fixed finger width and gap between the fingers. Furthermore, because of the 2D nature of the models, capacitance changes and vertical forces due to temperature-dependent vertical curling of comb-fingers can only be approximated by assuming a mean vertical position of two uncurled combs.

Numerical methods such as Finite Element Analysis (FEA) and Boundary Element Analysis (BEA) can capture charge concentration, curling and generalized motion effects on capacitance and force. However, numerical convergence needs to be closely monitored. Experience with BEA done in this thesis suggests that even four times higher refinement of a boundary element mesh which yielded converged capacitance values did not yield converged force values. This is probably because capacitance convergence requires only that the overall charge on a conductor does not change with more refinement, however, force convergence imposes a much stricter condition that the charge distribution remains invariant with more refinement. Also, numerical methods requiring meshed models and significant computation time, are not convenient for direct inclusion into a system simulation loop. Therefore, the modeling goal of this work is a behavioral description of the comb which exhibits the ease of use of analytical equations in system-level simulation [7] while incorporating the extended validity range of numerical methods.

In [73] Gabbay et al. have presented a general macromodeling system, which employs a rational fraction of multivariate polynomial as the fitting equation. However, the proposed model consists of analytical equations at the core superposed with polynomial curve fits for data obtained from 3D BEA [74]. The former approach is more suited (and probably necessary) for arbitrary shaped, non-parameterized geometry, deformable actuator systems. The electrostatic comb can be generally considered to be non-deformable (except for curvature, which is being treated as a geometrical parameter) and also, the model domain not only includes the position and orientation of the comb, but also the geometrical parameters. Additionally, by combining analytical models with regression one can take advantage of the existing literature on electrostatic comb models.

5.3 Modeling Goals and Approach

The modeling methodology is intended to capture the following physical effects:

1. Fringe capacitance for movement in gap direction
2. General 3D motion of the combs
3. Multi-layer effects in CMOS-MEMS combs
4. Vertical curling effects
5. Charge concentration effects at corners

The model is intended to be valid for multi-layer combs across a range of comb finger geometries, combined movement in vertical and lateral directions, and also include curling effects and fringe and corner capacitances. The model is also aimed at providing a energy-conserving description of the electrostatic comb actuator by fitting the derivative of the capacitance equation to numerically obtained force values. The modeling approach is to derive an analytical equation for comb movement in the gap direction and use numerical data to capture the other effects.

In Section 5.4 the library of existing analytical models is extended for movement of the comb in the direction along the gap. The analytical equations derived, which also form the core of the proposed model, are briefly described. The next section describes the numerical part of the modeling methodology. Choice of design variables, variable screening, design of experiments for BEA are described in Section 5.5. Section 5.6 details the form of the model and the approach for combined modeling of capacitance and force. The regression results, accuracy of the fit and an application of the model to estimate manufacturing variation induced drive amplitude changes in a gyroscope are discussed in Section 5.9. Experimental verification is described in Section 5.10 and the chapter summary is presented in Section 5.11.

5.4 Analytical Model for Movement in Gap Direction

In this section, the analytical equations which will be used in the fitting formula are described. The cross-section of a comb with three fixed (F) and two movable (M) fingers is shown in Figure 5.4(a). The movable fingers are displaced from the nominal position to the right by a distance x . The capacitance between the movable and fixed fingers can be

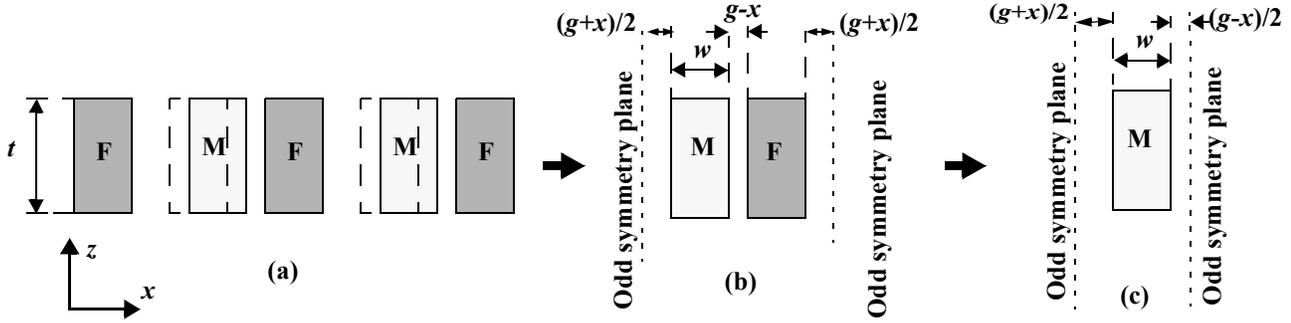


FIGURE 5.4. Simplification of laterally displaced comb cross-section using symmetry (a) Comb section showing 3 fixed fingers and 2 movable fingers displaced in the x direction from the nominal symmetrical position (shown with dotted lines). (b) Simplification by introduction of odd symmetry planes (c) Equivalent configuration with the fixed comb-fingers replaced by odd-symmetry plane placed midway between the rotor and stator fingers

written as a sum of two 2D capacitances, the parallel plate capacitance and the fringe capacitance. The parallel plate capacitance per unit length of overlap is given as:

$$c_p(t, g, x) = \epsilon_0 \left(\frac{t}{g+x} + \frac{t}{g-x} \right) \quad (5.1)$$

where, the parameters t , g and x are shown in Figure 5.4.

The derivation of the fringe capacitance is described below. Conformal mapping has been used in [42][51] to derive the capacitance for symmetric 2D cross-sections of different parts of the simple comb. However, movement of one comb in the gap direction destroys the symmetry boundary conditions assumed in those derivations and renders the equations for fringe capacitance invalid. An alternative approach, which is valid for movement along the gap direction, is used here. Assuming that there are a large number of fingers, symmetry is used to simplify the geometry. By placing two odd symmetry planes the simplified configuration of Figure 5.4(b) is obtained, which has two conductors and two odd symmetry boundary planes. Noting the presence of more symmetry, another odd symmetry plane can be inserted between the two conductors yielding the configuration of Figure 5.4(c). The configuration of Figure 5.4(c) is basically a rectangular conductor placed asymmetrically between two ground planes. Note that the configuration of Figure 5.4(c) is symmetrical about the x axis. Therefore, an even symmetry plane can be inserted to further simplify the geometry.

By use of conformal mapping this geometry can be mapped onto the real axis of a complex plane. At this point there are still too many right angles in the geometry to obtain a simple conformal mapping equation for capacitance calculation. As a rule, a geometry needs to have less than four right angles for it to have a closed form solution for conformal mapping [43]. Therefore, in order to isolate the fringe capacitance of only the top half of the configuration, the conductor is assumed to be semi-infinite. Following the conformal mapping, the parallel plate portion of the capacitance is subtracted to yield only the fringe capacitance. Such a simplified geometry is shown in Figure 5.5(a). Note that the structure now has only 2 right-angles, the other two turns being 180° as the edges are traversed starting from the points labelled A to M.

Conformal mapping for a similar problem is given in [75] (Figure 5.5). The conductor is assumed to be semi-infinite for the purpose of obtaining the fringe capacitance. The conductor and ground planes traversed in the order A-B-C-D-E-F-G-H-K-M are mapped from the Z plane to the real axis of the W plane. Note that the coordinate axes shown in Figure 5.5 such that the complex plane Z is defined as: $z = u_1 + iv_1$ and W as: $w = u_2 + iv_2$. A Schwarz-Christoffel transformation is used for this mapping [43]. The differential form of the mapping is given as:

$$\frac{dz}{dw} = A \frac{\sqrt{(w+1)(w+\beta)}}{w(w-\delta)} \quad (5.2)$$

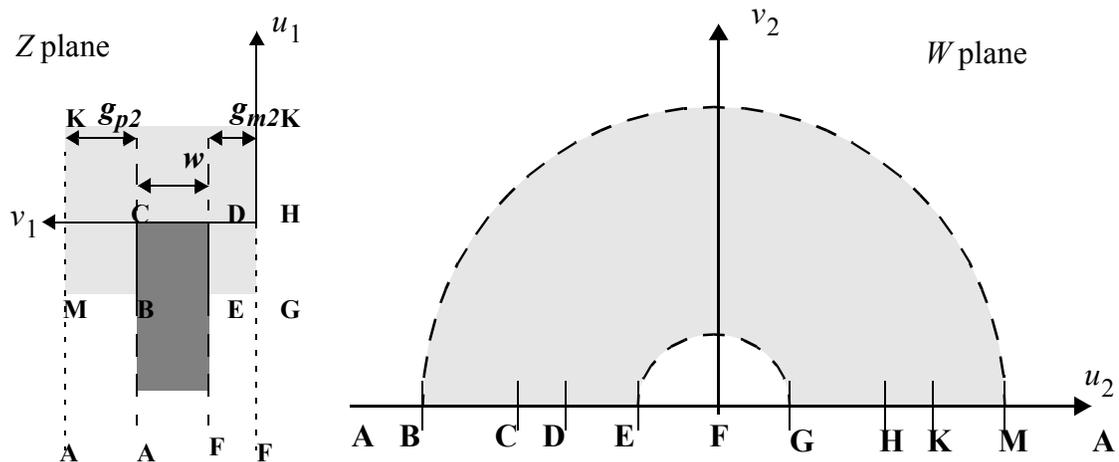


FIGURE 5.5. (a) Conformal mapping for a single conductor placed asymmetrically between two ground planes

where A , δ and β are computed by integration of (5.2) followed by applying the mapping between known points shown in Table 5.2.

Table 5.2 Mapping of points from Z plane to W plane

Point	Z plane	W plane	Point	Z plane	W plane
A	$-\infty$	$\pm\infty$	F	$-\infty$	0
B	$-t + j(g_{m2} + w)$	$-r_1$	G	$-t$	r_2
C	$j(g_{m2} + w)$	$-\beta$	H	0	r_H
D	jg_{m2}	-1	K	∞	δ
E	$-t + jg_{m2}$	$-r_2$	M	$-t + j(g_{m2} + w + g_{p2})$	r_1

$$z = \frac{2h}{\pi} \left(\alpha \operatorname{atanh} \left(\sqrt{\frac{(\delta + \beta)(w + 1)}{(1 + \beta)(w + \delta)}} \right) - \gamma \operatorname{atanh} \left(\sqrt{\frac{w + 1}{w + \delta}} \right) - \operatorname{atanh} \left(\sqrt{\frac{w + \delta}{\delta(w + 1)}} \right) \right) \quad (5.3)$$

where,

$$g_{m2} = \frac{g - x}{2}; g_{p2} = \frac{g + x}{2}$$

$$\alpha = \frac{w + g}{(g - x)/2}; \gamma = \frac{g + x}{g - x}$$

$$\delta = \frac{1}{2} \left(\alpha^2 - \gamma^2 - 1 + \sqrt{(\alpha^2 - \gamma^2 - 1)^2 - 4\gamma^2} \right); \beta = \frac{\delta^2}{\gamma}$$

$$\ln(r_2) = -\frac{\pi t}{2h} - 2\alpha \operatorname{atanh} \left(\sqrt{\frac{\beta + \delta}{\beta(1 + \delta)}} \right) + 2\gamma \operatorname{atanh}(\sqrt{\beta}) + \ln \left(\frac{4\beta}{\beta - 1} \right)$$

$$\ln(r_1) = \frac{1}{\gamma} \left(\frac{\pi t}{2h} + 2\alpha \operatorname{atanh} \left(\sqrt{\frac{1 + \delta}{\beta + \delta}} \right) + \gamma \ln \left(\frac{\beta - 1}{4} \right) - 2 \operatorname{atanh} \left(\frac{1}{\sqrt{\beta}} \right) \right)$$

The total fringe capacitance is then given by:

$$c_f = \lim_{t \rightarrow \infty} \left(\frac{2}{\pi} \left(\ln \left(\frac{r_1}{r_2} \right) \right) - c_{pp} \right) \quad (5.4)$$

where, $c_{pp} = \left(\frac{t}{g_{m2}} + \frac{t}{g_{p2}} \right)$ is the parallel plate contribution of the capacitance. To obtain

separate contributions of capacitance to the top and bottom ground plane, the mapping of the point K, where the field lines get split into the two ground planes, is required. Evaluat-

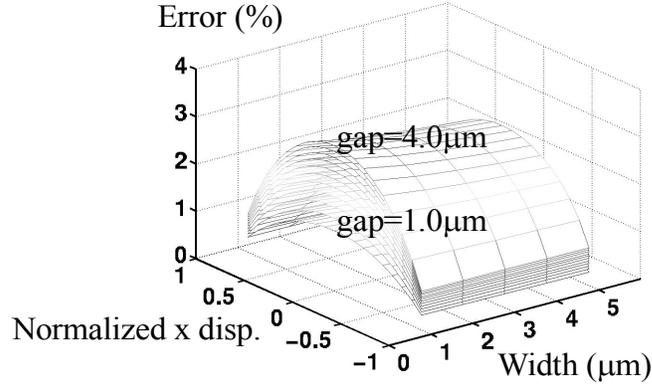


FIGURE 5.6. Comparison of analytical model adapted from [75] and FEA for x displacement of the comb-finger cross-section shown in Figure 5.4.

ing the limit in (5.4) and using the mapping of point K (δ in the W plane) the contributions of the capacitance to the left and right ground planes in Figure 5.4 are obtained as:

$$C_l = \frac{\epsilon_0}{\pi} \left(2\alpha \operatorname{atanh} \left(\sqrt{\frac{\beta + \delta}{\beta(1 + \delta)}} \right) - 2\gamma \operatorname{atanh} \left(\frac{1}{\sqrt{\beta}} \right) - \ln \left(\frac{4\beta}{\beta - 1} \right) + \ln(\delta) \right) \quad (5.5)$$

$$C_r = \frac{\epsilon_0}{\pi} \left(\left(\frac{1}{\gamma} \right) \left(2\alpha \operatorname{atanh} \left(\sqrt{\frac{1 + \beta}{\beta + \delta}} \right) + \gamma \ln \left(\frac{\beta - 1}{4} \right) - 2 \operatorname{atanh} \left(\frac{1}{\sqrt{\beta}} \right) \right) - \ln(\delta) \right) \quad (5.6)$$

$$c_f(w, g, x) = C_l + C_r \quad (5.7)$$

From this model, the equations for the lateral force between the rotor and the stator comb-fingers are also derived.

The model is verified by comparison with 2D FEA (Figure 5.6). The x displacement was varied from 0 to 0.9 times the gap on either side. The model matches FEA to within 2%. The maximum error of 2% is probably because of the finite number (5) of comb-fingers in the simulation. The model underestimates the fringe capacitance on the end fingers because it assumes symmetry boundary conditions. It is seen that the error is maximum for maximum gap. This is because, the fringe capacitance of the end-fingers (and other fingers too) is significant when the gap is larger. At smaller gaps, the parallel plate capacitance begins to dominate. The small error suggests that the model is reasonably accurate

even when there are only 5 fingers in the comb. Naturally, with more fingers, the accuracy is expected to improve. The total 2D capacitance per unit length per finger is given as:

$$C_{2D}(w, t, g, x) = c_p(t, g, x) + c_f(w, g, x) \quad (5.8)$$

where $c_f(w, g, x)$ is as shown in (5.7). The capacitance thus derived is extended to 3D by multiplying C_{2D} with the overlap between the comb-fingers. The equation shown is valid for a single-layer structure and does not account for curling and corner capacitances. The next section describes the design of experiments for numerical analysis used to extend (5.8) to incorporate vertical movement, multi-layer effects, curling and 3D charge concentration effects.

5.5 Design of Experiments and Simulation

In this section the procedure for designing the experiments for BEA-based data collection is described. Broadly, the design of experiments process can be viewed in 5 steps: comb parameterization, variable selection, variable screening, choice of variable ranges and data collection. Each of these steps is listed below.

5.5.1 Comb Parameterization and Variable Selection

The comb is parameterized into geometrical, position and orientation variables. The geometrical variables are: width (w), gap (g), overlap length (o_{lp}) and the number of fingers. The thickness of the comb depends on the composition of the multi-layer stack. For maximizing in-plane actuation force designers include all metal layers and the polysilicon layer. Therefore, we use the fixed comb cross-section containing all three metal layers and the polysilicon as shown in Figure 5.1(b). The three position (x, y, z) and three orientation variables (ϕ_x, ϕ_y, ϕ_z) correspond to the six degrees of freedom. Additionally, temperature (T) is also chosen as a variable because the curvature of the comb fingers is temperature dependent. Lateral combs are commonly used for actuation with amplitudes of the order of 5 μm . Since the force is independent of the finger length, the finger length is usually tightly linked to the overlap length. Assuming a clearance of about 5 μm at maximum displacement, the finger length (l) is set to be:

$$l = o_{lp} + 10\mu\text{m} \quad (5.9)$$

5.5.2 Variable Screening

The next step is to screen out variables which are known to have relatively small effect on the capacitance and force. Using data collected from initial runs varying each variable in isolation, it was decided that variables ϕ_y and ϕ_z can be kept at zero because of the small change in capacitance produced by them. The final set of variables used for the design of experiments are width, gap, overlap, positions x , y and z , orientation ϕ_x and the temperature T .

5.5.3 Choice of Variable Ranges

The values chosen for the variables are summarized in two sets in Table 5.3 and

Table 5.3 Experimental plan set 1 for simple comb BEA

Variable type	Name (units)	Lower bound	Upper bound	Intermediate values
Geometry	w : width (μm)	2.0	4.0	3.0
	g : gap (μm)	1.5	2.5	2.0
	o_{lp} : overlap (μm)	10.0	20.0	-
Position	x (μm)	0	0.8 g	0.4 g
	y (μm)	-4.0	4.0	0.0
	z (μm)	-2.0	2.0	0.0
Orientation	ϕ_x ($^\circ$)	-1.0	1.0	0.0
Temperature	temperature (K)	250	350	300
Constants	length (μm)	overlap + 10		
	numfingers	3		
	ϕ_y ($^\circ$)	0		
	ϕ_z ($^\circ$)	0		
Total Number of Runs				4374

Table 5.4. The first set of runs has 4374 points and has number of fingers set to 3. The second set has 10 fingers in each run and has a total of 192 runs. Higher number of fingers naturally lead to higher simulation times (e.g., the numfingers=10 run takes about 15 min-

utes, while a numfingers=3 run takes about 5 minutes). Therefore, the number of runs with higher number of fingers is kept very small.

Table 5.4 Experimental plan set 2 for simple comb BEA

Variable type	Name (units)	Lower bound	Upper bound	Intermediate values
Geometry	w : width (μm)	2.0	4.0	-
	g : gap (μm)	1.5	2.5	-
	o_{lp} : overlap (μm)	10.0	20.0	-
Position	x (μm)	0	$0.4 g$	-
	y (μm)	0	4.0	-
	z (μm)	-2.0	2.0	0.0
Temperature	temperature (K)	250	350	-
Constants	length (μm)	overlap + 10		
	numfingers	10		
	ϕ_x ($^\circ$)	-1.0	1.0	0.0
	ϕ_y ($^\circ$)	0		
	ϕ_z ($^\circ$)	0		
Total Number of Runs				192

The ranges for the width and the gap reflect commonly used values. The overlap length is limited to 20 μm since actuation combs are not likely to require greater than a few micrometers of movement, and when used for sensing, the capacitance change per unit length of displacement is independent of the overlap length. Larger number of fingers lead to larger number of panels in the boundary element mesh. Therefore, the number of fixed fingers was set to three to minimize the analysis time. Three fingers is the minimum number for which at least one finger on the movable part has symmetrical neighbors. Movements along the gap (x) direction are normally restricted to less than half the gap by limit stops or other means. However, the maximum movement is chosen to be 0.8 times the gap in order that the potential displacements are well within the validity range of the fitted model and also, in order to capture the highly non-linear capacitance change in the gap direction. The displacement along the length (y) was chosen to be $\pm 4 \mu\text{m}$, with a view to keeping a clearance as well as a minimum overlap of at least twice the gap in order to avoid potentially non-linear regions which are undesirable for gyroscope actuation.

For maximizing sensitivity and actuation force, designers attempt to ensure maximum vertical overlap in the comb by using curl-matching techniques [58]. However, the resultant curl-matching is usually never perfect and the ranges chosen for the z and ϕ_x variables are intended to capture curl mismatch in the combs. It should also be mentioned that $z = 0$ and $\phi_x = 0$ refer to the nominal curl-matched positions, from where z and ϕ_x are measured. At room temperature, the comb fingers are curled upward and the curling reduces as the temperature increases. The curvature of the comb fingers, computed using thermal multimorph theory presented in [58][76], is inversely proportional to the temperature. For a single finger the curvature ρ is related to the temperature T :

$$\frac{1}{\rho} \propto (T - T_0) \quad (5.10)$$

where, T_0 is the temperature where the finger will become flat and the constant of proportionality depends upon the composition of the finger and the material properties. The temperature range of interest is chosen to be ± 50 K around the room temperature so that the corresponding range in curvature covers possible variations in finger composition and material properties from sensor to sensor. Also, it should be noted that measured curvatures of the finger can be directly fed into the comb model rather than computing curvature from measured temperature.

5.5.4 Data Collection

AutoBEM software from Coyote Systems (now part of Cadence Design) [54] was used for BEA. A number of manual iterations with the BEA mesh led to an efficient template for the mesh which showed reduced simulation time with accuracy comparable to that obtained by adaptive refinement and iterative solution. Convergence of BEA using the template mesh was verified initially by splitting each element in the mesh into two and comparing the capacitance values obtained. A mesh generation program for electrostatic combs was implemented in C++. Figure 5.7 shows the boundary element mesh generated for a 10 finger curled comb. The curling and the curl-mismatch in the figure is exaggerated to aid visualization. The initial mesh is further refined internally inside AutoBEM to obtain very small elements near edges and vertices. Adaptive refinement is not used

because it requires multiple solutions which are time-consuming. However, the internal mesh refinement commands used (listed in Appendix A3) are tailored to closely replicate the optimal mesh produced by successive adaptive refinement. The final mesh used for solution leads to very small elements and is not shown in the figure.

Force computation using BEA did not converge even with a mesh which was more than four times as fine as (and therefore about 4 times slower than) the mesh which yielded converged capacitance values. The refinement steps for this mesh are also listed in Appendix A3. Therefore, only capacitance convergence was obtained for the first set of 4374 runs. Force convergence was attempted on a smaller subset of 1458 runs, in which ϕ_x was also kept constant at 0. However, convergence testing revealed that the force values were not accurate, though the capacitance values obtained were more accurate than those in the first set of 4374 runs. Therefore, numerical differentiation of the well-converged capacitance values in the second set of 1458 runs was used to obtain the forces (F_y and F_z), in the y and z directions respectively, for 972 intermediate points. F_y values were obtained at $y = -2$ and $+2 \mu\text{m}$, from capacitance values at $y=-4, 0$ and $4 \mu\text{m}$. F_z values at $z= -1$ and $+1 \mu\text{m}$ were obtained from capacitance values at $z=-2, 0$ and $2 \mu\text{m}$. This method could not be used for computing the force (F_x) in the x direction because, the capacitance

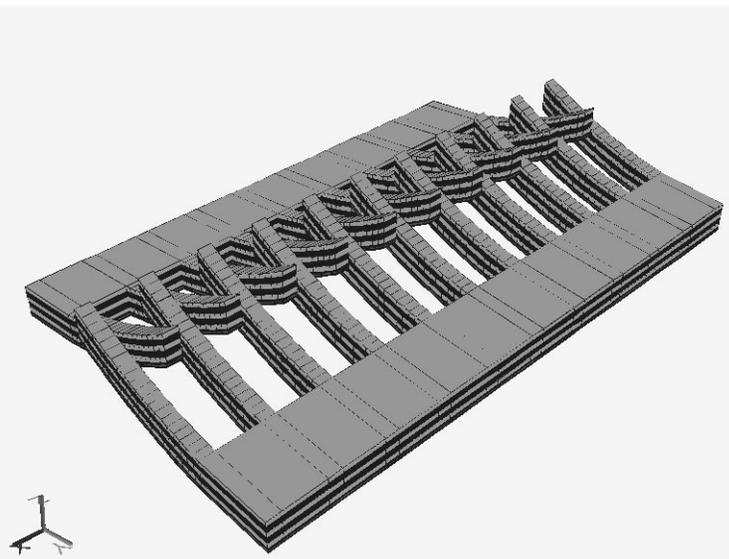


FIGURE 5.7. Boundary-element mesh for a 10 finger vertically curled lateral comb with all CMOS layers.

change in the x direction being highly non-linear, closely spaced points are required for precise computation of force. Therefore, a separate set of 162 runs, as summarized in Table 5.5, was designed to compute F_x at 54 points. Note that because of the rapid change

Table 5.5 Experimental plan to obtain F_x values for simple comb using BEA

Variable Type	Name (units)	Lower bound	Upper bound	Intermediate values
Geometry	w : width (μm)	2.0	4.0	-
	g : gap (μm)	1.5	2.5	-
	o_{lp} : overlap (μm)	10.0	20.0	-
Position	x (μm)	$0.4 g - 0.1$	$0.4 g + 0.1$	$0.4 g$
	y (μm)	-4.0	4.0	-
	z (μm)	-2.0	2.0	0.0
Temperature	temperature (K)	250	350	-

in capacitance with x , three capacitance values were used to compute each force value.

In this section the variables used in the design of experiments to obtain numerical data were introduced. The subsequent section details the modeling methodology.

5.6 Modeling Methodology

The form of the equation for the capacitance model is described first. Following this the combined modeling method for capacitance and force is introduced. With this methodology, the force models can use the same fitting equation and coefficients as the capacitance.

5.6.1 Capacitance Modeling

The two analytical equations described in Section 5.4 are used as the core of the model. The two functions are weighted by polynomial functions of variables. The fitting function used for linear regression is of the form:

$$C = c_1(\dots)f_1(\dots) + c_2(\dots)f_2(\dots) + c_3(\dots)f_3(\dots) \quad (5.11)$$

where,

t is the total thickness of the multi-layer comb finger which is held constant

(...) indicates function of all the variables in Table 5.3

c_1 , c_2 and c_3 are capacitance functions of the design variables given as:

$c_1(\dots) = (o_{lp} + y)c_p(t, g, x)$ is the 2D parallel-plate capacitance given in (5.1)

$c_2(\dots) = (o_{lp} + y)c_f(w, g, x)$ is the 2D fringe capacitance as given by (5.5) to (5.7)

$c_3(\dots) = 1$

$f_1(\dots)$, $f_2(\dots)$, $f_3(\dots)$ are the polynomial weighting functions. $f_3(\dots)$ corresponds to that part of the capacitance which is neither due to the parallel plate nor the analytical fringe capacitance.

Let us assume that the number of polynomial terms in f_1 , f_2 and f_3 are k_1 , k_2 and k_3 respectively and that all the terms are sequentially numbered from 1 to k , where $k = k_1 + k_2 + k_3$. The polynomial weighting function can be written as:

$$f_i(\dots) = \sum_j r_j \left(w^{a_{j1}} g^{a_{j2}} o_{lp}^{a_{j3}} \rho^{a_{j4}} x^{a_{j5}} y^{a_{j6}} z^{a_{j7}} \phi_x^{a_{j8}} \right) \quad (5.12)$$

where, the summation index j goes from 1 to k_1 for f_1 , from k_1+1 to k_1+k_2 for f_2 , and from k_1+k_2+1 to $k_1+k_2+k_3$ for f_3 , r_j are the coefficients and a_{j1}, \dots, a_{j8} are the powers to which the respective variables are raised to in the j th polynomial term. Note that the unity term as well as negative indices can also be included in the polynomial representation. The coefficients of the polynomial terms will be obtained by regression. If the entire fitting equation is expanded out then it is seen that each fitting coefficient is associated with a single term which is a product of one of the capacitance functions (c_1, c_2, c_3) and one of the terms from the corresponding polynomial weighting function. This product term is referred to as a *predictor* in statistical analysis. For the above regression model, there are k predictor terms [77] which are products of the three capacitance functions c_1, c_2 , and c_3 and the polynomial terms in f_1, f_2 and f_3 respectively, associated with that function. Evaluating each of k predictor terms at the n capacitance data points, the model matrix can be formed as:

$$[C]_{n \times 1} = [p_1 \ p_2 \ \dots]_{n \times k} \begin{bmatrix} r_1 \\ r_2 \\ \dots \end{bmatrix}_{k \times 1} \quad (5.13)$$

where,

$[C]$ is the vector of capacitance values corresponding to n runs in the experimental plan.

p_j is the j th predictor vector. There are k predictor vectors of size $n \times 1$ corresponding to the n settings of the variables.

r_j is the regression coefficient associated with the j th predictor vector.

5.6.2 Combined Capacitance-Force Modeling

The force produced by an electrostatic actuator, assuming constant voltage, in a generalized direction ξ is given as:

$$F_\xi = \frac{1}{2} \left(\frac{dC}{d\xi} \right) V^2 \quad (5.14)$$

Therefore, we can write the regression model for the force (assuming unit voltage) in the direction ξ as:

$$[F_\xi]_{m_\xi \times 1} = \frac{1}{2} \left[\frac{dp_1}{d\xi} \ \frac{dp_2}{d\xi} \ \dots \right]_{m_\xi \times k} \begin{bmatrix} r_1 \\ r_2 \\ \dots \end{bmatrix}_{k \times 1} \quad (5.15)$$

where, m_ξ is the total number of data points for which we have the force values in the direction ξ . To obtain regression coefficients which produce accurate fitted values for both force and capacitance simultaneously, we combine (5.13) and (5.15) into a common regression model:

$$\begin{bmatrix} C \\ F_x \\ F_y \\ F_z \end{bmatrix}_{(n+m_x+m_y+m_z) \times 1} = \frac{1}{2} \begin{bmatrix} 2p_1 & 2p_2 & \dots \\ \frac{dp_1}{dx} & \frac{dp_2}{dx} & \dots \\ \frac{dp_1}{dy} & \frac{dp_2}{dy} & \dots \\ \frac{dp_1}{dz} & \frac{dp_2}{dz} & \dots \end{bmatrix}_{(n+m_x+m_y+m_z) \times k} \begin{bmatrix} r_1 \\ r_2 \\ \dots \end{bmatrix}_{k \times 1} \quad (5.16)$$

In the above equation n, m_x, m_y, m_z correspond to the number of data points for which capacitance, F_x , F_y and F_z are respectively available. As per the experimental plan described in Section 5.5, we have $n = 4374$, $m_x = 54$, $m_y = 972$ and $m_z = 972$. Since the capacitance and force obtained may differ in their relative magnitudes, weights may be necessary to scale the residual errors corresponding to each point so that points with smaller absolute values of capacitance or force but relatively large percentage errors are not ignored by the regression.

5.7 Differential Comb Modeling

In the previous sections the modeling methodology for the linear comb was described in detail. In this section modeling of the differential comb is summarized. There are some important differences between the linear comb and the differential comb:

1. There are three conductors in a differential comb as opposed to two in a linear comb
2. The comb fingers are typically much longer in a differential comb ($\sim 60 \mu\text{m}$) than in a linear comb ($\sim 10 \mu\text{m}$). Therefore, in-plane rotation, which is relatively insignificant in a simple comb, cannot be neglected in a differential comb
3. Longer fingers lead to greater simulation times

The design of experiments for the differential comb reflect the above differences. As before, two sets of the experimental plan are designed as shown in Table 5.6 and Table 5.7.

Table 5.6 Experimental plan set 1 for differential comb BEA

Variable type	Name (units)	Lower bound	Upper bound	Intermediate values
Geometry	w : width (μm)	2.0	4.0	3.0
	g : gap (μm)	1.5	2.5	2.0
	o_{lp} : overlap (μm)	50.0	80.0	-
Position	x (μm)	0	$0.8 g$	$0.4 g$
	y (μm)	-4.0	4.0	0.0
	z (μm)	-2.0	2.0	0.0
Orientation	ϕ_x ($^\circ$)	-1.0	1.0	0.0
	ϕ_z ($^\circ$)	0	0.5	0.25
Temperature	temperature (K)	250	350	300
Constants	length (μm)	overlap + 10		
	numfingers	3		
	ϕ_y ($^\circ$)	0		
Total number of runs				7290

Table 5.7 Experimental plan set 2 for differential comb BEA

Variable type	Name (units)	Lower bound	Upper bound	Intermediate values
Geometry	w : width (μm)	2.0	3.0	-
	g : gap (μm)	1.5	2.0	-
	o_{lp} : overlap (μm)	75.0	75.0	-
Position	x (μm)	0	$0.4 g$	-
	y (μm)	-4.0	4.0	0.0
	z (μm)	-2.0	2.0	0.0
Orientation	ϕ_x ($^\circ$)	-1.0	1.0	0.0
	ϕ_y ($^\circ$)	0	0.5	-
	ϕ_z ($^\circ$)	0	0.5	-
Temperature	temperature (K)	300	300	-
Constants	length (μm)	overlap + 10		
	numfingers	10		
Total number of runs				224

(Note that in set 2, the runs with non-zero ϕ_y and ϕ_z are only done with zero value set for all other position and orientation variables. In other words only one run each with non-zero ϕ_y and ϕ_z for each unique differential comb geometry is done. Therefore, the total number of runs is only 224.)

The total number of runs in set 1 is 7290. However, a few of these points have to be dropped from the plan because of the rotor fingers crashing into the stator fingers due to rotation. Of the 7290 points, 1458 are perfectly symmetric points i.e., $x = 0$ and $\phi_z = 0$. Of the remaining 5832 points the situations shown in Table 5.8 will lead to comb crashing

Table 5.8 Points to be omitted due to comb finger crashing

o_{lp} (μm)	g (μm)	x (μm)	ϕ_z ($^\circ$)	Number of points
50	1.5, 2.0, 2.5	0.8 g	0.5	243
50	1.5	0.8 g	0.25	81
80	1.5, 2.0, 2.5	0.8 g	0.25, 0.5	486
80	1.5	0.4 g	0.5	81
Total number of discarded runs				891

and are therefore, discarded. The total number of discarded points is 891. Therefore, the number of useful data points is $1458 + 5832 - 891 = 1458 + 4941$. Note that each of the 4941 points yields two capacitance values corresponding to the left and the right halves of the differential comb. Similarly 114 points in set 2 have different values for capacitance for the left and right halves of the differential comb. Therefore, the total number of data points to which the regression model is fitted is $1458 + 2*4941 + 110 + 2*114 = 11340 + 338 = 11678$. Additional runs for computing forces through numerical derivatives have not been done for the differential comb. Instead, the derivative of the capacitance values are used directly. This is acceptable because the displacements expected in a differential sense comb of a gyroscope are really small (few pm) and at the same time the voltages applied in a differential comb are much smaller than the voltages applied to the drive combs. Therefore, we can expect the forces produced in a differential sense comb to be modeled by using the derivatives of the fitted capacitance equation, which is very accurate for small displacements.

5.8 BEA Summary

The number of BEA and the type of data obtained from them for the linear comb is summarized in Table 5.9. The times shown are for analyses run on one CPU of a 450 MHz

Table 5.9 Summary of BEA runs for simple comb

Quantity	Number of values obtained	Number of BEA runs	No. of panels	Memory per run (MB)	Time per run (minutes)
Capacitance	$n=4374$	4374	22000	230	5
F_x	$m_x=54$	162	90000	850	24
F_y	$m_y=972$	1458	90000	850	24
F_z	$m_z=972$	1458	90000	850	24

Sun Ultra-80 workstation. Multi-processor usage leads to corresponding speed up. Note that the F_x values were computed using three closely spaced capacitance data points while the F_y and F_z values used a common set of 1458 capacitance values to compute derivatives at 972 points. The runs used to obtain capacitance values for computing forces used higher mesh refinement, and therefore, required higher memory and CPU times. The entire set of runs was accomplished in about two weeks time using a 5 CPU workstation. The final mesh for the differential comb has about 33000 panels and takes about 9 minutes on a single CPU of a 450 MHz Sun Ultra-80 workstation.

5.9 Results

The first part of this section describes the results of the curve fitting. The second part discusses the implementation of the resultant capacitance and force model for the combs in the NODAS environment and simulations to illustrate the applicability of the comb model.

5.9.1 Model Generation by Curve Fitting

The model matrix shown in (5.16) was constructed in the S-Plus [78] environment. Polynomial terms and fitting weights were introduced by iterative manual analysis of residual errors. The reusability of the proposed model justifies the time investment in this procedure. The final model for the simple comb has 106 coefficients and fits the capacitance data to within $\pm 3\%$ as shown in Figure 5.8 for 4374 points. It should be noted that

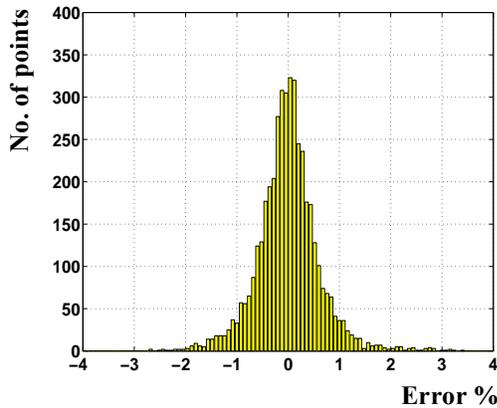


FIGURE 5.8. Histogram of error in % between model and numerical data for 4566 capacitance points. The error is almost within $\pm 3\%$

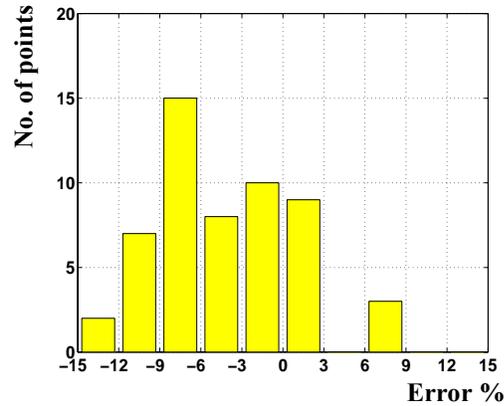


FIGURE 5.9. Histogram of error in % between model and F_x from numerical derivative of capacitance w.r.t x

there are very few points that lie outside the $\pm 2\%$ band. The fitted values of force in all three directions match the values obtained by numerical differentiation to about $\pm 10\%$, as seen in Figure 5.9, Figure 5.10 and Figure 5.11. The error tends to be high for points where the absolute values of force is small (i.e., the difference in the capacitances used to compute the derivative is small). This may be because the difference is close to the precision limit of the numerical capacitance values. Also, the plots indicate that the fit for F_y is the best of the three forces and the fit for F_x is the worst. This is probably because the capacitance is highly non-linear in the x direction and mostly linear in the y direction, with the variation in z direction being moderately non-linear leading to inaccurate numerical

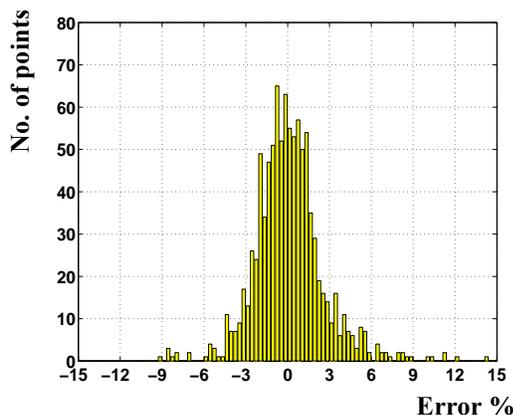


FIGURE 5.10. Histogram of error in % between model and F_y obtained from numerical derivative of capacitance w.r.t. y

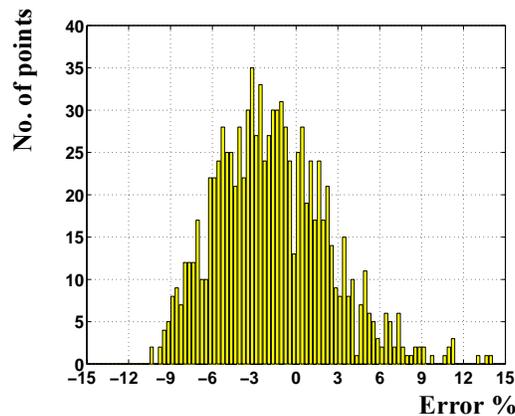


FIGURE 5.11. Histogram of error in % between model and F_z obtained from numerical derivative of capacitance w.r.t. z

derivatives. Furthermore, the model may require more polynomial terms to capture the non-linearities accurately. The fitting quality of the differential comb model which has 78 terms is shown in Figure 5.12. The absolute error is higher than 4% only at displacements equal to 80% of the nominal gap, i.e., about 1 μm . In microgyroscopes the full-scale sense displacements does not exceed 1 nm and therefore, the higher error at large displacements is of little concern.

5.9.2 Behavioral Model Implementation

The comb models were implemented in VerilogA[®], an Analog Hardware Description Language, as part of the NODAS framework and were used in simulation of a CMOS-MEMS gyroscope [10] (Figure 5.13). The complete listing of the comb models is provided in Appendix A4. In order to demonstrate the applicability of the model to study manufacturing variations two sets of simulations were done. An extended study of manufacturing variations on the gyroscope is described in [79] and will be expanded upon in Chapter 7. The scope of the current discussion is limited to the drive amplitude of the gyroscope.

One of the major issues in microgyroscopes is the Zero Rate Output (*ZRO*) and its variation with temperature and over time. The *ZRO* is closely related to the drive amplitude. The linear comb is used to generate the actuation force to set the gyroscope into oscillations. As noted earlier, vertical curl in the multi-layer structures, can lead to a verti-

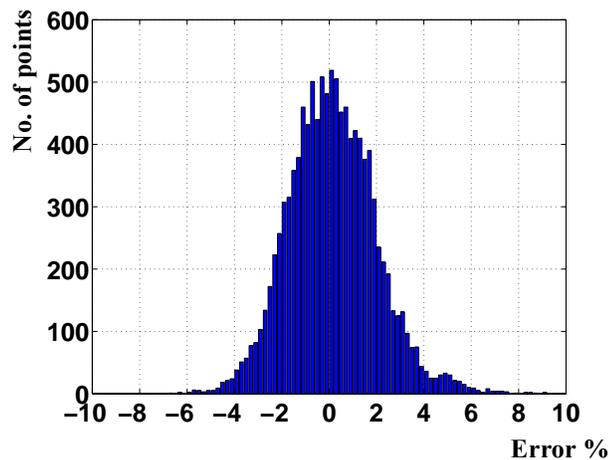


FIGURE 5.12. Histogram of capacitance error in % between model and numerical data for the differential comb

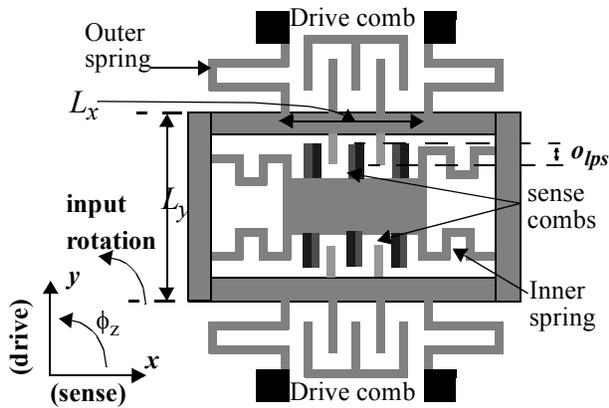


FIGURE 5.13. Functional diagram of the CMOS-MEMS gyroscope used in simulation [10]

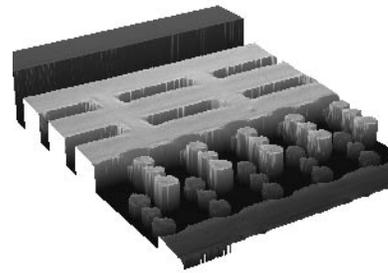


FIGURE 5.14. Interferometry image showing the vertical offset in drive comb

cal offset between the movable and fixed portions of the linear comb. Temperature changes can lead to variation in the vertical offset leading to a corresponding change in the actuation force. NODAS simulations using the behavioral comb model were done to capture the curvature-dependent actuation force. The vertical offset between the movable and fixed portions of the actuation comb in the gyroscope was varied from 0 to 1 μm (Figure 5.15). As seen in Figure 5.15, a 1 μm offset leads to about 4% decrease in the drive amplitude. This change in the drive amplitude directly corresponds to a change in the sensitivity of the gyroscope. Thus, a significant source of temperature-induced sensitivity variation in gyroscopes can be estimated using the behavioral comb models.

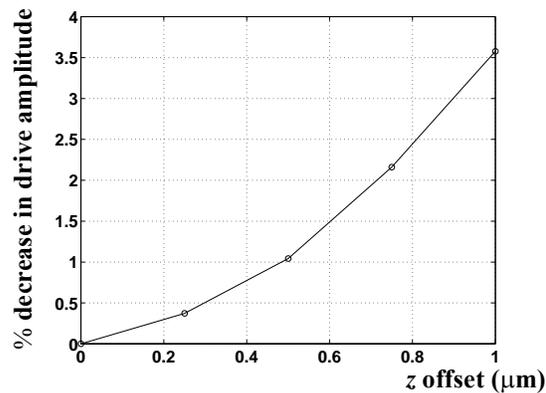


FIGURE 5.15. Drive amplitude variation of a gyroscope with change in vertical overlap between the movable and fixed portions of the comb actuator

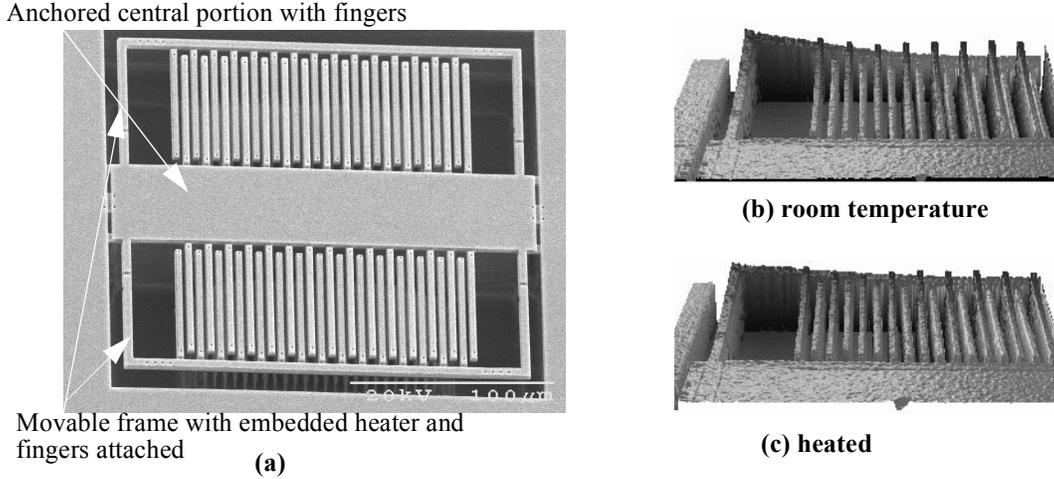


FIGURE 5.16. (a) SEM of capacitance test structure with in-built heaters. The curling can be changed by changing the current passing through the polysilicon wires which pass through the outer frame of the structure. Interferometry images of a quarter of the structure at (b) room temperature and (c) heated are also shown.

5.10 Experimental Verification

In order to validate the comb models experimentally, test structures for measurement of capacitance changes were fabricated (Figure 5.16). Each test structure consists of two sets of interdigitated comb fingers. The outer fingers are suspended from a frame which has an embedded polysilicon heater. The inner fingers are attached to the central portion which is anchored. The overall test structure is made up of two structures identical to that shown in Figure 5.16, which are connected in series as shown in Figure 5.17. Current passing through resistors R_1 and R_2 heats up the capacitors C_1 and C_2 respectively. Due to change in curvature of the fingers and increased vertical overlap between the fixed and movable finger, the capacitance values change. The capacitance change is sensed and amplified using a chopper-stabilized amplifier circuit [69], which has a known gain set by ratioed resistors. The output voltage is given as:

$$\frac{V_o}{V_m} = \frac{C_1 - C_2}{C_1 + C_2 + C_p} A \quad (5.17)$$

where, V_m is the fractional modulation voltage (i.e., $V_m = D(V_{mp} - V_{mn})$, where D is the duty cycle of the chopping waveform) and A is the overall gain.

The profile of the entire structure is obtained using interferometry for different values of heating current. For the same values of heating current the output voltage of the amplifier is also noted. The comparison of the capacitance change predicted by the model and the measured capacitance change is shown in Figure 5.18. There are two sets of curves corresponding to voltage applied to the two heaters. In both cases the values of capacitance change simulated using the model developed match the experimentally measured values to about 10% at higher heater voltages. At lower heater voltages the larger error is probably due to the limited accuracy of the profile of the comb fingers. All the fingers in the comb do not have the same vertical offset, because of curling of the movable frame. Therefore, the measurements are made on the fingers which approximately represent the mean vertical offset. This can also potentially contribute to the total error. Larger gaps due to overetching can also lead to lower measured capacitance change.

5.11 Summary

A modeling methodology which combines the ease of use of analytical equations and the higher accuracy of numerical methods has been demonstrated for simple and differential CMOS-MEMS combs. The models take into account the corner capacitances as well as curling of the multi-layer comb fingers. The methodology automatically results in an energy conserving model for the comb actuator. BEA is used to obtain capacitance values

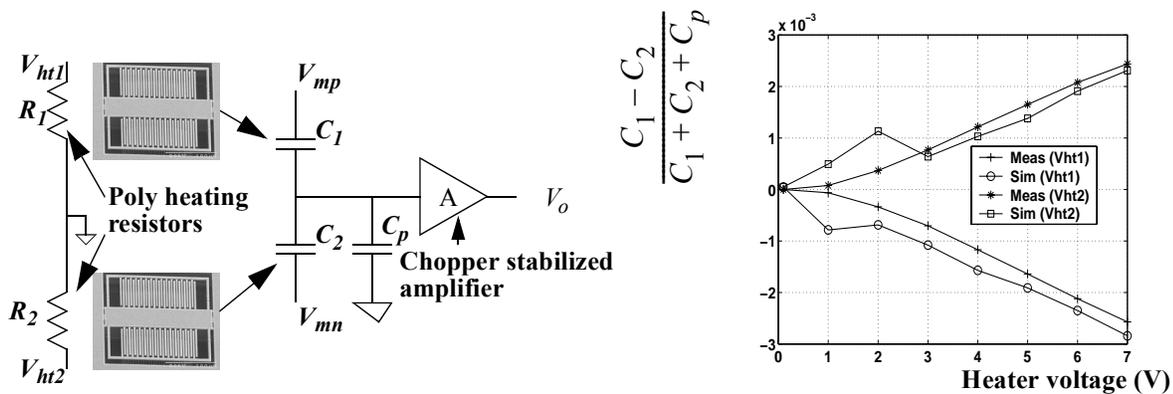


FIGURE 5.17. Capacitance change measurement schematic shown with heating resistors for each structure. C_p is the parasitic capacitor whose value is obtained using layout extraction.

FIGURE 5.18. Comparison of measured and predicted capacitance change. The two sets of measured data correspond to the voltage applied to each of the two resistors, with 0.1 V applied to the other resistor.

for a designed set of 4374 runs and force values for a reduced subset of these runs for the simple comb and a set of 5399 runs for the differential comb. The comb model fits the BEA capacitance data to within $\pm 3\%$ and to within $\pm 6\%$ for the differential comb. However, the error in the useful range is even smaller. Convergence of force requires higher mesh refinement for BEA. Therefore, BEA for a large number of runs to obtain converged force values was found to be infeasible. Numerical derivatives of a reduced subset of the capacitance runs were used to obtain force values. The fitted values of force match the numerically computed values to within $\pm 10\%$, though a large fraction of the points match to within $\pm 5\%$. The models obtained are used to predict the capacitance change of thermally actuated combs and they match measured capacitance changes to about 10%. The comb model has been implemented in a behavioral simulation framework and its applicability for simulating manufacturing variations in a gyroscope has been demonstrated.

Chapter 6. Convergence and Speed Issues in MEMS Behavioral Simulation

6.1 Introduction

The previous chapters dealt with various aspects of MEMS modeling, focusing on the elastic and electrostatic domains. The models derived in the previous chapters are encoded in VerilogA, a high-level hardware description language to describe analog behavior. Behavioral simulation using high-level hardware description languages (HDLs) has become very useful for mixed-domain MEMS simulation [80]. Behavioral models of mechanical, electrostatic, optical and fluidic components have been implemented in analog HDLs such as VerilogA [7], MAST [51] and VHDL-AMS [81]. Analog HDLs provide a powerful methodology to combine different physical domains, such as electrical, mechanical, thermal and others. Therefore, they are well-suited for integrated MEMS simulation. In this chapter, certain simulation issues arising from the use of analog HDLs in MEMS simulation are addressed.

Behavioral simulation provides the model developer with freedom to implement the physics of the component in a number of different ways. However, the high-level analog HDL code renders the final simulation matrix inaccessible to the developer. Therefore, the choice of the best implementation is not immediately apparent to the developer. Different implementations lead to different number of equations, convergence properties and simulation speed in transient analysis. Without a thorough understanding of the translation of the analog HDL code to the simulation matrix, the resulting simulation times may be non-optimal and the simulation may even be non-convergent in the worst case. Such problems can be compounded by simulation of high-Q resonant vibrations, as is the case in the drive mode of the gyroscope. The available reference material mainly addresses syntax issues and does not provide insight into mapping of analog HDL representation to the equations for nodal analysis [82][83]. In this chapter, analog HDL code is correlated to the matrix formulation during transient analysis, as presented in [84]. The convergence and simulation speed of transient analysis are then explained with the aid of the matrix formulation.

This work was mainly motivated by convergence problems encountered during behavioral simulations of the CMOS-MEMS gyroscope.

Comparisons between electrical circuit simulation and MEMS simulation for choice of *through* and *across* variables for nodal analysis have been suggested previously [62][81]. *Force* has been the preferred *through* variable. *Displacement* and *velocity* are possible candidates for the *across* variable. From a user perspective *displacement* as *across* variable is more convenient to observe motion. A preliminary evaluation of simulations using the two choices for the *across* variable showed that the *velocity* as *across* variable implementation has some convergence difficulties [85]. Simulation times have not been compared previously. A more comprehensive comparison of the convergence and simulation times for the two choices is presented here.

The matrix equations resulting from a behavioral model may have widely different coefficients due to the different numerical regions of interest in different domains. Wide range in matrix element values can lead to ill-conditioned matrices and thus to convergence difficulties. To overcome this problem scaling of specific domains has been implemented earlier [86]. The improvement in the simulation matrix condition number produced by such scaling is explained numerically in this chapter.

The general procedure for linear transient analysis is briefly described in the background section. This is followed by an explanation of the different implementations along with the analog HDL code and the expected matrix implementation in the simulator. Scaling of quantities for better convergence is then discussed, followed by simulation results and analyses. In the penultimate section, the guidelines for better convergence properties are illustrated through two implementations of a squeeze film damping model. Finally, conclusions and suggestions for analog HDL modeling are presented.

6.2 Background

The behavioral models in this chapter are implemented as part of the NODAS framework. The MEMS designer constructs a schematic representation of the device by interconnecting elements from this library. The differential equations for the elements in NODAS are encoded in VerilogA. The VerilogA language enables the model developer to

define a number of *natures* in each physical domain. Combining two *natures*, one for the across variable and one for the through variable the definition of a *discipline* is obtained. For example, the mechanical domain has three *disciplines*: *displacement-force*, *velocity-force* and *acceleration-force*. Every node introduced in the VerilogA model is defined to belong to one of the pre-defined discipline. Each *nature* also has associated values of absolute tolerances, called *abstols*, which are used to set the convergence criterion during simulation. The Spectre™ [87] simulator from Cadence is used for simulations.

The VerilogA module is based on *constitutive relationships*, which describe the behavior of the element, and *interconnection relationships*, which describe the structure of the network. The simulator combines *constitutive relationships* with *Kirchhoff's laws* in *nodal analysis* to form a system of differential-algebraic equations [88]. Numerical integration methods are employed to solve these equations for transient analysis. In the following sub-sections numerical integration in MEMS behavioral simulation is explained briefly, with the help of examples from the electrical domain. First, the general form of time-discretization is introduced. Following this, transient analysis of an inductor is used to develop the analogy between MEMS behavioral simulation and electrical circuit simulation.

6.2.1 Numerical Integration

During transient analysis, circuit simulators replace the time derivative operator with a discrete-time finite difference approximation and solve for the node variables at individual time points. Interval between time points (time step h) is controlled by the simulator to ensure accuracy of the finite difference approximation.

Common integration methods for time-discretization include Backward Euler (BE), Trapezoidal rule (TR) and Gear methods [89]. BE is used in this chapter to illustrate the formulation of transient analysis matrix due to its simplicity, accuracy and stability. In BE, the node value at time instant n , when time = t is computed based on the derivative value at t . For example, the equation relating the velocity v to the displacement x , $v = \dot{x}$, can be written as:

$$x(t) = \int_0^t v(\tau) d\tau \quad (6.1)$$

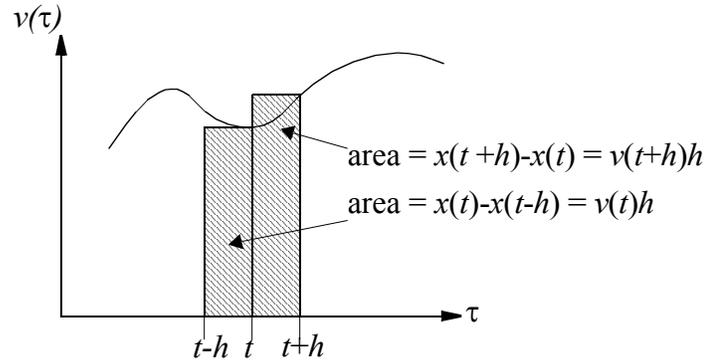


FIGURE 6.1. Computation of state variable using Backward Euler integration rule. The areas of the rectangles obtained by integration are shown.

The above equation is discretized with a time-step h . Solving for node variable *displacement* x at time t using the BE integration rule:

$$v(t) = \frac{x(t) - x(t-h)}{h} \quad (6.2)$$

$$\frac{x(t)}{h} = \frac{x(t-h)}{h} - v(t) \text{ i.e., } x_n = x_{n-1} + hv_n \quad (6.3)$$

where, the subscripts n and $n-1$ refer to time instants n and $n-1$ respectively. Figure 6.1 is an illustration of the BE integration method. In the following sub-section, the constitutive relationships which relate the through and across variables for an element are combined with the BE integration rule to obtain the linear equations which are solved at every time instant during transient analysis. Inductors and capacitors in the electrical domain as well as, springs, masses and dampers in the mechanical domain are used as examples.

6.2.2 Time Discretization of Components

Analogy Between MEMS and Circuit elements is used in order to explain the concept of time-discretization. Using the BE rule, the instantaneous I-V relationship for an inductor is written as:

$$\left(V = L \frac{di}{dt} \right) \Rightarrow \left(i_n = \left(\frac{h}{L} \right) V_n + i_{n-1} \right) \quad (6.4)$$

Similarly for a capacitor

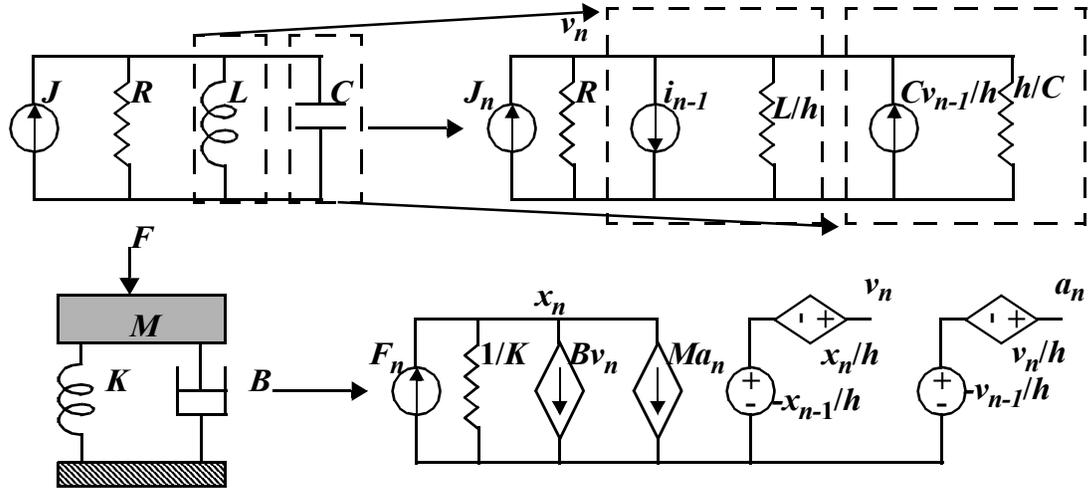


FIGURE 6.2. Circuit interpretation of Backward Euler integration rule for (a) Parallel RLC network (b) Mechanical spring-mass-damper system modeled by use of two additional states to hold v_n and a_n

$$\left(i = C \frac{dV}{dt}\right) \Rightarrow \left(i_n = \left(\frac{C}{h}\right) V_n - \frac{C V_{n-1}}{h}\right) \quad (6.5)$$

Note that in both the cases, the current is written in terms of the instantaneous voltage, giving a conductance term, and voltage/current obtained as a solution of the previous time step. This is because the system matrix is commonly built on the basis of Modified Nodal Analysis (MNA) in most circuit simulators [89]. Figure 6.2 shows the circuit interpretation at time instant n during transient analysis using the BE integration rule for a parallel RLC network and a mechanical second order system governed by the equation: $F = Kx + B\dot{x} + M\ddot{x}$. In this implementation of the second-order system, two additional states are used to hold v_n and a_n . Elastic, viscous and inertial elements are discretized as described above in the remainder of this chapter. A brief explanation of the transient analysis procedure using MNA is given below.

In Modified Nodal Analysis, the voltages at the nodes of the circuit are chosen as the independent state variables. During transient analysis, the node voltages are solved for at each time instant. It should be noted that the BE integration method is equivalent to a first order time polynomial approximation of the solution. The error in this approximation, referred to as the *Local Truncation Error*, is estimated to be of one order (i.e., quadratic for the BE method) higher than the approximation itself [89]. The simulator chooses the time-

step h to reduce the LTE to be less than a user-defined value. For simulations with high accuracy the SpectreTM simulator, sets the LTE to be one-tenth of the absolute tolerance (*abstol*) specified for convergence of the non-linear equation solving for that discipline. The concept of LTE is explained below for a node belonging to the mechanical discipline and defined to have an *acceleration-force* discipline. Assuming that the displacement (x) which corresponds to the acceleration (a) at the node is sinusoidal, the LTE using the BE integration method is estimated to be of the order of:

$$O(\text{LTE}) \sim \left(h^2 \frac{d^2 a}{dt^2} = h^2 \omega^4 x_0 \sin(\omega t) \right) \quad (6.6)$$

where, h is the time-step, ω is the radian frequency of the sinusoidal waveform and x_0 is the amplitude of the sinusoidal displacement corresponding to the acceleration at the node. Assuming typical values of $\omega = 10^4$ (*rad/s*) and $x_0 = 1 \mu m$, maximum values of the LTE occur when $\sin(\omega t) = 1$:

$$O(\text{LTE}) \sim h^2 10^{10} (m/s^2) \quad (6.7)$$

To establish the significance of the LTE, it is assumed for now that the *abstol* for the node is defined to the same as for other displacements. The maximum *abstol* for the displacements has to be less than $1 \mu m$ for the displacements to be accurate. Then, from (6.7) it is seen that the time-step can at most be only be 10 ns. It should also be noted that for lower frequency displacements, the same LTE setting allows larger time-steps. Thus, it is seen that the acceleration, derived from a time-dependent displacement, limits the time-steps, if the same *discipline* is used to define both the displacement and acceleration nodes. In general higher-order time-derivatives will limit the time-steps severely at higher frequencies, if same *discipline* are used to define the original signal as well as nodes carrying the time-derivatives. This issue will be revisited during the discussion of results.

6.3 Model Formulation

In this section two issues regarding the implementation of MEMS in HDL models are discussed. The first issue arises from the presence of multiple physical domains in the

models, such as translational, rotational, and electrical. The second issue is related to the presence, in each physical domain, of time-derivatives such as velocity and acceleration in the mechanical translational domain. The first sub-section addresses the issue of combining multiple domains together. The second sub-section investigates implementation of time-derivatives by exploring several alternatives.

6.3.1 Multi-domain Simulation

In multi-domain simulation, implementation of behavioral models without insight into the simulation matrix can lead to ill-conditioned matrices. For example, natural division of nodes into mechanical and electrical arises during HDL implementation of MEMS. Further sub-division of the mechanical nodes into displacements and rotations is also natural. Such classification is not only desired for clarity, but is also imperative to ensure well-conditioned simulation matrices.

Ill-conditioning of the simulation matrix arises due to the widely different numerical ranges in which the various physical domains of interest in MEMS are located. Typical values of through and across variables for electrical, mechanical and thermal disciplines in MEMS are shown in Table 6.1. While the ranges for the through and across variables themselves indicate a range of about 10 orders of magnitude, they are only part of the complete picture. The condition number of the simulation matrix is significantly impacted by the diagonal elements, i.e., the “conductance” entries in the MNA matrix, which can be loosely considered as the ratio of the through variable to the across variable. As can be seen in Table 6.1, there is a considerable spread in the ratios. Considering only the rotation

Table 6.1 Typical ranges for various physical domains in MEMS

<i>through-across</i> discipline	through variable			across variable			typical ratio	
	min	max	unit	min	max	unit		unit
current-voltage	10^{-10}	10^{-2}	A	10^{-6}	10^2	V	10^{-4}	Mhos
force-displacement	10^{-12}	10^{-6}	N	10^{-12}	10^{-6}	m	1	N/m
moment-rotation	10^{-16}	10^{-10}	N-m	10^{-8}	10^{-2}	radians	10^{-8}	N-m
heat flow-temperature	10^{-6}	10^{-2}	W	0.1	1000	K	10^{-5}	W/K

$$\begin{array}{c}
 Fx \\
 Fy \\
 Mz
 \end{array}
 \begin{bmatrix}
 \delta x & \delta y & \delta \phi \\
 \frac{EA}{l} & 0 & 0 \\
 0 & \frac{12EI_z}{l^3} & \frac{6EI_z}{l^2} \\
 0 & \frac{6EI_z}{l^2} & \frac{4EI_z}{l}
 \end{bmatrix}
 \begin{array}{l}
 \text{(a) No scaling} \\
 \text{(b) Scaling of } \delta\phi \text{ by } 1e-6
 \end{array}
 \begin{bmatrix}
 6600 & 0 & 0 \\
 0 & 2.64 & 1.32 \times 10^{-4} \\
 0 & 1.32 \times 10^{-4} & 8.8 \times 10^{-9}
 \end{bmatrix}
 \begin{bmatrix}
 6600 & 0 & 0 \\
 0 & 2.64 & 132 \\
 0 & 132 & 8800
 \end{bmatrix}$$

FIGURE 6.3. In-plane stiffness matrix for a beam [61]. Beam with *length* = 100 μm , *width* = 2 μm , *thickness* = 2 μm and Young's Modulus $E = 165 \text{ GPa}$. A is the cross-section area and I is the moment of inertia. (a) The large span of the diagonal elements of the stiffness matrix is evident. (b) The stiffness matrix after scaling the rotational discipline by 10^{-6} has much smaller condition number.

and displacement domains, the wide variation of the diagonal matrix elements is illustrated in Figure 6.3 by use of the in-plane stiffness matrix for a single cantilever beam. The cantilever beam has width and thickness of $2\mu\text{m}$, and a length of $100\mu\text{m}$. In-plane stiffness matrix of the beam is shown with and without scaling of the rotational domain. The condition number of the matrix is the ratio of the largest to the smallest eigen value of the matrix and is considered to be a measure of the numerical precision to which the inverse of the matrix can be computed [90]. It is evident that the resulting matrix is better conditioned (smaller condition number) with scaling. The scaling factor of 10^{-6} for the rotational domain was suggested prior to this work, and has been found to be necessary for convergence of behavioral simulations using NODAS. Therefore, the scaling factor of 10^{-6} is used in all the simulations in this thesis.

6.3.2 Implementation of Time-derivatives

Five different behavioral model formulations of the second order mechanical system shown in Figure 6.2 are studied. Broadly, the five formulations can be classified into two groups, one using *displacement* as the *across* variable and the other using *velocity* as the *across* variable, *force* being the *through* variable in all the cases. The latter bears a direct

analogy to traditional electrical circuit simulation which uses *current* as the *through* variable and *voltage* as the *across* variable.

VerilogA code, equivalent circuit and matrix representations for transient analysis with BE are shown for each case. In the equivalent circuit representations, derivatives and integrals are shown as they are and are not expanded into their BE forms to maintain compactness. Additional state variables explicitly defined and used by the model developer are listed in the beginning of the code. In some cases, it has been seen that the simulator inserts additional states, some of which have trivial solutions (i.e., being exactly equal to an existing state variable). The non-trivial states inserted by the simulator are also included in the equivalent circuit and the matrix representations. The equivalent circuits represent the equations solved by the simulator at each time step. They are composed of conductances which enter the diagonal elements of the matrix and voltage controlled current sources which contribute to the off-diagonal terms in the transient analysis matrix.

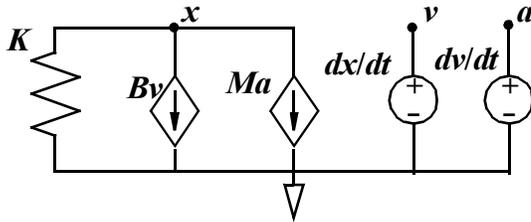
6.3.3 Displacement as across variable

Implementation x1: Additional states (v , a) are used to hold the velocity and acceleration. In addition to the elastic force modeled as a conductance, two voltage-controlled current sources corresponding to the damping and inertial forces also contribute to the *force* flowing through node x . In the matrix, off-diagonal elements ($1/h$) become large when the time-step h becomes small.

VerilogA

```
kinematic v, a;
Pos(v) <+ ddt(Pos(x));
Pos(a) <+ ddt(Pos(v));
F(x) <+ - M*Pos(a)
        - B*Pos(v)
        - K*Pos(x);
```

Equivalent circuit



Matrix

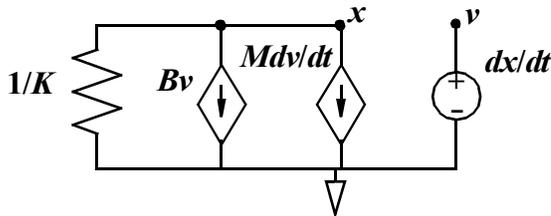
$$\begin{bmatrix} K & B & M \\ \frac{1}{h} & -1 & 0 \\ 0 & \frac{1}{h} & -1 \end{bmatrix} \begin{bmatrix} x_n \\ v_n \\ a_n \end{bmatrix} = \begin{bmatrix} F_n \\ \frac{x_{n-1}}{h} \\ \frac{v_{n-1}}{h} \end{bmatrix}$$

Implementation x2: One additional state (v) is used to hold velocity, leading to a more compact matrix.

VerilogA

```
kinematic v;
Pos(v) <+ ddt(Pos(x));
F(x) <+ - M*ddt(Pos(v))
      - B*Pos(v)
      - K*Pos(x);
```

Equivalent circuit



Matrix

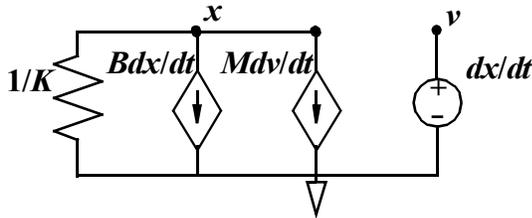
$$\begin{bmatrix} K & B + \frac{M}{h} \\ \frac{1}{h} & -1 \end{bmatrix} \begin{bmatrix} x_n \\ v_n \end{bmatrix} = \begin{bmatrix} F_n + \frac{Mv_{n-1}}{h} \\ \frac{x_{n-1}}{h} \end{bmatrix}$$

Implementation x3: One additional state (v) is used to hold the velocity. This implementation differs from implementation 2 only in that Bdx/dt is used instead of Bv . The damping and inertia terms occur in different elements of the matrix.

VerilogA

```
kinematic v;
Pos(v) <+ ddt(Pos(x));
F(x) <+ - M*ddt(Pos(v))
      - B*ddt(Pos(x))
      - K*Pos(x);
```

Equivalent circuit



Matrix

$$\begin{bmatrix} K + \frac{B}{h} & \frac{M}{h} \\ \frac{1}{h} & -1 \end{bmatrix} \begin{bmatrix} x_n \\ v_n \end{bmatrix} = \begin{bmatrix} F_n + \frac{Bx_{n-1}}{h} + \frac{Mv_{n-1}}{h} \\ \frac{x_{n-1}}{h} \end{bmatrix}$$

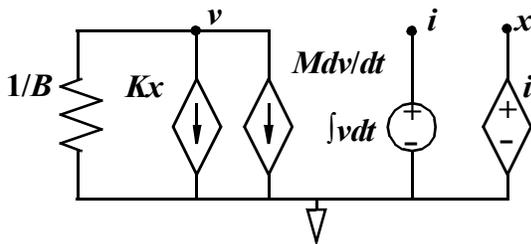
6.3.4 Velocity as across variable

Implementation v1: One extra state (x) is explicitly used to hold position (obtained by integrating velocity). Moreover, the simulator inserts an additional state (i) to hold the integral of velocity.

VerilogA

```
kinematic x;
Pos(x) <+ idt(Pos(v), 0);
F(v) <+ - M*ddt(Pos(v))
      - B*v;
      - K*Pos(x);
```

Equivalent circuit



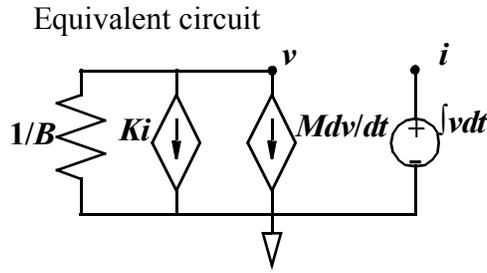
Matrix

$$\begin{bmatrix} B + \frac{M}{h} & K & 0 \\ 0 & 1 & -1 \\ h & 0 & -1 \end{bmatrix} \begin{bmatrix} v_n \\ x_n \\ i_n \end{bmatrix} = \begin{bmatrix} F_n + \frac{Mv_{n-1}}{h} \\ 0 \\ i_{n-1} \end{bmatrix}$$

Implementation v2: No explicit additional states are used in the VerilogA code. However, the simulator inserts an additional state (i) to hold the integral of velocity. Post-processing of the velocity solution is needed in order to obtain the displacement.

VerilogA

```
F(v) <+ - M*ddt(Pos(v))
      - B*v;
      - K*idt(Pos(v), 0);
```



Matrix

$$\begin{bmatrix} B + \frac{M}{h} & K \\ h & -1 \end{bmatrix} \begin{bmatrix} v_n \\ i_n \end{bmatrix} = \begin{bmatrix} F_n + \frac{Mv_{n-1}}{h} \\ i_{n-1} \end{bmatrix}$$

6.4 Simulation Results

The z-axis CMOS-MEMS gyroscope introduced in earlier chapters [10], is used as the benchmark for simulations. Unlike accelerometers which exhibit very small displacements during operation, the drive displacements in microgyroscopes is of the order of a few micrometers. Furthermore, the drive displacements occur at resonance, thereby implying that the inherent damping in the system is small for a high-Q (i.e., $Q \sim 50$) system. Under such conditions, the convergence properties of the behavioral models are crucial.

For simulations to test convergence properties, the comb-drives are removed from the schematic in order to eliminate non-linear physical effects. Then, the gyroscope can be modeled as a linear nested spring-mass-damper system. By reducing the problem to a linear system, convergence problems arising from device non-linearity are eliminated. Thereby, focus is maintained on the mathematical representation of the system and its relation to convergence and speed of simulation. Simulations of the gyroscope were done using the five different implementations. They are abbreviated as x1, x2, x3, v1 and v2. A $1 \mu\text{N}$ sinusoidal force was applied and transient analysis was done from 0 to 40 ms using the SpectreS simulator from Cadence, version 4.43 [87]. The results of simulation are summarized in Table 6.2.

Table 6.2 Comparison of five implementations

Type	Converged	Correct	Time (min.)	No. of Eqns	Time-steps
x1	No	NA	NA	2751	NA
x2	Yes	Yes	180	1809	68811
x3	Yes	Yes	110	1809	59875

Table 6.2 Comparison of five implementations

Type	Converged	Correct	Time (min.)	No. of Eqns	Time-steps
v1	Yes	Yes	133	2240	60023
v2	Yes	Yes	134	1556	60021

All the implementations, other than x1, converged. It should be noted that these simulations were performed with the acceleration and velocity nodes having the same *discipline* and therefore, the same LTE as the displacement nodes. Therefore, the simulator is forced to take smaller time-steps to meet the LTE condition. The condition number for the matrix for a single spring-mass-damper goes asymptotically as $1/(M.h)$ for x1 and as $(1/M)$ for x2, x3, v1 and v2. This explains why x1 does not converge for the gyroscope simulation. Though the number of time-steps taken by v1 and v2 is nearly the same as those taken by x3, the overall time taken is larger. This is probably due to the fact that multiple iterations are required for each time-step since, the initial time-step attempted by the simulator is not likely to satisfy the LTE criterion.

From the above simulations, it can be concluded that the use of additional nodes to hold the acceleration and velocity states can be harmful to the convergence of the simulations if, the acceleration and velocity states are constrained to have the same LTE as the displacement states. Alternatively, they can be defined to have a different *discipline* with independent LTE settings as shown below. In the implementation shown below, the LTE

VerilogA

```

kinematic_v v;
kinematic_a a;
Vel (v) <+ ddt (Pos (x) ) ;
Acc (a) <+ ddt (Vel (v) ) ;
F (x) <+ - M*Acc (a)
          - B*Vel (v)
          - K*Pos (x) ;

```

for the acceleration and velocity states (defined using the `kinematic_a` and the `kinematic_v` statements respectively) can be defined to be 10^8 and 10^4 times the LTE of the displacement state according the analysis done in Section 6.2.2. It was found that the simulations using this implementation also converged. This is to be expected, because the LTE constraints have been relaxed by a few orders of magnitude. Therefore,

the time-steps can be significantly larger than the x1 implementation, thereby preventing the worsening of the condition number of the simulation matrix.

It should be noted that the 10^8 and 10^4 numbers arise due to the frequency of interest for the gyroscope (~ 10 kHz). Since the frequency range of MEMS sensors range from near DC (~ 10 Hz in low bandwidth, high resolution accelerometers) to hundreds of kHz (in band-pass filters). The LTE setting may have to be changed depending upon the device being simulated. This requires a certain amount of numerical expertise on part of the MEMS designer. In contrast, the x3 implementation, shows robustness even with the LTE settings being severely constrained and is, therefore, the preferred implementation choice in this thesis.

6.5 Model Implementation Example: Squeeze-film Damping

In this section, two VerilogA implementations of a squeeze-film damping models are contrasted with respect to their convergence properties. This exercise is intended to convey the overall guidelines deduced from the simulation experiments described above, which include minimizing the number of off-diagonal elements and using appropriate *disciplines* for states. This is done by bringing out the equivalent circuit underlying the two behavioral models.

A semi-empirical compact model for squeeze-film damping was proposed and implemented in VerilogA by Vemuri et al. [91]. The initially proposed damping model is constructed as an electrical circuit with a number of parallel branches, each branch consisting of a resistor and an inductor in series (Figure 6.4). Dissipative damping between the two nodes \mathbf{x}_1 and \mathbf{x}_2 is modeled by a resistor and the non-dissipative elasticity is modeled by the inductor. It should be noted that the resistors can be replaced by dampers and the inductors by springs in an equivalent mechanical representation.

The VerilogA code for the two implementations are given in Figure 6.5 and Figure 6.6 respectively. The number of controlled or dependent sources in the equivalent circuits can be considered to be an approximate estimate of the number and location of the off-diagonal elements in the simulation matrix. During simulation it is seen that the simulator inserts additional states in the first implementation, which have not been shown in the

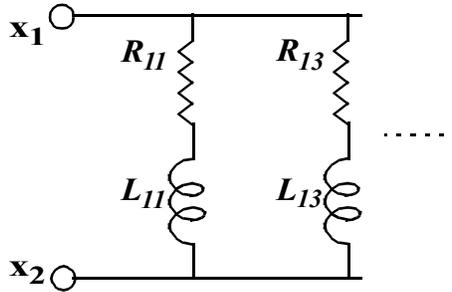


FIGURE 6.4. Squeeze-film damping modeled by equivalent resistors and inductors. Only the first two R-L branches are shown in the figure. More accurate models need more number of branches.

equivalent circuit representation. Those additional states probably lead to larger number of off-diagonal terms than is seen in the equivalent circuit. This also causes the first implementation to have severe convergence problems which completely disappear when the second implementation is used. The second implementation is completely in the mechanical domain, which is better suited to set the tolerances for convergence. Also, the off-diagonal elements in the second implementation are more localized (e.g., nodes **n11** and **vn11**

```

electrical vel;
vel = ddt(Pos(x1, x2));

V(n11) <+ I(t11, bot)*L11;
I(t11, bot) : ddt(V(n11)) == vel - R11*I(t11, bot);

V(n13) <+ I(t13, bot)*L13;
I(t13, bot) : ddt(V(n13)) == vel - R13*I(t13, bot);

```

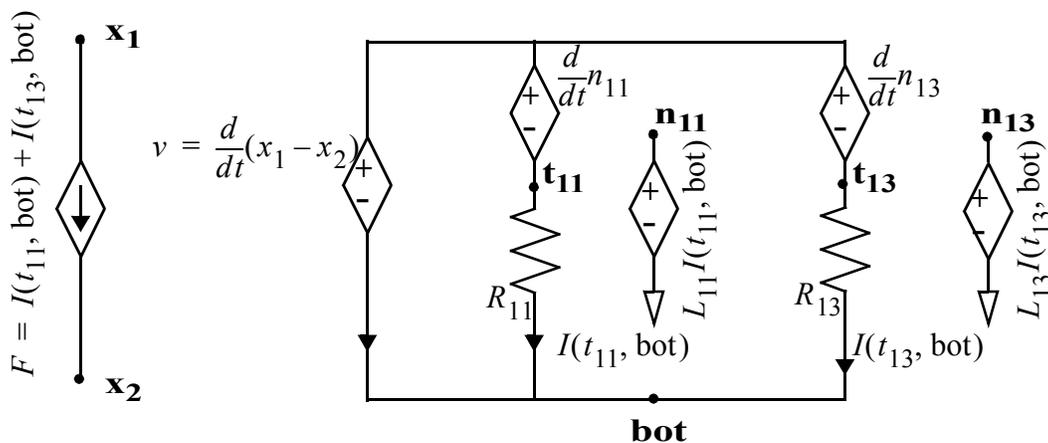


FIGURE 6.5. VerilogA code and equivalent circuit for first implementation of the squeeze-film damping model. Only two of the RL branches are shown. The actual circuit interpretation by the simulator is not exactly known but is probably more complex because it is observed that additional states are implicitly introduced during simulation.

```

kinematic disp_node;
Pos(disp_node) <+ disp;

F(disp_node, n11) <+ (1/L11)*Pos(disp_node,n11);
Pos(vn11) <+ ddt(Pos(n11));
F(n11) <+ Pos(vn11)/R11;

F(disp_node, n13) <+ (1/L13)*Pos(disp_node,n13);
Pos(vn13) <+ ddt(Pos(n13));
F(n13) <+ Pos(vn13)/R13;

```

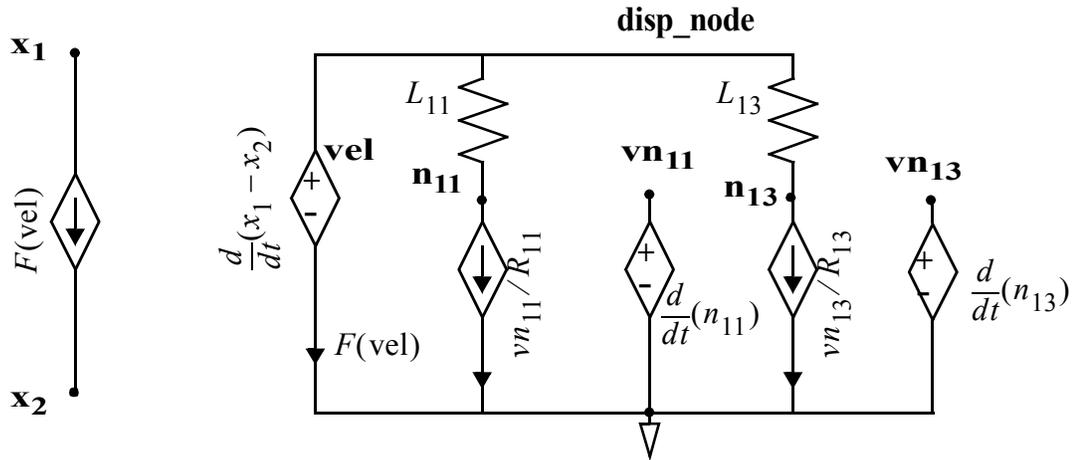


FIGURE 6.6. VerilogA code and equivalent circuit for second implementation of the squeeze-film damping model. The controlled sources inside the damping model are more locally distributed compared to the first implementation. This implementation shows better convergence properties.

are directly linked, in contrast with nodes **n11** and **t11** in the first implementation). In fact the nodes **n11** and **vn11** are simply implementing the equivalent of a capacitor in the mechanical domain since the force (current) through node **n11** is proportional to the derivative of the position (voltage) at that node. The second implementation also allows better intuitive understanding of the damping model compared to the first one. We have to keep in mind that the non-dissipative elements L_{11} , L_{13} appear as “resistors” in the second implementation, because the *across* variable is position and not velocity. Therefore, the capacitor-like elements model the dissipative components.

6.6 Summary

In this chapter three issues which impact the convergence and simulation time in MEMS behavioral simulations were addressed. First, velocity and displacement were compared for the choice of the across variable in nodal simulation. The through variable is force. Second, three state space implementations of displacement as across variable were

compared. Finally, relative scaling of a domain (the rotational domain, in this case) in order to improve convergence properties was considered. A minimal equation matrix representation with low condition number using displacement as across variable and scaling of the rotational domain gives the best convergence and simulation time.

It is seen that the implementation of the analog HDL encoding of the differential equations describing the element behavior directly impacts the convergence and simulation speed in transient analysis. There is no significant speed advantage of using *velocity* as the *across* variable. Therefore, keeping in mind ease of use, *displacement* as *across* variable is a better choice. On the basis of the simulations and analysis performed, the following guidelines are presented:

1. Additional states ($a = \dot{v}$, $v = \dot{x}$) to hold derivatives lead to bigger simulation matrices. They introduce large off-diagonal terms during the transient analysis and lead to ill-conditioned matrices if the local truncation error (LTE) settings are not carefully optimized. Therefore, care must be taken when using additional states.
2. Appropriate scaling must be used when different domains are combined together so that the composite nodal analysis matrix remains well-conditioned.
3. Between two equivalent implementations, the one with lower number of off-diagonal coupling terms leads to better convergence properties.

Chapter 7. Analysis and Simulation of a CMOS-MEMS Gyroscope

7.1 Introduction

In the previous chapters models for elastic and electrostatic domains have been presented and gyroscope simulation problems have been addressed. In this chapter, detailed analyses of gyroscope non-idealities are done. The non-idealities are quantified using behavioral simulation of schematics composed of the previously described models. Elastic cross-axis coupling theory, detailed in Chapter 3, is used extensively for understanding the gyroscope.

The working of the microgyroscope and associated typical magnitudes of displacement are described in Chapter 1. A brief recap is given here for convenience. Microgyroscopes can be functionally decomposed into a proof-mass, driving and sensing electromechanical comb-drives and suspension springs made up of beams as shown in Figure 7.1 [2]. Voltage applied across the driving comb forces the proof-mass into oscillation. When placed in a rotational field, the Coriolis force induces a vibration in a direction orthogonal to the driven oscillations. The induced vibration is proportional to the angular

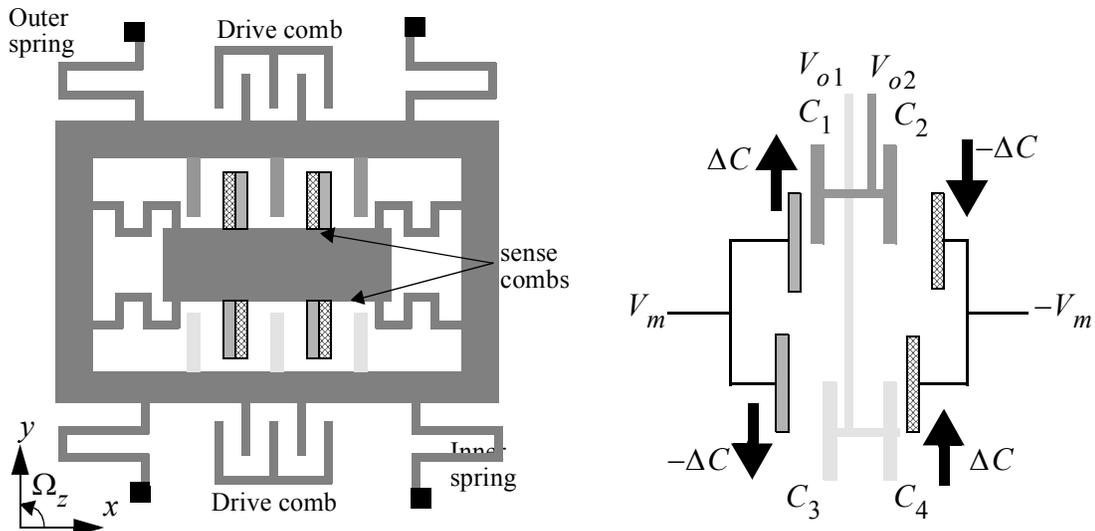


FIGURE 7.1. (a) Nested gyroscope design showing the drive and sense combs, the outer and inner springs, the input axis, the direction of driven vibrations and the direction of Coriolis-force induced (sense) vibrations (b) Sense capacitance bridge for movement of inner proof-mass in positive x axis

rate and is sensed capacitively. As the Coriolis force induced vibrations are much smaller (~ 1 ppm) compared to the driven vibrations, the proof-mass displacements can be as small as a few picometers. Capacitive sensing of these picometer-scale displacements requires tight integration of circuits with the Coriolis force transducer, and can be accomplished in a CMOS-MEMS process [7][10][30]. Furthermore, as the Coriolis force is only one of the weak coupling forces in the gyroscope, such devices are prone to imperfections such as input offset, linear acceleration sensitivity, vibration sensitivity and cross-axis sensitivity, making microgyroscopes difficult to design. Therefore, commercialization attempts have met with limited and much-delayed success [5][28].

Non-idealities in microgyroscopes such as offsets and sensitivity to non-rotational inputs are poorly understood. Optimal design to reject such non-idealities is currently hampered by the limitations of the available simulation methodologies which do not model such effects. Though it is commonly acknowledged [5] that the coupling of the drive motion to the sense mode needs to be as small as a few ppm, there is no comprehensive study in public literature of drive motion coupling, external accelerations and cross-axis rotations. Quadrature error arising from elastic cross-coupling has been considered in a few studies. However, in-phase coupling may also arise in gyroscope designs with intentionally mismatched drive and sense modes.

Behavioral simulations of gyroscopes have been reported in the past. Non-idealities in the gyroscope simulations using circuit-level schematics reported in [51] are limited to the effect of drive coupling onto the sense mode. Simulations presented in [52] are limited to a theoretical motion analysis arising from elastic and viscous coupling. Gyroscope simulations previously done using NODAS include nominal sensitivity and distortion effects of centripetal forces and linear accelerations [33]. Intra-die thickness variation for a bulk-micromachined gyroscope has been considered in [31]. Robust design techniques to reject width variations across different chips [92] cannot compensate for width mismatch within a device. Causes of input offset and coupling of linear acceleration to the output and cross-axis sensitivity are not analyzed in any of the above studies.

Geometrical asymmetries in a microgyroscope may arise due to random manufacturing variations. In this thesis the focus is on variations in beam width and comb gap across

the gyroscope and metal mask misalignment effects. Data on material property variations is even less known than geometrical variations and are not considered in this thesis. This is the first publicly known attempt, to understand the effect of intra-die variations on micro-gyroscope non-idealities. MEMS circuit-level simulation is employed to correlate gyro performance measures such as zero rate output (ZRO), linear acceleration sensitivity (S_a), vibration sensitivity, (S_{a2}) and cross-axis sensitivity (S_{ca}) to elastic and electrostatic asymmetries in the gyroscope. The correlations are then used to develop pointers for robust design.

The rest of the chapter is organized as follows. First, the CMOS-MEMS gyroscope and the circuit-level representation used for simulation are described. A detailed derivation of the transduction equation of the CMOS-MEMS gyroscope is detailed. This is followed by a discussion on the disadvantages of asymmetric drive of the gyroscope. The subsequent sections describe the analyses and simulations for gyroscope non-idealities: Zero Rate Output, linear acceleration sensitivity, vibration sensitivity and cross-axis sensitivity. The considered sources of non-idealities are beam width and comb gap variations, and mask misalignment. Analyses of non-idealities are supported by behavioral simulations of each individual variation as well as results of Monte-Carlo simulations. Finally, a summary of pointers and trade-offs to be considered for robust design is presented.

7.2 Gyroscope Description and Circuit-level Representation

7.2.1 Gyroscope Description

The SEM of a CMOS-MEMS gyroscope [10] is shown in Figure 7.2(a). The NODAS schematic representation of the gyroscope is shown in Figure 7.2(b). The atomic-elements which are used in the schematic are rigid *plates*, flexible *beams* and attachment points called *anchors*. Electrostatic *comb* elements are also used to implement actuation and sensing. The gyroscope, referred to as a “nested-gyroscope”, consists of an inner accelerometer nested inside an outer resonator. The entire gyroscope is suspended from a curl-matching frame which is anchored to the substrate. The inner accelerometer and the outer resonator are each composed of four compliant springs and a number of interconnected rigid plates. The rigid plates in the outer resonator form a rigid frame from which the inner

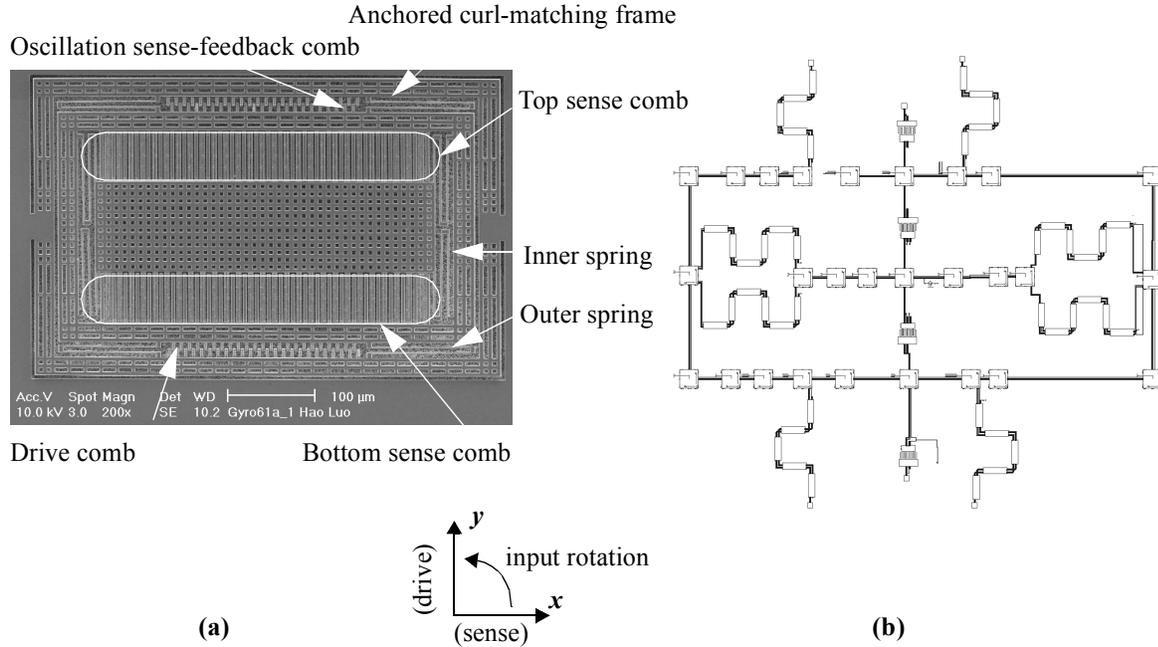


FIGURE 7.2. (a) SEM of the nested-gyroscope (b) Corresponding NODAS schematic obtained through layout extraction.

accelerometer is suspended. The rigid frame is modeled by a number of *plates* forming an outer ring. These *plates* are connected to the curl-matching frame using four identical chains of *beams* which form the outer springs. In the inner accelerometer, the central proof-mass is modeled by a row of seven *plates*. The *plates* on the extreme left and extreme right are each connected through two identical chains of *beams* (inner springs) to the rigid frame.

Two linear combs, one at the top and one at the bottom, each with one set of anchored (i.e., attached to the curl-matching frame) fingers and the other set attached to the rigid outer-resonator frame, produce electrostatic force in the drive (y) direction, when a voltage is applied across the two sets of fingers. This electrostatic force drives the rigid outer-resonator frame and the inner accelerometer into resonant oscillations in the y direction. When attached to a rotating frame, Coriolis force produces oscillations of the inner central plate relative to the outer resonator in the x direction. The Coriolis force induced oscillations are capacitively picked up by two differential *combs*, one each at the top and bottom, with one set of fingers attached to the rigid outer-resonator frame and the other set of fingers attached to the central *plate* of the inner accelerometer. The two differential combs

are, ideally, only sensitive to relative motion between the central proof-mass and the outer-resonator rigid frame in the sense (x) direction.

A small note with regard to the actuation combs is in order here. The original design intent with the two linear combs was to use one for actuation and the other to sense the driven oscillations [10][34]. The overwhelming undesirable effects in an asymmetrical (only one actuation comb) driven gyroscope are described in Section 7.4. To eliminate these effects from the following analyses in this thesis, both the linear combs are considered as actuation combs only.

The ideal gyroscope has a frequency domain output as shown in Figure 7.3(a). The final gyroscope output is obtained by demodulation as shown in Figure 7.3(b) [10]. The modulation voltage V_m is assumed to be DC since, in this thesis the analysis is restricted to the gyroscope. The circuits used for amplification and demodulation contribute to these

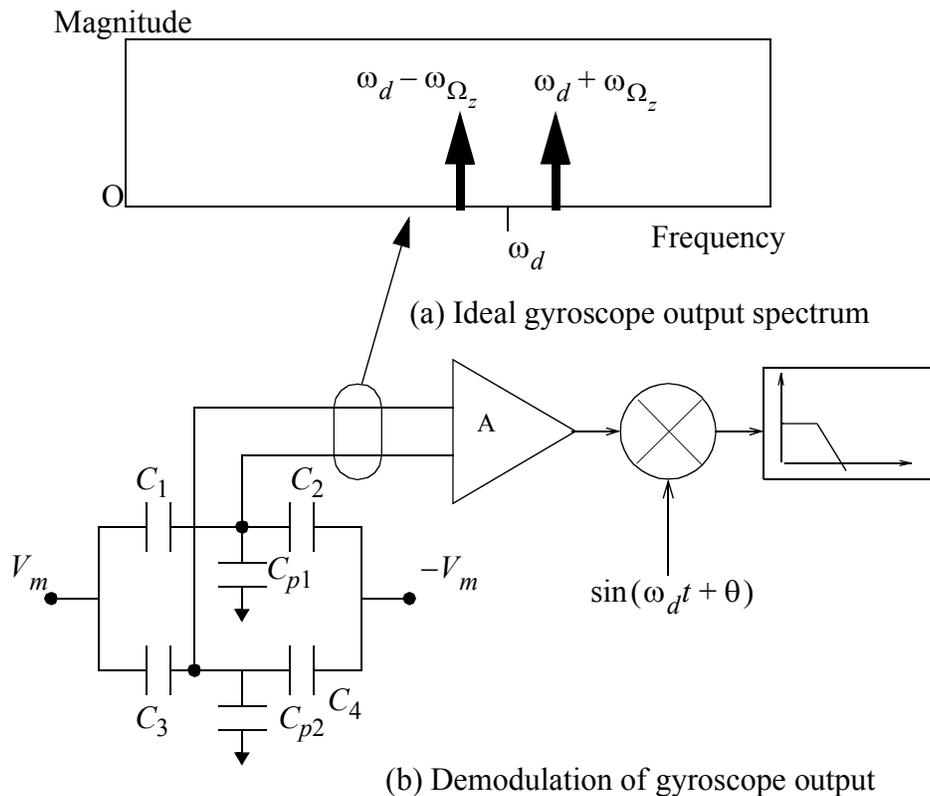


FIGURE 7.3. (a) Output spectrum of an ideal gyroscope for an input sinusoidal rotation rate (b) sense schematic showing demodulation of gyroscope capacitance bridge output to yield voltage proportional to input rate. The angle θ needs to be adjusted to maximize sensitivity and minimize offsets

non-idealities through offsets, differential to common-mode conversion, phase errors, carrier feed-through and other well-understood circuit phenomena and are therefore, not being considered here. Having studied the physical structure of the gyroscope, the system level abstraction of the gyroscope is now described.

7.2.2 Gyroscope Parameters

At the system-level, gyroscopes are known only by their main performance characteristics. In this sub-section, definitions for common gyroscope parameters, which are particularly relevant to this thesis, are given [8][9].

Sensitivity/Scale factor: The constant of proportionality between the input rotation rate and the output voltage is called the sensitivity of the gyroscope. The sensitivity may vary with the input rotation rate leading to non-linearity.

Resolution: The smallest detectable change in input rotation rate. The resolution is determined by the mechanical thermal noise and the electronic noise in the sensing circuits.

Zero Rate Output (ZRO)/Input offset/Bias: Input rate required to drive the output voltage to zero.

Bias Drift: The change in the *Zero Rate Output* or *Bias* over time.

Acceleration sensitivity: A linear acceleration applied to the gyroscope may result in an output voltage indistinguishable from that produced by an input rotation. Typically, gyroscopes show a linear as well as quadratic dependence on acceleration. The linear dependence on acceleration is called the *Acceleration Sensitivity*. The quadratic dependence on acceleration is referred to as *Vibration or Acceleration-squared Sensitivity*.

Cross-axis sensitivity: Output produced by an angular rotation about an axis orthogonal to the input axis of the gyroscope.

All the above performance characteristics depend on the operating conditions i.e., the ambient temperature and pressure, in addition to the gyroscope geometry. The resolution is limited by energy leakage processes such as viscous damping and resistive loss. The various sources of non-idealities in microgyroscopes can be broadly classified as shown in Figure 7.4. The sensor (including the nested resonators) as well as the electrostatic actuation and sensing combs) contributes to non-idealities arising from elastic, viscous or inertial coupling, from elastic and electrostatic non-linearities, from electrostatic multi-

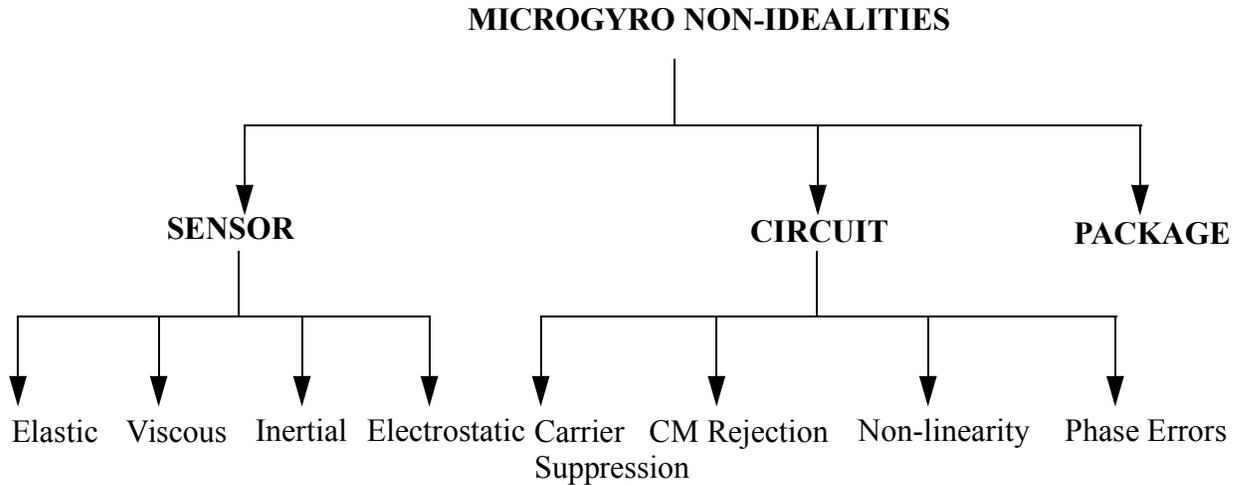


FIGURE 7.4. Classification of sources of microgyroscope non-idealities

directional drive and sense. As mentioned earlier, the sense and demodulation electronics can contribute to non-idealities through lack of carrier suppression, common-mode to differential conversion, non-linearity leading to spurious side-bands and phase errors in demodulation leading to increased Zero Rate Output. Furthermore, packaging can exacerbate existing non-idealities by introducing additional stress gradients in the microstructure. Of all these, this chapter attempts to understand elastic and electrostatic causes of non-idealities. Viscous and inertial effects can be analyzed in a manner similar to that of elastic cross-axis coupling as described in Section 3.8. Circuit non-idealities have been well-characterized over the years and packaging effects can be modeled by adding onto the non-idealities modeled in this chapter. The Zero Rate Output (input offset), cross-axis sensitivity and acceleration sensitivity arising from geometrical asymmetries are the primary focus of this thesis.

7.2.3 Notation

The symbols used in this chapter for applied voltages, displacements, external accelerations and rotations, and geometrical and functional parameters of the gyroscope are

compiled in Table 7.1. Additional symbols are defined as and when they appear. For anal-

Table 7.1 Symbols used

Symbol	Description
V_m	Modulation voltage applied to sense combs
V_C	Gyro output voltage
V_d	Actuation voltage applied to drive combs
y_{do}	Drive displacement of outer resonator
y_{di}	Drive displacement of inner resonator
x_{co}	Relative displacement between outer-resonator and inner accelerometer in the x direction due to Coriolis force on outer-resonator frame
x_{ci}	Relative displacement between outer-resonator and inner accelerometer in the x direction due to Coriolis force on inner accelerometer plate
x_C	Coriolis force induced displacement in sense direction
y_{ds}	Relative displacement of differential sense combs in drive direction
x_{os}	Lateral offset in differential sense combs in sensing direction
$\Omega_x, \Omega_y, \Omega_z$	External applied angular rates equal to $1(^{\circ}/s) = (\pi/180)(rad/s)$
A_x, A_y, A_z	External applied linear accelerations = $1\text{ g} = 9.8\text{ (m/s}^2\text{)}$
o_{lpd}, g_d	Nominal overlap and gap in drive combs
o_{lps}, g_s	Nominal overlap and gap in sense combs
Δ	Relative mismatch between beam widths or gaps
Q_{ix}, Q_{ox}	Quality factors of the inner and outer resonator respectively in the x direction
Q_{iy}, Q_{oy}	Quality factors of the inner and outer resonator respectively in the y direction
Q_d	Quality factor of the drive mode ($Q_d \approx Q_{oy}$)
Q_s	Quality factor of the sense mode ($Q_s \approx Q_{ix}$)
ω_{ix}, ω_{ox}	Resonant frequencies of the inner and outer resonator respectively in the x direction
ω_{iy}, ω_{oy}	Resonant frequencies of the inner and outer resonator respectively in the y direction

Table 7.1 Symbols used

Symbol	Description
ω_d	Gyro drive frequency ($\omega_d \approx \omega_{oy}$)
ω_s	Gyro sense mode resonant frequency ($\omega_s \approx \omega_{ix}$)
$K_{\zeta\xi i}, K_{\zeta\xi o}$	Elements of the stiffness matrix for the inner and outer springs respectively. ζ, ξ are one of $x, y, z, \phi_x, \phi_y, \phi_z$
$B_{\zeta\xi i}, B_{\zeta\xi o}$	Elements of the damping matrix for the inner and outer resonators respectively. ζ, ξ are one of $x, y, z, \phi_x, \phi_y, \phi_z$
$M_{\zeta\xi i}, M_{\zeta\xi o}$	Elements of the inertia matrix for the inner and outer resonators respectively. ζ, ξ are one of $x, y, z, \phi_x, \phi_y, \phi_z$
M_o, M_i	Approximations: $M_o = M_{xxo} = M_{yyo} = M_{zzo}$ and $M_i = M_{xxi} = M_{yyi} = M_{zzi}$

ysis purposes the input angular rate and input external acceleration are considered to be constant as given in Table 7.1.

7.2.4 CMOS-MEMS Gyroscope Design Parameters

The important geometrical and functional parameters of the CMOS-MEMS gyroscope reported in [10], are listed in Table 7.2 and Table 7.3 respectively. The functional

Table 7.2 Geometrical parameters of the CMOS-MEMS gyroscope

Group	Parameter name	Symbol	Value	Units
Outer spring	Outer spring beam length	l_{bo}	110	μm
	Outer spring beam width	w_{bo}	1.8	μm
Inner spring	Inner spring beam length	l_{bi}	102	μm
	Inner spring beam width	w_{bi}	1.8	μm
Drive comb	finger length	l_{fd}	11.4	μm
	finger width	w_{fd}	2.7	μm
	overlap	o_{lpd}	3.3	μm
	gap	g_d	1.8	μm

Table 7.2 Geometrical parameters of the CMOS-MEMS gyroscope

Group	Parameter name	Symbol	Value	Units
	number of fingers	N_d	23	
Sense comb	finger length	l_{fs}	61.5	μm
	finger width (single finger attached to outer-resonator rigid frame)	w_{fs}	3.9	μm
	finger width (double finger attached to inner central plate)	w_{f2s}	5.7	μm
	overlap	o_{lps}	60	μm
	gap	g_s	1.8	μm
	number of fingers	N_s	21	
Inner plates	total length	l_{pi}	372.65	μm
	total width	w_{pi}	111.6	μm
	percentage holes	f_{pi}	20.87	%
Outer-resonator frame	overall length	l_{ro}	456	μm
	overall width	w_{ro}	283.2	μm
	frame width	w_{fr}	22.8	μm
	percentage holes	f_{ro}	28.4	%

Table 7.3 Functional parameters of the CMOS-MEMS gyroscope

Group	Name	Symbol	Value	Units
Outer spring	x stiffness	K_{xxo}	235	N/m
	y stiffness	K_{yyo}	2.83	N/m
	z stiffness	K_{zzo}	16.6	N/m
Inner spring	x stiffness	K_{xxi}	2.91	N/m
	y stiffness	K_{yyi}	95.2	N/m
	z stiffness	K_{zzi}	10.0	N/m
Outer-resonator frame + inner plate	mass	M_o	1.02	$nkg (\mu g)$
Inner plate	mass	M_i	0.594	$nkg (\mu g)$

Table 7.3 Functional parameters of the CMOS-MEMS gyroscope

Group	Name	Symbol	Value	Units
Outer resonator	x mode frequency	ω_{ox}	83820	$2\pi \text{ rad/s}$
	y mode frequency	ω_{oy}	8440	$2\pi \text{ rad/s}$
	z mode frequency	ω_{oz}	17110	$2\pi \text{ rad/s}$
	Quality factor outer frame, in y	Q_{oy}	80.9	1
Inner resonator	x mode frequency	ω_{ix}	11130	$2\pi \text{ rad/s}$
	y mode frequency	ω_{iy}	76440	$2\pi \text{ rad/s}$
	z mode frequency	ω_{iz}	25500	$2\pi \text{ rad/s}$
	Quality factor inner resonator, in x	Q_{ix}	11.6	1

design parameters are obtained by ac simulations of the gyroscope schematic using NODAS.

In this section, the context for the detailed analyses to follow has been set up by a description of the gyroscope, the important performance characteristics under consideration, notations used, and the geometrical, operational and functional quantities. In the next section, the analysis of the gyroscope begins with a derivation of the gyroscope transduction equation, i.e., the conversion from input angular rate to output voltage. The analysis for gyroscope sensitivity assumes that both the linear actuation combs are used to drive the gyroscope. The effects of using only one of the two linear actuation combs to drive the gyroscope are considered in Section 7.4. The subsequent three sections analyze the Zero Rate Output, acceleration sensitivity and the cross-axis sensitivity in terms of the functional parameters in Table 7.3.

7.3 Gyroscope Sensitivity

The sensitivity of the gyroscope is derived in a step-by-step manner, listing the simplifying assumptions on the way. Simultaneously, the relative phases of the applied actuation voltage V_d , and the drive (y_{do}) and sense (x_C) displacements are also considered. It is important to understand the phase relationships so that non-idealities which appear in-

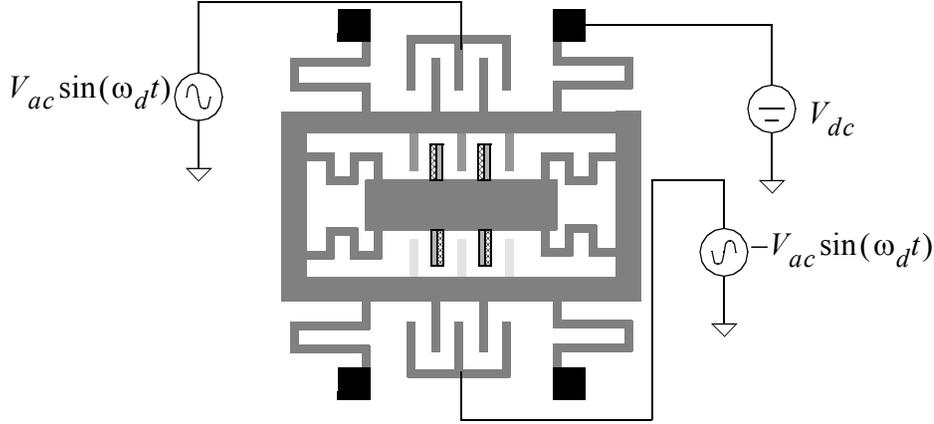


FIGURE 7.5. Anti-phase voltages applied to drive the gyroscope into oscillations.

phase with the sense signal and those which only appear in quadrature can be discriminated. Gyroscope transduction can be viewed as a succession of four steps:

1. Voltages applied to drive combs produce an electrostatic force $V_d \rightarrow F_y$
2. The outer resonator is set into oscillations by the applied electrostatic force $F_y \rightarrow y_{do}$
3. When placed in a rotational field the Coriolis force produces vibrations orthogonal to the driven oscillations $\Omega_z, y_{do} \rightarrow x_C$
4. The induced orthogonal oscillations are capacitively sensed leading to an output voltage $x_C \rightarrow V_C$

Equations for each of the above steps are derived below. In order to keep in mind the actual magnitudes of various quantities as observed in a gyroscope, the equations derived are also numerically evaluated at every step. Such evaluation also provides instant justification for simplifying assumptions.

As mentioned earlier, the analyses presented here assumes that both the linear combs are used for actuation. Anti-phase voltages applied to the two drive combs, as shown in Figure 7.1, set the outer resonator and the inner proof-mass into oscillation in the y direction. The forces in the top and bottom combs are respectively given as:

$$F_{yt} = N_d c_0 \epsilon_0 \frac{t}{g_d} (V_{dc}^2 + 2V_{dc} V_{ac} \sin(\omega_d t) + V_{ac}^2 \sin^2(\omega_d t)) \quad (7.1)$$

$$F_{yb} = N_d c_0 \varepsilon_0 \frac{t}{g_d} (V_{dc}^2 - 2V_{dc} V_{ac} \sin(\omega_d t) + V_{ac}^2 \sin(\omega_d t)^2) \quad (7.2)$$

where, the constant c_0 depends on the contribution of the fringing fields to the total force.

The total force is obtained by taking the difference of (7.1) and (7.2) as:

$$F_y = N_d c_0 \varepsilon_0 \frac{t}{g_d} 4V_{dc} V_{ac} \sin(\omega t) \quad (7.3)$$

For an applied sinusoidal voltage $V_{ac} = 1 V_{pp}$ and DC voltage (V_{dc}) 18 Vpp and assuming $c_0 = 1$ i.e., neglecting the force due to the fringe fields, we get $F_y = 39.5 nN$. Note that application of anti-phase voltages leads to symmetric drive wherein, the DC and first harmonic components are canceled out and the resulting force only contains the fundamental drive frequency. Usually, the oscillations are sensed, amplified and fed back to set up a feedback loop which leads to resonant oscillations of the outer resonator and the inner proof-mass at the fundamental y resonant mode of the gyroscope. At resonance, the displacement of the outer frame can be given as:

$$y_{do} = -j \frac{Q_d F_y}{K_{yyo}} \quad (7.4)$$

Note that complex number notation (j), which is commonly used to represent sinusoidal voltages, is used above. Using values from Table 7.3, $y_{do} = -j1.39 \mu m$. The $-j$ in the above equation indicates that the displacement lags behind the force by 90° , as expected, at resonance for a second-order system. For the transduction analysis it is assumed that the outer resonator of the gyroscope is driven in the y direction at constant amplitude y_{do} . Since the inner accelerometer is coupled to the outer resonator, through the inner springs with stiffness K_{yyi} in the y direction, the amplitude of the inner accelerometer is given as:

$$y_{di} = y_{do} \left(1 - \frac{M_{yyi} \omega_{oy}^2}{K_{yyi} - M_{yyi} \omega_{oy}^2 + jB_{yyi} \omega_{oy}} \right) = y_{do} \left(1 - \frac{\omega_{oy}^2}{\omega_{iy}^2 - \omega_{oy}^2 + j \frac{\omega_{iy} \omega_{oy}}{Q_{iy}}} \right) \quad (7.5)$$

$$y_{di} = y_{do} \left(1 - \frac{\eta_{iy}^2}{1 - \eta_{iy}^2 + j \frac{\eta_{iy}}{Q_{iy}}} \right) \quad (7.6)$$

where, the ratio $\eta_{iy} = \omega_{oy}/\omega_{iy}$ depends upon the stiffness K_{yyi} of the coupling spring between the two resonators. (7.6) can be derived intuitively by considering an external acceleration equal to $-\omega_{oy}^2 y_{do}$ applied to the accelerometer and obtaining the relative y displacement between the outer-resonator rigid frame and the inner accelerometer. Normally the designer ensures that $\eta_{iy} \ll 1$ and is typically of the order of 0.1. Therefore, to the first order, the drive displacement is approximated to be y_D given as:

$$y_{di} = y_{do} = y_D \quad (7.7)$$

In the presence of an angular rate Ω_z about the z axis, the inner rigid plate and the outer-resonator rigid frame experience a Coriolis force in the x direction respectively given as:

$$F_{co} = j2(M_o - M_i)\Omega_z\omega_{oy}y_D \text{ and } F_{ci} = j2M_i\Omega_z\omega_{oy}y_D \quad (7.8)$$

$M_o - M_i$ arises in the above equation because the M_o term encompasses the entire resonating mass during oscillation, and therefore, includes the mass M_i of the inner accelerometer also. Using (7.3), (7.4) and (7.8) the magnitudes of F_{co} and F_{ci} can be written as:

$$F_{co} = 2(M_o - M_i)\Omega_z\omega_{oy} \left(\frac{Q_d F_y}{K_{yyo}} \right) = \frac{2(M_o - M_i)\Omega_z\omega_{oy}Q_d}{K_{yyo}} \left(N_d c_0 \varepsilon_0 \frac{t}{g_d} (4V_{dc} V_{ac}) \right) \text{ and}$$

$$F_{ci} = 2M_i\Omega_z\omega_{oy} \left(\frac{Q_d F_y}{K_{yyo}} \right) = \frac{2M_i\Omega_z\omega_{oy}Q_d}{K_{yyo}} \left(N_d c_0 \varepsilon_0 \frac{t}{g_d} (4V_{dc} V_{ac}) \right) \quad (7.9)$$

Numerical magnitudes are computed as: $F_{co} = 1.09\text{pN}$ and $F_{ci} = 1.53\text{pN}$. The displacements due to F_{co} and F_{ci} can be computed by solving the simultaneous differential

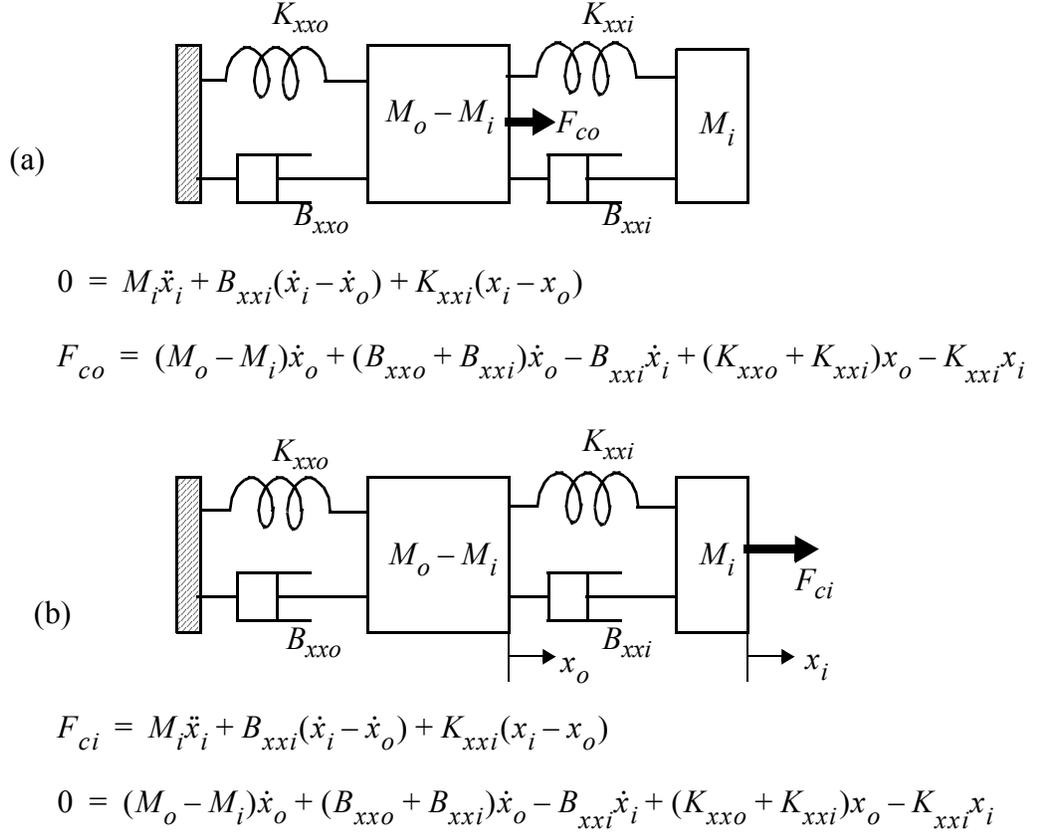


FIGURE 7.6. Nested resonator system and dynamical equations (a) when a force F_{co} is applied to the outer frame and (b) when a force F_{ci} is applied to the inner mass.

equations shown in Figure 7.6 in the frequency domain. Assuming that the resonant frequency of the outer frame in the x direction is much higher than the drive frequency and that the damping force $B_{xxo}\dot{x}_o$ is insignificant compared to the spring restoring force $K_{xxo}x_o$, the relative displacement between the outer and inner resonators due to F_{co} is:

$$x_{co} = -\frac{F_{co}Q_{ix}\eta_{ix}^2}{K_{xxi}(Q_{ix}\eta_{ix}^2 - Q_{ix}\gamma + Q_{ix}\eta_{ix}^2\gamma + j\eta_{ix}^3 - j\eta_{ix}\gamma)} = -\frac{F_{co}\Gamma(\eta_{ix})}{K_{xxi}\gamma_x} \quad (7.10)$$

where,

$\eta_{ix} = \omega_{oy}/\omega_{ix}$ is the ratio of the drive frequency to the resonant frequency of the inner resonator and $\gamma_x = K_{xxo}/K_{xxi}$ is the ratio of the stiffnesses of the outer and inner springs

in the x direction. The function $\Gamma(a)$, which will be used repeatedly in following analyses, is defined as:

$$\Gamma(a) = \frac{Q_{ix}a^2\gamma_x}{(Q_{ix}a^2 - Q_{ix}\gamma_x + Q_{ix}a^2\gamma_x + ja^3 - ja\gamma_x)}$$

which can be simplified for $a \approx 1$ and $\gamma_x \gg 1$ to:

$$\Gamma(a) = \frac{a^2}{1 - a^2 + ja/Q_{ix}} \quad (7.11)$$

For $\eta_{ix} = 0.755$ $\Gamma(\eta_{ix}) = -1.32 + 0.201j$. With the same assumptions as before, the relative displacement between the outer and the inner resonators due to F_{ci} is:

$$x_{ci} = -\frac{F_{ci}Q_{ix}\gamma_x}{K_{xxi}(Q_{ix}\eta_{ix}^2 - Q_{ix}\gamma_x + Q_{ix}\eta_{ix}^2\gamma_x + j\eta_{ix}^3 - j\eta_{ix}\gamma_x)} = -\frac{F_{ci}\Gamma(\eta_{ix})}{K_{xxi}\eta_{ix}^2} \quad (7.12)$$

$x_{ci} = 1.19 - 0.181j$ pm and $x_{co} = -0.0120 + 0.00181j$ pm. Comparing (7.10) and (7.12) it is seen that x_{co} and x_{ci} are exactly in phase with each other. The ratio of their magnitudes is:

$$\frac{x_{co}}{x_{ci}} = \frac{(M_o - M_i)\eta_{ix}^2}{M_i\gamma_x} \ll 1 \quad (7.13)$$

Note that in spite of F_{co} being larger than F_{ci} , the above ratio is much less than one. This is because $M_o - M_i$ and M_i are of the same order of magnitude, η_{ix} is, by design, usually slightly less than 1 and γ_x is typically much greater than 1 because the outer springs are designed to be much stiffer relative to the inner springs in the x direction. Therefore, for simplicity, the total Coriolis force induced vibrations x_C can be assumed to consist only of x_{ci} . Simplifying (7.12) by using $\eta_{ix} \ll \gamma_x$:

$$x_C = x_{ci} \approx \frac{F_{ci}}{K_{xxi}(1 - \eta_{ix}^2 + j\eta_{ix}/Q_{ix})} \quad (7.14)$$

In terms of the drive displacement y_D , x_C can be written as:

$$x_C = \frac{2M_i\Omega_z\omega_{oy}y_D}{K_{xxi}(1 - \eta_{ix}^2 + j\eta_{ix}/Q_{ix})} = \frac{2\Omega_z\omega_{oy}y_D}{\omega_{ix}^2(1 - \eta_{ix}^2 + j\eta_{ix}/Q_{ix})} \quad (7.15)$$

Note that (7.14) is simply the response of the inner resonator as if the outer resonator was anchored and could have been obtained directly. The numerical value of x_C for unit input rotation rate ($1^\circ/s$) is $x_C = x_{ci} = 1.18 - 0.179j$ pm. Note that from (7.15), it is seen that ω_{oy} should be maximized to increase the displacement produced by Coriolis force. However, at the same time, η_{ix} should be less than one i.e., $\omega_{oy} \ll \omega_{ix}$ so that there is no attenuation due to the inner accelerometer being unable to respond to the Coriolis force.

Next, the conversion of the Coriolis force induced displacement to a differential voltage through the sense combs is analyzed. Referring to Figure 7.3 (b), the differential voltage output of the sense combs can be written as:

$$\frac{V_C}{V_m} = \left(\frac{C_1 - C_2}{C_1 + C_2 + C_{p1}} - \frac{C_3 - C_4}{C_3 + C_4 + C_{p2}} \right) \quad (7.16)$$

where, C_{p1} and C_{p2} are the parasitic capacitances as shown in Figure 7.3(b).

$$\frac{V_C}{V_m} = \frac{N_s \epsilon_0 t o_{lps}}{C_T} \left(\frac{4x_C}{g_s^2 - x_C^2} \right) \quad (7.17)$$

where it has been assumed that the total capacitance on the denominator is approximately the same for the top and the bottom and is equal to C_T . In order to get an idea of the magnitude of the signal obtained, the parasitic capacitance at each sense node is assumed to be about 500 fF. The sense capacitances themselves are computed as:

$$C_1 = C_2 = C_3 = C_4 = \frac{N_s \epsilon_0 o_{lps} t}{g_s} = 30.05 fF \quad (7.18)$$

The above value neglects the contribution of fringing fields, which can be expected to add about 20% to the capacitance values. Simplifying (7.17), using $x_C \ll g_s$, the output voltage normalized to the modulation voltage V_m corresponding to the above Coriolis force induced displacement is given as:

$$\frac{V_C}{V_m} = \frac{N_s \epsilon_0 t o_{lps}}{C_T} \left(\frac{4x_C}{g_s^2} \right) = \frac{C_1}{C_T} \left(\frac{4x_C}{g_s} \right) \quad (7.19)$$

Using previously computed values for x_C and $C_T = 560 fF$ the relative sensitivity is

obtained as $\frac{V_C}{V_m} = (0.142 \mu) / (^\circ/s)$. Most MEMS gyroscopes operate in the circuit-noise

limited regime. Therefore, the relative sensitivity number computed above, combined with the input-referred noise of the sense electronics, determines the resolution of the gyroscope. The above derivation of the output voltage of a gyroscope has also been done previously, by others [31][34].

Starting from symmetric drive voltages applied to the two actuation combs, the output voltage for a given input angular rate was derived above. Before proceeding to study non-idealities in the gyroscope caused by manufacturing variations it is instructive to look at the effect of asymmetric drive on the non-idealities in the gyroscope. This issue is addressed in the following section.

7.4 Effect of Asymmetrical Drive

The nested gyroscope design reported earlier [10][34] differs from the gyroscope described in this chapter. The difference being that a sinusoidal voltage is applied only to one of the two linear actuation combs. Sinusoidal voltage is applied to the top linear actuation comb. The bottom comb is used to capacitively sense the driven resonant oscillations. This sensed signal is amplified and fed back to the top linear comb thus completing an oscillator loop to sustain the oscillations. The main drawback of this approach is that

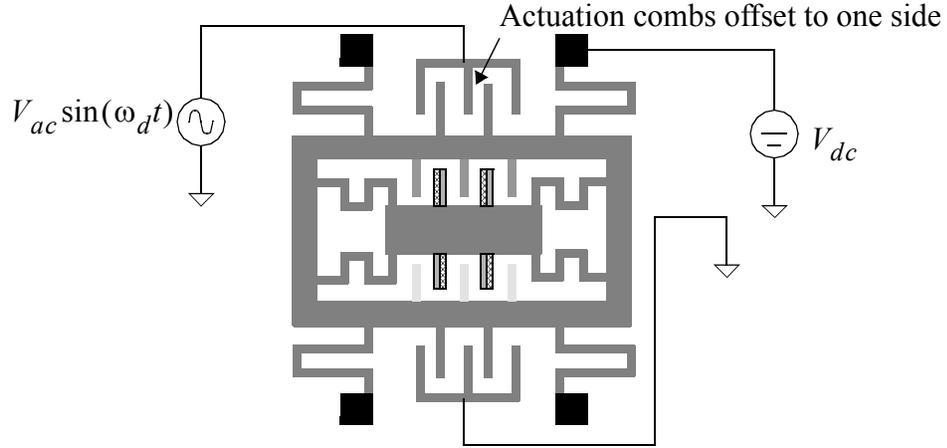


FIGURE 7.7. Asymmetrical drive: Actuation voltage applied to only the top linear comb

the gyroscope oscillations, being driven asymmetrically, will contain other modes (for example x modes) which can contribute significantly to the Zero Rate Output and other non-idealities of the gyroscope. The effect of such an asymmetric drive is analyzed in this section.

In the previous section the drive force was computed ((7.1) and (7.2)) assuming anti-phase voltages applied to the top and bottom drive combs and no displacement of the combs in the x direction. Equations (7.1) and (7.2) can be rewritten for a general case, where the rigid frame has been displaced in both the x and y direction. Let us assume that the frame along with the drive combs is offset, as shown in Figure 7.7, in the x direction by a distance x_{od} . To the first order, the drive force in the y direction can be assumed to be independent of the displacement in both the x and y directions. Therefore, the force in the y direction produced by each drive comb remains unchanged. Before proceeding to derive the total force for the asymmetrical drive case, the forces for each actuation comb are derived for the symmetrical anti-phase drive case. Following this, the force produced by the bottom comb is set to zero for the asymmetrical drive case. Assuming that the force due to parallel-plate capacitance dominates, the forces produced in the x direction can be written as:

$$F_{xt} = \frac{N_d \epsilon_0 t (o_{lpd} + y_D)}{2} \frac{4g_d x_{od}}{(g_d^2 - x_{od}^2)^2} (V_{dc}^2 + 2V_{dc}V_{ac} \sin(\omega_d t) + V_{ac}^2 \sin^2(\omega_d t)) \quad (7.20)$$

If the bottom actuation comb is driven by out of phase voltage, then the force due to the bottom comb can be written as:

$$F_{xb} = \frac{N_d \epsilon_0 t (o_{lpd} - y_D)}{2} \frac{4g_d x_{od}}{(g_d - x_{od})^2} (V_{dc}^2 - 2V_{dc}V_{ac} \sin(\omega_d t) + V_{ac}^2 \sin(\omega_d t)^2) \quad (7.21)$$

Assuming $x_{od} \ll g_d$ and summing (7.20) and (7.21) to get the total force in x ,

$$F_x = \frac{N_d \epsilon_0 t 4x_{od}}{2} \frac{1}{g_d^3} (2o_{lpd}(V_{dc}^2 + V_{ac}^2 \sin(\omega_d t)^2) + 2y_D(2V_{dc}V_{ac} \sin(\omega_d t))) \quad (7.22)$$

If the y displacement of the frame, y_D , is primarily composed of the fundamental drive frequency, then the above equation is significant because, it can be inferred that there is no component of F_x at the drive frequency. Components of F_x exist only at DC and the second harmonic frequency. Thus, it is seen that anti-phase voltages leading to symmetric drive conditions result in first order rejection of cross-axis forces at the drive frequency, which may arise due to a position offset in the drive combs. The cancellation of cross-axis forces due to the symmetrical drive is apparent in the above derivation.

On the other hand, if only the top comb was driven by a sinusoidal actuation voltage, as shown in Figure 7.7, the total force on the comb in the x direction is only F_{xt} , which has components at DC, the drive frequency and higher order harmonics too. Once again, assuming that the y motion is dominated by the fundamental frequency, the component at the fundamental frequency is:

$$F_{xt, \omega_d} = \frac{N_d \epsilon_0 t 4x_{od}}{2} \frac{1}{g_d^3} \left(o_{lpd}(2V_{dc}V_{ac} \sin(\omega_d t)) + y_D \left(V_{dc}^2 + \frac{V_{ac}^2}{2} \right) \right) \quad (7.23)$$

The above force will produce oscillations in the sense direction at the drive frequency and therefore, lead to a Zero Rate Output (*ZRO*). The *ZRO* referred to the input of the gyroscope is denoted as Ω_0 , i.e., the equivalent input rate required in an ideal gyroscope to produce the output observed in a non-ideal gyroscope when the external angular rate is zero. Recall from Section 7.3 that the drive motion lags behind the applied force (also the

applied voltage) by 90° . The resultant x force has components which are both in phase (due to the second term in the summation in (7.23)) and in quadrature to the drive motion (due to the first term in the summation in (7.23)). Furthermore, since the Coriolis force is in phase with the applied voltage, it is seen that there is a component of F_{xt, ω_d} which is exactly in phase with the Coriolis force. The magnitudes of the in-phase and quadrature components of F_{xt, ω_d} are given as:

$$F_{xt, \omega_d, I} = \frac{N_d \varepsilon_0 t^4 x_{od}}{2} \frac{1}{g_d^3} (o_{lpd} (2V_{dc} V_{ac})) \text{ and}$$

$$F_{xt, \omega_d, Q} = \frac{N_d \varepsilon_0 t^4 x_{od}}{2} \frac{1}{g_d^3} \left(y_D \left(V_{dc}^2 + \frac{V_{ac}^2}{2} \right) \right) \quad (7.24)$$

For an offset, $x_{od} = 0.1 \mu m$ and same voltages assumed as before i.e., $V_{dc} = 18V$ and $V_{ac} = 1V$, $F_{xt, \omega_d, I} = 4.02 nN$ and $F_{xt, \omega_d, Q} = -j11.8 pN$. Using the response obtained for the Coriolis force acting on the outer frame in (7.10), the relative displacement between the inner accelerometer central plate and the outer-resonator rigid frame due to F_{xt, ω_d} can be written as:

$$x_{va} = -\frac{F_{xt, \omega_d} \Gamma(\eta_{ix})}{K_{xxi} \gamma_x} \quad (7.25)$$

Comparing x_{va} in (7.25) with the dominant Coriolis force induced displacement in (7.12):

$$\frac{x_{va}}{x_C} = \frac{F_{xt, \omega_d} \eta_{ix}^2}{F_{ci} \gamma_x} \quad (7.26)$$

Using (7.9), (7.23) and (7.26) the in-phase and quadrature components of the motion coupling due to asymmetric drive can be obtained. The in-phase component is given as:

$$\frac{x_{va,I}}{x_C} = \frac{\frac{N_d \varepsilon_0 t 4 x_{od}}{2 g_d^3} (o_{lpd} (2 V_{dc} V_{ac})) \eta_{ix}^2}{\frac{2 M_i \Omega_z \omega_{oy} Q_d}{K_{yyo}} \left(N_d c_0 \varepsilon_0 \frac{t}{g_d} (4 V_{dc} V_{ac}) \right) \gamma_x} = \frac{K_{yyo} x_{od} o_{lpd} \eta_{ix}^2}{2 M_i \Omega_z \omega_{oy} Q_d c_0 \gamma g_d^2} \quad (7.27)$$

The quadrature component is given as:

$$\frac{x_{va,Q}}{x_C} = \frac{\frac{N_d \varepsilon_0 t 4 x_{od}}{2 g_d^3} \left(y_D \left(V_{dc}^2 + \frac{V_{ac}^2}{2} \right) \right) \eta_{ix}^2}{\frac{2 M_i \Omega_z \omega_{oy} Q_d}{K_{yyo}} \left(N_d c_0 \varepsilon_0 \frac{t}{g_d} (4 V_{dc} V_{ac}) \right) \gamma_x} = \frac{K_{yyo} x_{od} \left(y_D \left(V_{dc}^2 + \frac{V_{ac}^2}{2} \right) \right) \eta_{ix}^2}{(4 M_i \Omega_z \omega_{oy} Q_d) (c_0 (V_{dc} V_{ac})) \gamma_x g_d^2} \quad (7.28)$$

Using values from Table 7.2 and Table 7.3 the in-phase and quadrature ZRO due to asymmetric drive are, respectively, computed to be $\Omega_{0,va,I} = x_{va,I}/x_C = 37.2(^{\circ}/s)$ and $\Omega_{0,va,Q} = x_{va,Q}/x_C = -0.109(^{\circ}/s)$.

Thus, it is seen that asymmetric drive results in a ZRO even when there are no manufacturing induced variations, i.e., even in an otherwise ideal gyroscope. Furthermore, it can be shown similarly that, if each individual drive comb produces a vertical force, then symmetric drive will cause the two vertical forces to cancel out whereas asymmetric drive will lead to a vertical oscillatory motion at the drive frequency. Vertical oscillatory motion at the drive frequency introduces a small cross-axis sensitivity to the vertical-axis gyroscope, and is, therefore, not desirable. Having shown that asymmetric drive results in non-idealities even in a geometrically perfect gyroscope, manufacturing induced imperfections in gyroscope geometry and resultant non-idealities are analyzed in the following sections. In order to isolate the effects of manufacturing variations, symmetric drive is assumed in all the analyses and simulations in the remainder of this chapter.

7.5 Zero Rate Output (ZRO)

The causes of ZRO include beam width variations leading to elastic cross-coupling, comb gap variations, lateral offsets of the comb position, mask misalignment leading to in-plane to out-of-plane mode coupling. Each of these causes is analyzed in detail in this

section. Lateral offset in comb position is not considered separately but in combination with the other causes. One sub-section is devoted to each of the three causes: beam width variation, comb gap variation and mask misalignment. Equations relating *ZRO* to these manufacturing-induced variations are derived. As before, numerical evaluations are also carried out at each step so that the relative significances of different causes can be clearly identified. Complex number notation is used in order to maintain information about the magnitude as well as the phase of the signals.

7.5.1 Beam Width Variation

Beam width variation leads to in-plane elastic cross-axis coupling as explained in Chapter 3. In general, beam width variations across a wafer are modeled as a linear gradient. As discussed in Section 3.9, gradients along three directions are first considered in this section. Following this, the effect of mismatch in a single spring, with respect to the other three springs is analyzed. It should be noted that the mismatch of a single spring with respect to the other three springs is a canonical problem and can be used to solve instances with generalized mismatch between beam widths in different springs.

The results of *ZRO* resulting from linear gradients along x , y and along both x and y are shown in Table 7.4. The gradients are assumed to be such that when there is a gradient

Table 7.4 ZRO resulting from linear gradients in beam width

Gradient direction	ZRO ($^{\circ}/s$)
x	below numerical precision
y	below numerical precision
x, y	16

along the x direction, then the two springs on the left have their width diminished by 1% and the two springs on the right have their widths enhanced by 1%. Similarly for gradients along the y direction, beam widths in the two springs at the bottom are diminished by 1% and those of the two springs at the top are enhanced by 1%. When there is a simultaneous gradient along both x and y directions, the spring at the bottom left is diminished by 1.4% and that at the top right is enhanced by 1.4%. The other two springs are left unchanged in this case. The *ZRO* for gradients along the x and y directions was below the numerical pre-

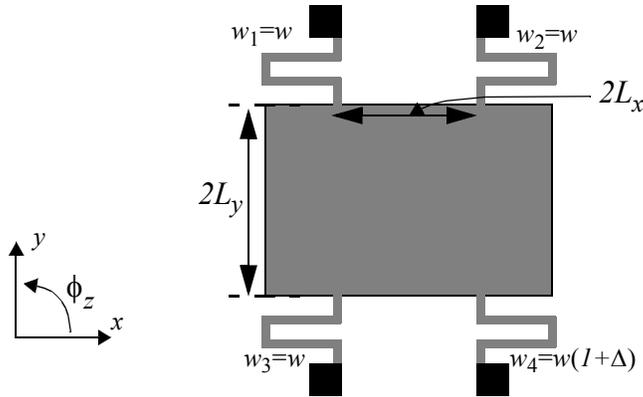


FIGURE 7.8. Example case for beam width variation and equation for coupling, $w_1 = w_2 = w_3 \neq w_4$.

cision of the simulation and is presumed to be zero. However, when there is a simultaneous gradient along both the x and y directions, there is a significant ZRO of about 16 ($^\circ/s$). This is exactly as predicted by the analysis presented in Section 3.9 which is further expanded below. The above result is in slight contradiction to analysis presented in [31], wherein it is reported that variations in the thickness of springs along the direction perpendicular to the drive axis lead to more cross-axis coupling. However, the simulations and analysis in this thesis show that cross-axis coupling is indeed maximum when the gradient in widths has both an x and y component. The reason for this apparent contradiction is explained below. The point at which cross-axis coupling is observed is important (i.e., the point at which the displacement is measured). The geometrical center of the layout is the ideal point to make this observation. If a point away from the center is chosen, then it will be difficult to distinguish between cross-axis motions produced due to rotation about the geometrical layout center and true translational motion itself. It should be noted that cross-axis motions produced due to rotations about the geometrical center will be nominally cancelled out by the sense combs. Therefore, it is possible that an off-center point was chosen to measure the cross-axis coupling in [31]. In the remaining simulations and analyses in this chapter it is assumed that the beam widths of only one spring is mismatched with respect to the other three springs. This is a canonical problem whose solution can be used to compute the solution for any combination of beam width variations.

In any system consisting of a proof-mass suspended by mismatched springs (see Figure 7.8), there are essentially two mechanisms by which in-plane elastic cross-coupling

can occur as discussed in Section 3.8. In both the following cases it is assumed that the width of one out of the four springs is varied by a fraction Δ as shown in Figure 7.8.

1. Individual springs suspending the proof-mass have a non-zero intrinsic k_{xy} . The ratio of x displacement to y displacement (referring to Figure 7.8) when there is no force in the x direction is given as:

$$\frac{x}{y} = \frac{K_{xy}}{K_{xx}} = \frac{\Sigma k_{xyi}}{\Sigma k_{xxi}} \quad (7.29)$$

where, k_{xyi} is the cross-axis stiffness of the i^{th} spring. The overall cross-axis term Σk_{xyi} , which is a summation of the individual k_{xyi} varies linearly with Δ . Recalling from Section 3.8 that a fractional change of Δ in the beam width produces a fractional change of 3Δ in the stiffness constants, the above equation can be rewritten as:

$$\frac{x}{y} = \frac{3k_{xy}\Delta}{4k_{xx}} \quad (7.30)$$

where, k_{xy} and k_{xx} are the stiffness constants of the nominal (ideal) spring.

2. The second case is when each individual spring has zero intrinsic k_{xy} as proved in Section 3.8 and [53]. Spring designs that are completely symmetrical possess this property and are, therefore, highly recommended for reducing elastic cross-coupling. In this case there is no direct coupling between the two in-plane translational modes x and y , but, there is second-order coupling through the rotational mode. In other words the y mode first couples to the rotational mode ϕ_z , which in turn couples to the x mode.

$$\frac{x}{y} = \frac{(\Sigma k_{x\phi_zi})(\Sigma k_{y\phi_zi})}{(\Sigma k_{xxi})(\Sigma k_{\phi_z\phi_zi})} \quad (7.31)$$

Each of the cross-axis terms Σk_{xyi} , $\Sigma k_{x\phi_zi}$ and $\Sigma k_{y\phi_zi}$ varies linearly with the width mismatch Δ . Therefore, in the first case the elastic cross-coupling is linear, whereas in the

second case, the linear portion has been eliminated by better spring design and the remaining coupling is a quadratic function of Δ . If the intrinsic rotational stiffness of the springs is neglected, the stiffnesses of a single nominal spring in (7.31) can be written as (from (3.16) and (3.17))

$$k_{x\phi_z i} = k_{xxi}L_y \quad (7.32)$$

$$k_{y\phi_z i} = k_{yyi}L_x \quad (7.33)$$

$$k_{\phi_z\phi_z i} = k_{xxi}L_y^2 + k_{yyi}L_x^2 \quad (7.34)$$

Note that the sign of the cross-axis terms $k_{x\phi_z i}$ and $k_{y\phi_z i}$ changes with the quadrant in which the spring is located, while the diagonal term $k_{\phi_z\phi_z i}$ is always positive.

If the springs are designed to be highly stiff in one direction (say x) and highly compliant in the orthogonal direction (y), then, noting that spring stiffnesses vary as the cube of the width, (7.31) can be simplified to:

$$\frac{x}{y} = \frac{(3k_{xx}L_y\Delta)(3k_{yy}L_x\Delta)}{(4k_{xx})(4k_{xx}L_y^2)} = \frac{9k_{yy}L_x\Delta^2}{16k_{xx}L_y} \quad (7.35)$$

Similarly,

$$\frac{y}{x} = \frac{(3k_{xx}L_y\Delta)(3k_{yy}L_x\Delta)}{(4k_{yy})(4k_{xx}L_y^2)} = \frac{9L_x\Delta^2}{16L_y} \quad (7.36)$$

where, k_{xx} and k_{yy} are the nominal values of the spring stiffnesses.

For the nested gyroscope under consideration, there are 2 sets of springs and the mismatch in the 2 sets are considered separately. Each outer spring is symmetric in itself, therefore, mismatch in the outer spring corresponds to the second case above. Each inner spring is not symmetric in itself, and therefore, mismatch in inner springs is analyzed using case one above.

Derivation similar to that used to derive (7.10) yields the following equations using (7.30). Coupling of the drive into the sense mode due to relative beam width mismatch (Δ) in outer (Figure 7.9) and inner springs are respectively given as:

$$\frac{x_{so}}{y_D} = \frac{(\Sigma k_{x\phi_z o})(\Sigma k_{y\phi_z o})}{(\Sigma k_{xxo})(\Sigma k_{\phi_z\phi_z o})} \Gamma(\eta_{ix}) \approx \frac{9L_x k_{yyo} \Delta^2}{16L_y k_{xxo}} \frac{\eta_{ix}^2}{\left(1 - \eta_{ix}^2 + \frac{j\eta_{ix}}{Q_{ix}}\right)} \quad (7.37)$$

$$\frac{x_{si}}{y_D} = \frac{\Sigma k_{xyi} \left(\frac{\omega_{oy}}{\omega_{iy}}\right)^2 \left(\frac{1}{1 - \eta_{ix}^2 + j\eta_{ix}/Q_{ix}}\right)}{\Sigma k_{xxi}} = \frac{3k_{xyi}}{4k_{xxi}} \frac{\eta_{iy}^2}{1 - \eta_{ix}^2 + j\eta_{ix}/Q_{ix}} \Delta \quad (7.38)$$

where, $\eta_{ix} = \frac{\omega_{oy}}{\omega_{ix}}$, $\eta_{iy} = \frac{\omega_{oy}}{\omega_{iy}} = \frac{\omega_d}{\omega_{iy}}$, as before, ω_{ix} is the resonant frequency of the inner resonator in the sense (x) direction, ω_{oy} and ω_{iy} are the resonant frequencies of the outer and inner resonators in the drive (y) direction and Γ is as defined in (7.11). x_{so} and x_{si} respectively denote the displacement produced in the sense direction due to spring mismatch in the outer and inner springs respectively. Using the spring stiffness computation procedure described in Chapter 4 $k_{xyi} = 0.010(N/m)$. The approximate numerical values are $x_{so} = -0.138 - 0.914j$ pm and $x_{si} = -115.3 - 761.3j$ pm. The outer springs being inherently symmetric, the coupling from drive to sense mode is proportional to Δ^2 ,

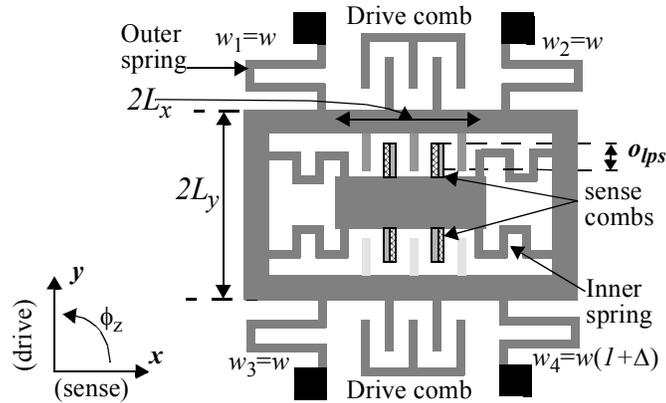


FIGURE 7.9. Example case for beam width variation, $w_1 = w_2 = w_3 \neq w_4$.

whereas, in the asymmetric inner springs the coupling is not only proportional to Δ but is also more than 10 times higher. Also, in case of the nested gyroscope, the topology lends itself to isolating the drive mode to the outer springs and the sense modes to the inner springs i.e., $k_{yyo}/k_{xxo} \ll 1$ and $k_{yyi}/k_{xxi} \gg 1$. The corresponding input referred ZROs can be computed by taking the ratios of x_{so} and x_{si} to x_C the Coriolis force induced displacement for unit rotation rate.

$$\Omega_{0, wo} = \frac{x_{so}}{x_C} = 0.845j(^{\circ}/s) \text{ and } \Omega_{0, wi} = \frac{x_{si}}{x_C} = 644.9j(^{\circ}/s) \quad (7.39)$$

Note that $\Omega_{0, wi}$ can also be approximated as the ratio of the elastic cross-coupling force to the Coriolis force acting on the inner plate:

$$\Omega_{0, wi} = \frac{x_{si}}{x_C} = \frac{K_{xy}y_{Ds}}{F_{ci}} = \frac{K_{xy}\eta_{iy}^2}{2jM_i\omega_{oy}} \quad (7.40)$$

The *ZRO* due to beam width mismatch is in exact quadrature with the Coriolis force induced output as indicated by the j in the denominator of (7.40).

From the above analysis, it is seen that beam width variations lead to *ZRO* through two different mechanisms in the nested gyroscope. The outer springs are symmetric, and therefore show only second-order coupling between the drive and sense modes, while the inner springs being asymmetric show first order coupling. Additionally, the outer springs have been designed to attenuate sense mode oscillations resulting in much smaller *ZRO* due to width mismatch in the outer springs than due to width mismatch in the inner springs. It is also interesting to note that the *ZRO* caused by beam width mismatch is in exact quadrature with the normal gyroscope output and therefore, can be eliminated by demodulation with appropriate phase. However, phase errors in the local oscillator of the demodulator will lead conversion of the quadrature *ZRO* to final gyroscope output.

Design options to reduce *ZRO* include use of wider beam widths to average out lithographic variations, use of symmetric springs to eliminate first order coupling and springs

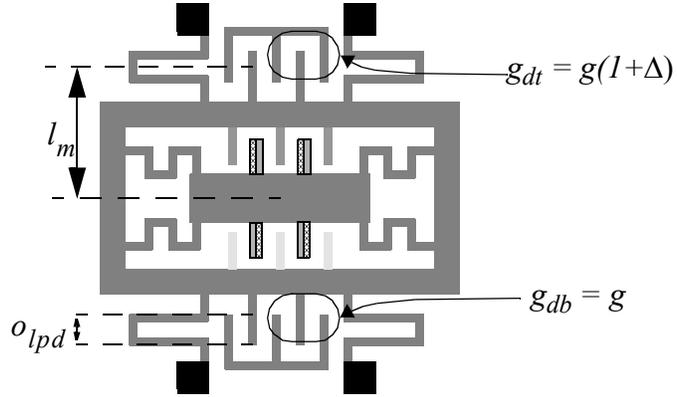


FIGURE 7.10. Mismatch in the gaps in the drive combs on the top and bottom

which have lower k_{yy}/k_{xx} . Increasing the separation between the y modes of the inner resonator and the outer resonator (i.e., decreasing η_{iy}) will also lead to decreased ZRO .

7.5.2 Comb gap variation

In this sub-section, the impact of mismatch in the gaps between the comb fingers on the ZRO is analyzed. All the gaps in the same comb, are assumed to be identical and variations are assumed to occur only between different combs. There are two cases:

1. mismatch between the gaps in the top and bottom linear actuation combs and
2. mismatch between the gaps in the top and bottom differential sense combs

Before analyzing the multidirectional forces produced due to mismatch in the actuation combs, it is asserted here that if there is no mismatch between the two actuation combs and if symmetrical anti-phase voltages are applied, then there is no net force produced by the actuation combs at the drive frequency other than in the drive direction. This statement is contradictory to prior analysis [31]. This is because in the analysis presented in [31], only one drive comb is considered. Therefore, the cancellation effect of the other drive comb is not observed. Now mismatch in the gaps in the two drive combs as shown in Figure 7.10 will be shown to produce multi-directional forces at the drive frequency in the presence of mismatch. It is assumed that the gaps in the top and bottom drive combs are different and are represented by g_{dt} and g_{db} respectively. For a displacement x_t in the x direction, the x force in the top comb is given as:

$$F_{xt} = \frac{\varepsilon_0 t (o_{lpd} + y_D)}{2} \frac{4g_{dt} x_t}{(g_{dt} - x_t)^2} (V_{dc}^2 + 2V_{dc} V_{ac} \sin(\omega t) + V_{ac}^2 \sin^2(\omega t)) \quad (7.41)$$

If the bottom comb is being driven out of phase, then, the x force produced by the bottom comb can be written as:

$$F_{xb} = \frac{\varepsilon t (o_{lpd} - y_D)}{2} \frac{4g_{db} x_b}{(g_{db} - x_b)^2} (V_{dc}^2 - 2V_{dc} V_{ac} \sin(\omega t) + V_{ac}^2 \sin^2(\omega t)) \quad (7.42)$$

Let us now consider two cases for computation of the total force in x and the total moment about z . In the following analyses a parallel plate approximation is used to compute the forces and the moments.

The first case is a common mode x displacement for the top and bottom combs: $x_t = x_b = x_{cm}$. Such a displacement will be produced by x translational motion. The total force and moment are obtained by summing (7.41) and (7.42). The nominal part and the mismatch dependent part of the force and moment are written separately as:

$$F_{x, nom} = \frac{N_d \varepsilon_0 t \left(\frac{8x_{cm}}{3} \right)}{2 g_d} \left((o_{lpd} (V_{dc}^2 + V_{ac}^2 \sin^2(\omega t)) + y_D (2V_{dc} V_{ac} \sin(\omega t))) \right) \quad (7.43)$$

$$F_{x, \Delta} = \frac{N_d \varepsilon_0 t 4x_{cm}}{2 g_d} (3\Delta) \left((o_{lpd} (2V_{dc} V_{ac} \sin(\omega t)) + y_D (V_{dc}^2 + V_{ac}^2 \sin^2(\omega t))) \right) \quad (7.44)$$

$$T_{\phi, nom} = \frac{N_d \varepsilon_0 t 4x_{cm}}{2 g_d} 2 \left(\left(o_{lpd} l_m + \frac{y_D^2}{2} \right) (2V_{dc} V_{ac} \sin(\omega t)) + \left(l_m + \frac{o_{lpd}}{2} \right) y_D (V_{dc}^2 + V_{ac}^2 \sin^2(\omega t)) \right) \quad (7.45)$$

$$T_{\phi, \Delta} = \frac{N_d \varepsilon_0 t 4x_{cm}}{2 g_d} 3\Delta \left(\left(o_{lpd} l_m + \frac{y_D^2}{2} \right) (V_{dc}^2 + V_{ac}^2 \sin^2(\omega t)) + \left(l_m + \frac{o_{lpd}}{2} \right) y_D (2V_{dc} V_{ac} \sin(\omega t)) \right) \quad (7.46)$$

l_m is the distance in the y direction from the center of the overlapped region of the actuation combs to the center of the gyroscope itself. If it is assumed that x_{cm} has only low frequency content and y_D has only drive frequency component, then the nominal force given by (7.43) has components only at DC and at even harmonics of the drive frequency. This is expected from the discussion in Section 7.4. However, (7.44) shows that in the presence of a lateral offset x_{cm} in the drive combs and a mismatch Δ , there is a force in the sensing direction at the drive frequency. Note that this force is present in spite of the symmetric nature of the drive voltage. Such a lateral offset can either arise due to manufacturing effects or be caused by a linear acceleration thus leading to a linear acceleration sensitivity. Displacements caused by linear accelerations are usually about 2 orders of magnitude smaller than manufacturing-induced offsets. Magnitude of the force is obtained as:

$$F_{xgd, I} = \frac{N_d \epsilon_0 t^4 x_{cm}}{2} \frac{(3\Delta)(o_{lpd} 2V_{dc} V_{ac} \sin(\omega t))}{g_d^3} = 0.121 nN \quad (7.47)$$

$$F_{xgd, Q} = \frac{N_d \epsilon_0 t^4 x_{cm}}{2} \frac{(3\Delta)(y_D (V_{dc}^2 + V_{ac}^2 \sin^2(\omega t)))}{g_d^3} = -0.459 j nN \quad (7.48)$$

where, the I and Q represent in-phase and quadrature with respect to the Coriolis force. Reusing the analysis used to derive (7.10), relative displacements between the inner plate and the outer frame due to $F_{xgd, Q}$ and $F_{xgd, I}$ are given as:

$$x_{gd, I} = -\frac{F_{xgd, I} \Gamma(\eta_{ix})}{K_{xxi} \gamma_x} = (-0.676 + 0.103j) pm \quad \text{and} \quad (7.49)$$

$$x_{gd, Q} = -\frac{F_{xgd, Q} \Gamma(\eta_{ix})}{K_{xxi} \gamma_x} = (0.393 + 2.57j) pm \quad (7.50)$$

The corresponding ZROs are given as:

$$\Omega_{0, gd, Q} = \frac{x_{gd, Q}}{x_C} = 2.18j (^\circ/s) \quad \text{and} \quad (7.51)$$

$$\Omega_{0, gd, I} = \frac{x_{gd, I}}{x_C} = 0.572(^{\circ}/s) \quad (7.52)$$

Since the rotational mode (ϕ_z) frequency is usually higher than the greater of the two translational modes (see Appendix A5) the rotation produced by $T_{\phi, \Delta}$ will not be large enough to compete with the expected range of external rotation rates and is, therefore, not analyzed further. It should be noted that the magnitudes of the forces in (7.47) and (7.48) are about two orders of magnitude larger than the Coriolis force. However, as will be seen below, they do not lead to significant *ZRO*. The reason for this is that the outer springs are much stiffer in x effectively attenuating the displacement produced by the above forces. This is in sharp contrast to single layer gyroscopes such as [19][28].

The second case is when $-x_b = x_t = x_d$. This case corresponds to rotational movement of the outer frame. Then the forces and moments are again separated into nominal and mismatch dependent parts and written as:

$$F_{x, nom} = \frac{\varepsilon t}{2} \frac{4x_d}{(g_d)^3} 2((o_{lpd})(2V_{dc}V_{ac}\sin(\omega t)) + y_D(V_{dc}^2 + V_{ac}^2\sin(\omega t)^2)) \quad (7.53)$$

$$F_{x, \Delta} = \frac{\varepsilon t}{2} \frac{4x_d}{(g_d)^3} \Delta((o_{lpd})(V_{dc}^2 + V_{ac}^2\sin(\omega t)^2) + y_D(2V_{dc}V_{ac}\sin(\omega t))) \quad (7.54)$$

$$T_{\phi, nom} = \frac{\varepsilon t}{2} \frac{4x_d}{(g_d)^3} 2\left(\left(o_{lpd}l_m + \frac{y_D^2}{2}\right)(V_{dc}^2 + V_{ac}^2\sin(\omega t)^2) + \left(l_m + \frac{o_{lpd}}{2}\right)y_D(2V_{dc}V_{ac}\sin(\omega t))\right) \quad (7.55)$$

$$T_{\phi, \Delta} = \frac{\varepsilon t}{2} \frac{4x_d}{(g_d)^3} \Delta\left(\left(o_{lpd}l_m + \frac{y_D^2}{2}\right)(2V_{dc}V_{ac}\sin(\omega t)) + \left(l_m + \frac{o_{lpd}}{2}\right)y_D(V_{dc}^2 + V_{ac}^2\sin(\omega t)^2)\right) \quad (7.56)$$

If it is assumed that x_{cm} has only low frequencies and x_d has only drive frequency present, then from (7.53) and (7.56) then it can be concluded that $F_{x,nom}$ and $T_{\phi,\Delta}$ will have only low frequency and bands centered around *even* harmonics of the drive frequency whereas $F_{x,\Delta}$ and $T_{\phi,nom}$ will have only bands centered around the drive frequency and *odd* harmonics of the drive frequency. In general, the nominal components $F_{x,nom}$ and $T_{\phi,nom}$ are expected to be much larger than the mismatch-dependent components $F_{x,\Delta}$ and $T_{\phi,\Delta}$. Therefore, $F_{x,nom}$ having drive frequency components and $T_{\phi,nom}$ having low frequency components is a less acceptable situation than the two having low frequency and drive frequency components respectively. The reason being that $F_{x,nom}$ having drive frequency components will lead to coupling of motion at drive frequency to the sense mode and $T_{\phi,nom}$ having low frequency components will produce rotations of the rigid frame and the inner plate. Such a rotation will itself lead to a Coriolis force as if an external rotation had been applied. From (7.53) and (7.56) it is seen that low frequency rotations of the rigid frame lead to both $F_{x,nom}$ and $T_{\phi,nom}$ having components which will interfere with normal gyro operation. Comparing (7.53) with (7.44) if $x_d = x_{cm}$, the force produced in (7.53) is significantly larger (being independent of Δ) than the force produced in (7.44). Thus it is seen that rotational offsets produce coupling of drive motion to the sense mode, even with perfectly matched comb gaps. This completes the discussion on *ZRO* produced by coupling in the drive combs.

Next mismatch in the gaps between the two differential sense combs, as shown in Figure 7.11, is considered. The movement in the sense combs in the y direction is given as:

$$y_{ds} = y_{do} - y_{di} = y_{do} \left(\frac{\omega_{oy}}{\omega_{iy}} \right)^2 \approx y_D \eta_{iy}^2 = 24.2nm \quad (7.57)$$

It is now assumed that there is a lateral offset in the x direction (x_{oS}) in the differential sense combs. Rewriting (7.17) with different symbols for the top and bottom sense gaps:

$$V = \frac{N_s \varepsilon_0 t}{C_T} \left((o_{lps} + y_{ds}) \left(\frac{2x_{os}}{g_{ts} - x_{os}^2} \right) + (o_{lps} - y_{ds}) \left(\frac{2x_{os}}{g_{bs} - x_{os}^2} \right) \right) \quad (7.58)$$

Assuming $x_{os} \ll g_{ts}, g_{bs}$

$$V = \frac{V_m N_s \varepsilon_0 t}{C_T} \left((o_{lps}) \left(\frac{2x_{os}}{g_{ts}} + \frac{2x_{os}}{g_{bs}} \right) + (2x_{os} y_{ds}) \left(\frac{1}{g_{ts}} - \frac{1}{g_{bs}} \right) \right) \quad (7.59)$$

$$V = \frac{V_m N_s \varepsilon_0 t}{C_T} \left((o_{lps}) \left(\frac{4x}{g_s} \right) + (2xy_{ds}) \left(\frac{2\Delta}{g_s} \right) \right) \quad (7.60)$$

The above equation suggests that if there is a lateral offset in the sense direction, then there will be a voltage output proportional to the relative displacement of the sense combs in the drive direction y . The first term in the summation produces output proportional to the displacement in the sense mode i.e., the Coriolis force induced displacement as well as direct response of the inner accelerometer to external accelerations. Assuming the lateral offset $x_{os} = 0.1 \mu m$ and the relative mismatch $\Delta = 0.01$, the input referred ZRO is obtained as:

$$\Omega_{0,gs} = \frac{x_{os} y_{Ds} \Delta}{x_C o_{lps}} = (0.0507 - 0.335j) (^{\circ}/s) \quad (7.61)$$

Note that if the relative displacement in the sense combs in the drive direction was equal to the drive displacement (i.e., there is no decoupling, therefore, $y_{ds} = y_D$) then the

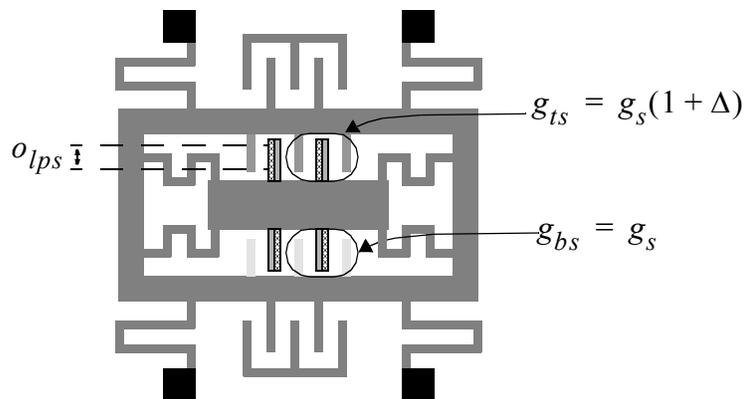


FIGURE 7.11. Mismatch in the gaps in the sense combs on the two sides

a constant. Using a parallel plate approximation for the sense comb capacitance, the normalized output voltage in such a case, can be written as:

$$\frac{V_{o, mm}}{V_m} = \left(\frac{2N_s \epsilon_0 o_{lp_s} z_s}{g_s^2} (2x_{os}) \right) / (C_T) \quad (7.62)$$

where, z_s is the relative vertical motion in the sense combs and $V_{o, mm}$ is the output voltage due to mask misalignment, even when there is no rotational input. The input referred ZRO is obtained by dividing by the output voltage for the Coriolis force induced displacement in (7.19):

$$\Omega_{0, mm} = \frac{z_s x_{os}}{t x_C} \quad (7.63)$$

Using the theory described in Section 3.7, z_s , the relative vertical motion in the sense combs caused due to mask misalignment is now estimated. Mask misalignment and elastic coupling in the outer springs are considered first. It is assumed that the angle by which the principal axes of a spring rotates is the same as the angle by which the principal axes of each of the individual compliant beams rotates (θ_{oyz}). This assumption is valid when all the compliant beams are parallel to each other and the trusses connecting the compliant beams are short and stiff compared to the compliant beams. With an additional assumption that the z resonant mode (ω_{oz}) is well separated from the drive frequency (ω_{oy}), viscous and inertial effects in the vertical direction can be neglected. Motion coupled to the z axis in the outer springs can then be written as:

$$z_o = \left(\frac{K_{yzo}}{K_{zzo}} \right) y_D \quad (7.64)$$

If the y stiffness is assumed to be small compared to the z stiffness then, using (3.70) and (3.72):

$$z_o = \left(\frac{\sin(2\theta_{oyz})}{2} \right) y_D \approx \theta_{oyz} y_D \quad (7.65)$$

for small values of θ_{oyz} . This is a crude approximation, since typically, $K_{zz}/K_{yy} \approx 4$ for CMOS-MEMS springs of width $1.8 \mu m$, but allows us to quickly simplify K_{yz} in (7.64). With inclusion of inertial effects a slightly more complex expression for relative vertical motion between the central plate and the frames is obtained:

$$z_s = \frac{F_z}{K_{zzo}} \frac{\eta_{iz}^2}{1 - \eta_{iz}^2 - \gamma_z \eta_{iz}^2 - \eta_{oz}^2 + \eta_{iz}^2 \eta_{oz}^2} \approx (\theta_{oyz} y_D) \frac{\eta_{iz}^2}{1 - \eta_{iz}^2 - \gamma_z \eta_{iz}^2 - \eta_{oz}^2 + \eta_{iz}^2 \eta_{oz}^2} \quad (7.66)$$

where, $\eta_{iz} = \frac{\omega_{oy}}{\omega_{iz}}$, $\omega_{iz}^2 = \frac{K_{zzi}}{M_i}$, $\eta_{oz} = \frac{\omega_{oy}}{\omega_{oz}}$, $\omega_{oz}^2 = \frac{K_{zzo}}{M_o}$ and $\gamma_z = \frac{K_{zzi}}{K_{zzo}}$. The above

equation is derived using a similar system of equations as is described in Figure 7.6, with the y subscripts replaced by z . Assuming sufficient mode separation and quality factors much greater than 1, the damping terms in those equations can be neglected. For a $0.1 \mu m$ misalignment of the METAL2 and METAL1 layers with respect to the METAL3 layer, $\theta_{oyz} = 0.02$ radians. Also note that the ratio ω_{oz}/ω_{oy} is approximately proportional to the aspect ratio of the CMOS-MEMS beams, typically between 2 to 3 for nested gyroscopes in a standard CMOS process. (for example width of $1.8 \mu m$ and thickness of $4.5 \mu m$). Using values in Table 7.3, $z_s = -0.332j$ nm. The j term indicates that the vertical motion is in-phase with the drive displacement, as expected from (7.66). Thus, it is seen that for a 0.02 radian rotation of the principal axes due to misalignment, about 0.02% of the drive motion couples (in phase) to the relative vertical motion in the sense combs. Since the Coriolis force induced displacement is not in exact quadrature with respect to the drive displacement, the vertical motion coupling leads to in-phase and quadrature ZRO . Back-substituting values for z_s in (7.63), $\Omega_{0,mm} = 0.840 - 4.62(^\circ/s)$. It is seen that $\Omega_{0,mm}$ is more significant than $\Omega_{0,gd}$ and $\Omega_{0,gs}$, for the same value of lateral offset in the sense comb. So far only mask misalignment in the outer spring beams has been considered. Mask misalignment in the inner springs is discussed next.

The long beams in the inner springs are parallel to the y (drive) direction and therefore, compliant in the x (sense) direction. Mask misalignment in the long beams of the

inner springs can only lead to coupling between the x and z modes. Mask misalignment in the shorter truss beams, which are parallel to the x direction will lead to negligible elastic cross-axis coupling between the y and z modes, because the truss beams are short and stiff. Furthermore, the relative y motion in the inner springs ($y_{ds} = y_{di} - y_{do} = y_D \Gamma(\eta_{iy}^2)$) is only a small fraction (about 1.5%) of the total drive motion y_D . Therefore, the elastic coupling between y and z modes in the inner springs is not analyzed further.

7.5.4 ZRO Summary

ZRO due to beam width mismatch in the inner (asymmetric) springs is found to be significant, but in exact quadrature with the Coriolis force induced displacement. *ZRO* due to mismatched gaps is seen to be directly related to the offset in the sense combs as well as the decoupling of sense mode from the drive comb and vice-versa. It is seen that mask misalignment in the outer springs can cause greater *ZRO* than gap mismatch, with reasonable assumptions for misalignment and mismatch values. It is also seen that the quadrature component of *ZRO* is more significant than the in-phase component for both gap mismatch and mask misalignment cases. This suggests that the in-phase component can be decreased even more by pushing the sense resonant mode further away from the drive resonant mode (i.e., decrease η_{ix}). There will, of course, be an accompanying sensitivity reduction.

The analyses for deriving *ZRO* equations establishes the basic effects of beam width mismatch, comb gap mismatch and mask misalignment. An external acceleration is added in the next section. The resulting analyses for acceleration sensitivity are closely related to analyses presented in this section.

7.6 Acceleration Sensitivity

In this section the impacts of beam width mismatch, comb gap mismatch and mask misalignment on the acceleration and acceleration-squared sensitivity of the gyroscope are considered. The spectrum of a non-ideal gyroscope when subjected to an external acceleration is shown in Figure 7.13. At ω_a the acceleration response of the inner accelerometer is seen. The gyroscope *ZRO* is seen at ω_d , the drive frequency. Sidebands at $\omega_d - \omega_a$ and

$\omega_d + \omega_a$ arise due to the acceleration sensitivity of the non-ideal gyro and sidebands at $\omega_d - 2\omega_a$ and $\omega_d + 2\omega_a$ due to the acceleration-squared sensitivity. In Section 7.5 ZRO was seen to arise due to completely linear phenomena. However, acceleration and acceleration-squared sensitivity necessarily imply the presence of a non-linear element which “mixes” the low-frequency (ω_a) acceleration signal and the drive frequency (ω_d) to produce the side-bands at $\omega_d \pm \omega_a$ and $\omega_d \pm 2\omega_a$ as seen in Figure 7.13. While both the drive and sense combs are non-linear elements, the drive comb is linear during actuation even for relatively large displacements (i.e., few μm), whereas, the sense comb can be considered linear only for displacements which are a very small fraction of the gap (i.e., tens of nm). Therefore, in the following analyses, the major source of non-linearity is the sense comb. Non-linearity in the beams in the outer springs, which experience significant displacement, is a complex area of research in itself and is not considered in this thesis [40][93][94][95].

Before the effects of variations on acceleration sensitivity are considered, a brief examination of the sense comb non-linearity is in order. From (7.17) the voltage output of the differential sense combs can be written as a Taylor’s series expansion in terms of the displacement x of the comb fingers in the lateral (gap) direction:

$$\frac{V_o}{V_m} = \frac{4N_s \epsilon_0 t o l p s}{g_s C_T} \left(\frac{x}{g_s} + \left(\frac{x}{g_s} \right)^3 + \left(\frac{x}{g_s} \right)^5 + \dots \right) \quad (7.67)$$

$x = x_{os} + x_{As} + x_{ds}$, is the total displacement of the comb in the sense direction, comprised of a DC offset term, x_{os} , a sinusoidal acceleration term $x_{As} = \frac{G}{\omega_{ix}^2}$, and a drive

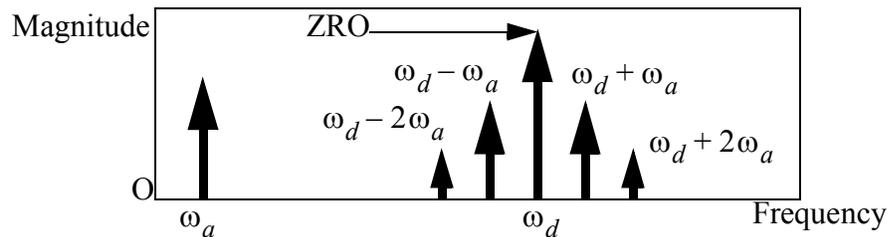


FIGURE 7.13. Spectrum of output voltage of a non-ideal gyroscope when subjected to an external acceleration.

frequency term x_{ds} due to coupling. The cubic term in (7.67) leads to mixing of the three components in the x direction. Furthermore, it is seen from (7.67) that periodic variation in the overlap length o_{lps} along y and the overlap length along the z direction, nominally represented by t , can give rise to additional mixing. In the following sub-sections, beam width variation, comb gap variation and mask misalignment are considered for their contributions to the mixing terms in (7.67).

7.6.1 Beam Width Variation

The cubic term in (7.67) mixes three displacement signals, all in x direction, leading to acceleration and acceleration-squared sensitivity. From Section 7.5, it is known that beam width variation leads to drive motion coupling onto the sense combs as described by (7.37), (7.38). Therefore, there are at least two components of motion in the x direction: the low frequency acceleration signal x_{As} and the drive frequency coupled signal x_{ds} . The presence of the cubic term requires another DC term in order to generate side-bands at $\omega_d \pm \omega_a$. Thus, it is seen that acceleration sensitivity due to beam width mismatch occurs only in the presence of a DC offset term x_{os} . Now, comparing the cubic term in (7.67) with (7.19) the acceleration sensitivity and the acceleration-squared sensitivity, obtained by taking the ratio of the output voltages produced by mixing to the output voltage produced by input rotation, are given by:

$$\frac{S_{ax}}{S} = \frac{6x_{os}x_{As}x_{ds}}{x_C g_s^2} \quad (7.68)$$

$$\frac{S_{a2x}}{S} = \frac{3x_{As}^2 x_{ds}}{x_C g_s^2} \quad (7.69)$$

where, S is the gyro sensitivity. x_{ds} in the above equations can arise due to width mismatch in the outer or the inner springs. The S_{a2} term is usually not very significant because the displacement produced by the acceleration x_{As} is typically an order of magnitude smaller than the offset (x_{os}), i.e., $x_{As} \ll x_{os}$. However, S_{ax} and S_{a2x} being strong

functions of the sense comb gap, they effectively constrain the smallest gap that can be used by designers. Also, $x_{As} \propto (1/\omega_{ix}^2)$ implies $S_{a2x}/S \propto 1/\omega_{ix}^2$, because $x_C \propto 1/\omega_{ix}^2$. A lower bound on ω_{ix} , while reducing the acceleration-squared sensitivity also limits the overall sensitivity of the gyroscope (from (7.15)). Maximizing sensitivity is vital to the gyroscope performance, particularly in circuit-noise limited systems. When x_{ds} is caused due to mismatch in the outer springs we have: $\left(\frac{S_{ax}}{S}\right)_{wi} = j2.87 \times 10^{-4}((^\circ/s)/g)$ and $\left(\frac{S_{a2x}}{S}\right)_{wo} = j2.67 \times 10^{-6}((^\circ/s)/g^2)$. When x_{ds} arises due to mismatch in the inner springs, $\frac{S_{ax}}{S} = (-j0.239)((^\circ/s)/g)$ and $\frac{S_{a2x}}{S} = j5.18 \times 10^{-6}((^\circ/s)/g^2)$. All the above quantities are in quadrature to the Coriolis force induced displacement because, as seen in Section 7.5, x_{ds} is in quadrature with x_C . It is also seen that the acceleration and acceleration-squared sensitivities are not very significant. For example, if there is a 1% mismatch in the inner springs, it was seen in Section 7.5 that x_{ds} is about 400 pm. Even with such a large drive motion coupling (relative to the Coriolis displacement) to the sense mode, 20g of acceleration will be required to produce a quadrature output equal in magnitude to the output due to a $1(^\circ/s)$ rotation rate. This concludes the analysis of acceleration sensitivity produced by beam width mismatch. Next the effect of comb gap variation on acceleration sensitivity is studied.

7.6.2 Comb Gap Variation

Two separate cases are considered for drive comb mismatch and sense comb mismatch; first drive comb mismatch.

If there is a relative mismatch Δ in the drive comb gap, g_d between the top and the bottom combs, the force produced in the sensing direction by the drive combs is obtained from (7.44) by replacing x_{cm} by x_{Ad} :

$$F_x = \frac{N_d \varepsilon_0 t^4 x_{Ad}}{2} \frac{3}{g_d} (3\Delta) (2o_{lpd} V_{dc} V_{ac} + y_D (V_{dc}^2 + V_{ac}^2 / 2)) \quad (7.70)$$

where x_{Ad} is the offset in the drive comb due to acceleration in the sensing direction. The outer springs are stiff in the x direction, and therefore, lead to small x_{Ad} and an even smaller response to the above force as was seen in the comparison between the displacements produced by the Coriolis forces acting on the outer frame and the inner plate in Section 7.3. Therefore, this effect is not elaborated.

Once again, acceleration sensitivity due to mismatch in the sense combs is related to *ZRO* due to sense comb mismatch. Non-identical gaps in the two sensing combs, cause a response to acceleration in the sense direction, obtained by replacing x in (7.60) by x_{As} , as:

$$\frac{V}{V_m} = \frac{4N_s \varepsilon_0 t o_{lps} x_{As}}{g_s^2} + \frac{4N_s \varepsilon_0 t y_{ds} x_{As}}{g_s^2} \Delta \quad (7.71)$$

The first term in (7.71) is the first order response of the inner accelerometer to the external acceleration. The second term mixes y_{ds} and x_{As} . Comparing the second term with the output produced by the Coriolis force in (7.19) the acceleration sensitivity is obtained as:

$$\frac{S_{ax}}{S} = \frac{y_{ds} x_{As} \Delta}{o_{lps} x_C} \quad (7.72)$$

Using (7.15), (7.57) and $x_{As} = \frac{A_x}{\omega_{ix}}$, we get:

$$\left(\frac{S_{ax}}{S}\right)_{gs} = \frac{A_x \omega_{oy} \Delta}{2\Omega_z o_{lps} \omega_{iy}^2} \left(1 - \eta_{ix}^2 + \frac{j\eta_{ix}}{Q_{ix}}\right) \quad (7.73)$$

Using standard values for all quantities: $\left(\frac{S_{ax}}{S}\right)_{gs} = -0.00101 + 0.00670j((^\circ/s)/g)$.

This effect can be reduced by increasing the mode-separation, ω_{iy}/ω_{oy} , or the overlap

length, o_{lps} . Note that (7.73) is independent of g_s and therefore, increasing the sense comb gap will not be very useful. The only effect of increasing the sense gap will be to reduce the relative mismatch Δ , if the absolute mismatch is assumed to remain constant.

It is important to recognize that the acceleration sensitivity is not very significant because very little drive motion couples to the sense combs (y_{ds}). If one side of the sense combs were anchored to the ground instead of being attached to the outer resonator, then y_{ds} in (7.72) will be equal to the drive amplitude y_D (unless there is a special decoupling suspension design as in [19]) and, as a result, the acceleration sensitivity would be much higher. This fact underscores the need for decoupling the drive oscillations from the sense combs. The decoupling is facilitated by the availability of multiple conductors in the CMOS-MEMS process.

7.6.3 Mask Misalignment

As described in Section 7.5.3, the differential sense combs become sensitive to vertical motions in the presence of a lateral offset. If the lateral offset is caused by an input acceleration, then the sense combs mix the vertical motion at drive frequency and the lateral (x) motion at low frequency giving rise to acceleration sensitivity. Assuming only parallel plate sense capacitances, the equation for acceleration sensitivity is obtained by replacing the lateral offset x_{os} in (7.63) by x_{As} , the displacement produced by external acceleration:

$$\left(\frac{S_{ax}}{S}\right)_{mm} = \frac{z_s x_{As}}{t x_C} = (0.0168 - j0.0926) \frac{(\text{°/s})}{g} \quad (7.74)$$

Using (7.15) for x_C , (7.65) for z_s and $x_{As} = \frac{A_x}{\omega_{ix}^2}$, (7.74) can be written as:

$$\frac{S_{ax}}{S} = \frac{\theta_{oyz} \omega_{oy} A_x}{2\Omega_z t \omega_{iz}^2} \frac{(1 - \eta_{ix}^2 + j\eta_{ix}/Q_{ix})}{1 - \eta_{iz}^2 - \gamma_z \eta_{iz}^2 - \eta_{oz}^2 + \eta_{iz}^2 \eta_{oz}^2} \quad (7.75)$$

The above equation suggests that the acceleration sensitivity due to mask misalignment is directly proportional to θ_{oyz} which is a measure of the drive motion coupled to the vertical axis. Furthermore, the acceleration sensitivity is also a strong function of the z resonant frequency of the inner resonator, thus providing yet another reason to design the drive mode to be well-separated and smaller than other vertical and lateral modes.

7.6.4 Summary of Acceleration Sensitivity

The three causes for acceleration sensitivity discussed above can also be viewed as mixing of the low frequency acceleration signal with drive motion coupled to sense comb in three orthogonal directions: beam width mismatch leading to x coupling, comb gap mismatch leading to sensitivity to y coupling and mask misalignment leading to z coupling. The results of acceleration sensitivity analyses reinforce some of the inferences from the *ZRO* analyses. Beam width mismatch leads to quadrature acceleration sensitivity, resulting from the cubic term of the differential comb capacitance equations. As a result beam width mismatch is not a significant cause of acceleration sensitivity. Comb gap mismatch is also not a significant cause because a relatively small fraction of the drive motion is coupled to the sense combs. Mask misalignment appears to cause considerable acceleration sensitivity. The main difference between the comb gap mismatch case and the mask misalignment case is that, in the former the large overlap length of the comb fingers lowers sensitivity to coupled drive motion, while in the latter the sensitivity to both vertical motion and Coriolis motion is linear with overlapped length of the sense comb fingers. The only design method to reduce the acceleration sensitivity, due to mask misalignment is to reduce the vertical motion coupled to the sense combs, i.e., increase the mode separation.

7.7 Cross-axis Sensitivity

Beam width and comb gap variations primarily result in in-plane elastic coupling and forces or motion sensitivities as described in the preceding sections. In order to sense out-of-plane rotations, the driven oscillations or the Coriolis force-induced oscillations need to have out-of-plane components. Therefore, sensitivity to rotation about the drive (y) or the sense (x) directions necessarily involves either out-of-plane mode coupling (elastic, electrostatic coupling or by other means) or comb sensitivity to out-of-plane motion. There-

fore, beam width and comb gap variations are not being considered in the analysis of cross-axis sensitivity. Instead, causes of out-of-plane motion and sensitivity to out-of-plane motion are investigated. Two cases are considered: rotation about the sense direction and rotation about the drive direction.

7.7.1 Rotation About Sense Axis

The gyroscope is driven into oscillations in the y direction. Just as rotation about the z axis leads to Coriolis force in the x axis, rotation about the x axis leads to Coriolis force along the z axis. The Coriolis force in z produced for Ω_x is equal to the force produced in x due to Ω_z . There are two main factors which prevent the Coriolis force due to rotation about x from being sensed. First, the sense combs are (ideally) not sensitive to vertical motion. Second, since the thickness of both the outer and the inner springs is about 2.5 times their width, the z resonant mode is higher than the x resonant mode leading to reduced Coriolis-force induced displacement.

First the z motion arising in the sense combs because of the Coriolis force is estimated. The z motion arises due to Coriolis forces acting on both the outer frame as well as the inner plate. The motion of the outer frame due to the Coriolis force acting on the outer frame is obtained by analysis of a dual mass system coupled with springs, as shown in Figure 7.6 and described in Section 7.3, given as:

$$z_{Co} = \frac{F_{Co}}{K_{zZo}} \frac{\eta_{iz}^2}{1 - \eta_{iz}^2 - \gamma_z \eta_{iz}^2 - \eta_{oz}^2 + \eta_{iz}^2 \eta_{oz}^2} \quad (7.76)$$

$$z_{Ci} = \frac{F_{Ci}}{K_{zZo} \gamma_z - \gamma_z \eta_{iz}^2 - 2\eta_{iz}^2 + \eta_{iz}^2 \eta_{oz}^2} \quad (7.77)$$

where, $\eta_{iz} = \frac{\omega_{oy}}{\omega_{iz}}$, $\omega_{iz}^2 = \frac{K_{zzi}}{M_i}$, $\eta_{oz} = \frac{\omega_{oy}}{\omega_{oz}}$, $\omega_{oz}^2 = \frac{K_{zZo}}{M_o}$ and $\gamma_z = \frac{K_{zzi}}{K_{zZo}}$. Using values

from Table 7.3, $z_{Co} = 6.16$ fm and $z_{Ci} = 102$ fm. Note that in deriving the above equations the effect of damping in the z direction has been neglected. This is indeed the case, if it is assumed that the quality factors of both the inner and outer z resonant modes are much

greater than 1 (i.e., at least 10) and also that the z modes are sufficiently separated from the drive mode (i.e., $\eta_{iz} \ll 1$ and $\eta_{oz} \ll 1$). Mode separation by a factor of 2 and minimum quality factors of about 10 are more than adequate to render the damping force at the drive frequency inconsequential in the above derivations.

If it is assumed that η_{iz} and η_{oz} are very small, then the Coriolis force induced displacement in the z direction, z_C , is dominated by z_{Ci} and can be approximately written as:

$$z_C = z_{Ci} \approx \frac{F_{Ci}}{K_{zzo}\gamma_z} \approx \frac{2\omega_{oy}y_D\Omega_x}{\omega_{iz}^2} \quad (7.78)$$

The sense combs exhibit a small sensitivity to vertical motion in the presence of a lateral offset in the combs as described in Section 7.5.3. The resultant cross-axis sensitivity is obtained by taking the ratio of output voltage produced by z_C to the nominal gyroscope output given by (7.19).

$$\frac{S_{cax}}{S} = \frac{z_C x_{os}}{t x_C} = 0.00185 + j0.000280 \quad (7.79)$$

where, the drive motion coupled to the z axis has been replaced by the Coriolis force induced displacement z_C . To get some more insight, at the expense of accuracy, substituting (7.20) and (7.78) into (7.79) when $\Omega_x = \Omega_z$:

$$\frac{S_{cax}}{S} = \frac{x_{os}\omega_{ix}^2}{t\omega_{iz}^2} \left(1 - \eta_{ix}^2 + \frac{j\eta_{ix}}{Q_{ix}} \right) \quad (7.80)$$

The above equation shows the dependence of the cross-axis sensitivity on the separation $\frac{\omega_{ix}}{\omega_{iz}}$ between the sense mode and the vertical mode. In fact given an offset x_{os} , this is the

only design option to reduce the cross-axis sensitivity. However, for the nested gyroscope it is seen that the cross-axis sensitivity to rotation about the sense axis is not very significant (about 0.2%), and therefore, does not need to be reduced.

7.7.2 Rotation About Drive Axis (Ω_y)

While rotations about the vertical axis (z) and the sense axis (x) both produce Coriolis forces when the gyroscope is oscillating along the drive axis, rotation about the drive axis itself does not produce any Coriolis force because the Coriolis force involves a cross-product of the rotation vector and the instantaneous velocity vector. On the other hand, rotations about the drive axis will lead to Coriolis force if some part of the drive motion couples to a non-drive axis. For example, if the drive motion is coupled to the z direction due to mask misalignment, Ω_y leads to a Coriolis force in x , the sense direction. Drive motion coupled to the x axis is not considered because that will result in a Coriolis force that induces vibrations in z . Sensing these vibrations involves cascading of two non-ideal effects: drive motion coupling to x axis and sense combs producing differential output due to vertical motion. Therefore, drive motion coupled to x axis is expected to be less significant than drive motion coupled to z axis for purposes of cross-axis sensitivity. The details of sensitivity to rotation about the drive axis due to drive motion coupling to the z axis, are presented below.

As shown in Section 7.3, the Coriolis force acting on the outer resonator rigid frame contributes insignificantly to the total Coriolis force induced relative displacement in x between the inner accelerometer plate and the outer resonator rigid frame. Therefore, only the effect of Coriolis force acting on the inner accelerometer plate is taken into account. The vertical displacement of the plate due to motion coupled from the drive mode is obtained by solving a dual mass spring system as shown in Figure 7.6 where a coupling force $F_z = K_{yzo}y_D$ acts on the outer frame:

$$z_{di} = \frac{K_{yzo}y_D}{K_{zZo}(1 - \eta_{oz}^2 - \eta_{iz}^2 - \gamma_z \eta_{iz}^2 + \eta_{iz}^2 \eta_{oz}^2)} = -j3.39 \text{ nm} \quad (7.81)$$

wherein a $0.1 \mu\text{m}$ mask misalignment has been assumed. Note that if the inner springs are highly rigid in z (i.e., $\eta_{iz} \ll 1$) then the above equation reduces to the case where the outer frame and the inner plate move in unison vertically. The normalized cross-axis sensitivity is obtained directly as a ratio of the Coriolis forces acting on the inner accelerometer plate

arising due to rotations about y and z . The Coriolis forces are, in turn, respectively proportional to amplitude of driven oscillations of the inner accelerometer plate in the z and the y directions. Thus,

$$\frac{S_{cay}}{S} = \frac{z_{di}}{y_D} = \frac{K_{yzo}}{K_{zz}(1 - \eta_{oz}^2 - \eta_{iz}^2 - \gamma_z \eta_{iz}^2 - \eta_{iz}^2 \eta_{oz}^2)} = 0.00243 \quad (7.82)$$

If the springs are much stiffer in z than in y a much simpler, approximate expression can be written:

$$\frac{S_{cay}}{S} \approx \theta_{oyz} \quad (7.83)$$

where θ_{oyz} is the angle through which the principal axes of stiffness of the outer resonator have rotated to produce coupling between the y and the z modes. From (7.82) and (7.83) it can be concluded that the normalized cross-axis sensitivity to rotations about the drive axis is directly proportional to the angle by which the principal axes of stiffness of the outer spring beams have rotated due to mask misalignment. From the denominator of (7.82), the importance of separation between the vertical modes and the drive mode is yet again seen. Similar to S_{cax} , S_{cay} is found to be insignificant (about 0.25%).

7.7.3 Summary of Cross-axis Sensitivity

Response of the vertical axis gyroscope to rotations about both orthogonal axes, x and y have been discussed in this section. While the former is seen to occur due to sensitivity of the drive comb to vertical oscillations the latter occurs due to coupling of drive motion to vertical axis. Both the cross-axis sensitivities are found to be less than 1% for expected values of comb offset and mask misalignment. In a way, this is an expected result. The displacements arising from Coriolis forces in off-axis directions (x, y) are relatively so small that there is little possibility of them interfering with the main axis sensitivity.

7.8 Simulation Results

In the previous sections, non-ideal manufacturing effects have been analyzed in detail, and correlated to Zero Rate Output, acceleration and acceleration-squared sensitiv-

ity and cross-axis sensitivity. In the first part of this section, results of NODAS behavioral simulation for each non-ideal case analyzed before are presented and compared with the analytical derivations. Non-idealities are considered only one at a time in the first part. In reality, all the non-ideal manufacturing effects occur in collusion, rendering hand analysis not only tedious, but also unfruitful in terms of intuitive understanding. In order to obtain a complete picture of manufacturing effects on the gyroscope, Monte-Carlo simulations are employed in the second part of this section. Monte-Carlo simulations have been used extensively in a variety of fields to understand effects of multiple, randomly varying factors [96].

In order to optimize the simulation time, three different schematics for the gyroscope are used to capture individual effects. The three schematics in increasing order of simulation time are:

1. 2D schematic for beam width and gap mismatch where there is no out-of-plane motion coupling.
2. 3D schematic for gap mismatch case requiring non-zero sensitivity of differential combs to vertical motion.
3. 3D schematic with detailed beam model capturing effects of metal mask misalignment.

It should be noted that all the simulations can be done with the third schematic listed above, at the expense of increased simulation time. The netlist for the 2D schematic is given in Appendix A7.

7.8.1 Mismatch Simulation Results

Comparison between the analytically derived numbers and those obtained from simulation are shown in Table 7.5, Table 7.6 and Table 7.7 for the *ZRO*, acceleration sensitivity

Table 7.5 Comparison between analytical calculations and NODAS simulations of *ZRO*

Mismatch case	Analytical ($^{\circ}/s$)	Simulation ($^{\circ}/s$)
Asymmetric Drive	37.2	41.4
Outer spring width mismatch	0.845	2.85
Inner spring width mismatch	645	653
Drive comb gap mismatch	2.25	2.73

Table 7.5 Comparison between analytical calculations and NODAS simulations of *ZRO*

Mismatch case	Analytical ($^{\circ}/s$)	Simulation ($^{\circ}/s$)
Sense comb gap mismatch	0.338	0.400
Mask misalignment	4.70	4.42

Table 7.6 Comparison between analytical calculations and NODAS simulations of acceleration sensitivity

Mismatch case	Analytical ($\times 10^{-3} \text{ }^{\circ}/s/g$)	Simulation ($\times 10^{-3} \text{ }^{\circ}/s/g$)
Outer spring width mismatch	0.287	0.512
Inner spring width mismatch	0.239	0.237
Drive comb gap mismatch	0.796	1.05
Sense comb gap mismatch	6.78	5.42
Mask misalignment	94.1	69.0

Table 7.7 Comparison between analytical calculations and NODAS simulations of cross-axis sensitivity

Mismatch case	Analytical ($\times 10^{-3}$)	Simulation ($\times 10^{-3}$)
Rotation about sense	1.87	1.44
Rotation about drive	2.43	2.34

and cross-axis sensitivity respectively. In the *ZRO* table, the simulation values are fairly close to the hand analysis values except for the case of width mismatch in the outer springs. Upon detailed investigation of the simulation results, this discrepancy was related to the effect of non-zero mass of the spring beams. When the width of one out of the four outer springs is changed by 1%, the effective mass of the spring also changes resulting in an asymmetric mass distribution for the outer resonator. This asymmetric mass distribution is not accounted for in the analysis presented, but is captured by the NODAS behavioral simulations. A similar discrepancy is also seen in the acceleration sensitivity estimates from analysis and simulations. Among the other results, the difference between the analytical and simulation results is somewhat higher for the two cases of gap mismatch for both the *ZRO* and the acceleration sensitivity. This higher difference can be attributed to the fact that the fringe capacitance and force was neglected in the hand analysis. An additional effect not considered in the hand analysis, is the different response of the sense

accelerometer at frequencies at $\omega_d - \omega_a$ and $\omega_d + \omega_a$, because the sense mode resonant frequency is closer to $\omega_d + \omega_a$ than to $\omega_d - \omega_a$. Following the above discussion, it can be summarized that the hand analysis provides adequate estimates of the non-idealities as well as insights into the causes for designers. However, behavioral simulation yields more accurate estimates of non-idealities and will, almost always, capture additional non-ideal effects than hand analysis. Therefore, behavioral simulation must be an integral part of the design process.

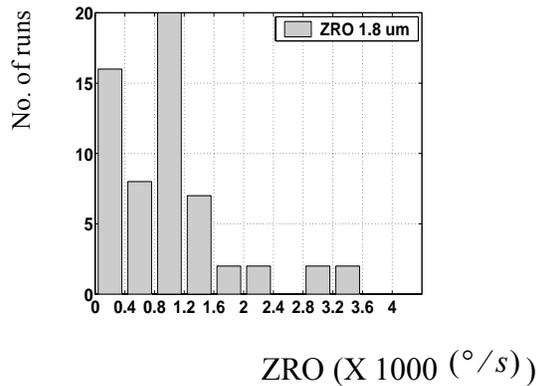
7.8.2 Monte-Carlo Simulations

Monte-Carlo analysis has been traditionally used by analog circuit designers to study the effect of manufacturing variations in the threshold voltage, gate-oxide thickness and other device parameters on the circuit performance. A similar technique is applied here for gyroscope simulations with twelve randomly varied geometrical parameters. The four outer spring beam widths, four inner springs beam widths, two drive comb gaps and the two sense comb gaps comprise the twelve randomly varied parameters. As in the preceding hand analysis, it is assumed that the beam widths (or gap) in a given spring (comb) are uniform within the spring (comb). However, each spring (comb) is considered independent. This assumption attempts to capture non-local variations in the beam widths and the comb gaps.

The nominal gyroscope design uses 1.8 μm beam widths and comb gaps. The analysis for the acceleration and acceleration-squared sensitivities in Section 7.6 suggests that larger gaps and larger sense mode resonant frequency will lead to lower acceleration sensitivity. Monte-Carlo simulations using the NODAS schematic described in Section 7.2 and shown in Figure 7.2(b) were done with nominal beam widths and gaps of 1.8 μm and 2.0 μm . In each case, beam widths in the eight springs and the gaps in the four combs were assumed to be independent, normally distributed random variables ($N(w, \sigma)$) with common mean w , equal to the layout dimension (1.8 or 2.0 μm), and standard deviation σ ($3\sigma = 0.05 \mu\text{m}$). Each Monte-Carlo analysis involves 59 transient analysis, with the 12 randomly generated dimensions $\sim N(w, \sigma)$. The 59 sets of 12 randomly distributed dimensions are listed in Appendix A6. Prior to the each transient analysis, ac analysis was performed in order to obtain the y resonant frequency of the outer frame, which was then used

Mean = 980 ($^{\circ}/s$)

$\sigma = 783$ ($^{\circ}/s$)



Mean = 1110 ($^{\circ}/s$)

$\sigma = 881$ ($^{\circ}/s$)

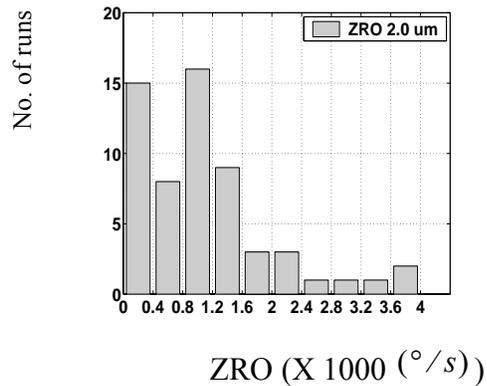


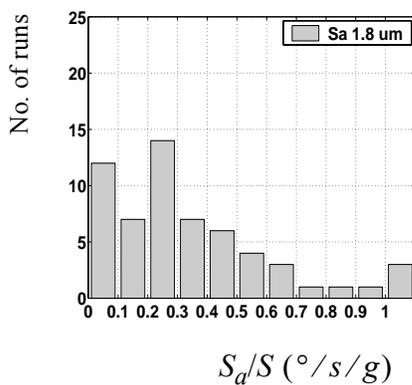
FIGURE 7.14. ZRO Histograms for widths and gaps = 1.8 μm and 2.0 μm from Monte-Carlo simulation.

as the drive frequency in the transient analysis. The OCEAN scripts used to run the ac and transient analysis are listed in Appendix A7.

The Zero Rate Output (*ZRO*), acceleration sensitivity (S_{ax}) and acceleration-squared sensitivity (S_{a2x}) for the 1.8 and 2.0 μm designs are shown in Figure 7.14, Figure 7.15. and Figure 7.16. The prominence of the third bin is nothing more than the fact that the number of runs in each bin is not too high. If key statistical measures such as the standard deviation of the *ZRO* do not change significantly as the number of Monte-Carlo simula-

Mean = 0.35 ($^{\circ}/s/g$)

$\sigma = 0.28$ ($^{\circ}/s/g$)



Mean = 0.19 ($^{\circ}/s/g$)

$\sigma = 0.16$ ($^{\circ}/s/g$)

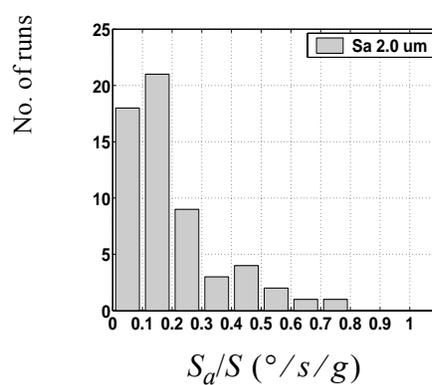


FIGURE 7.15. Acceleration sensitivity histograms for widths and gaps = 1.8 μm and 2.0 μm from Monte-Carlo simulation.

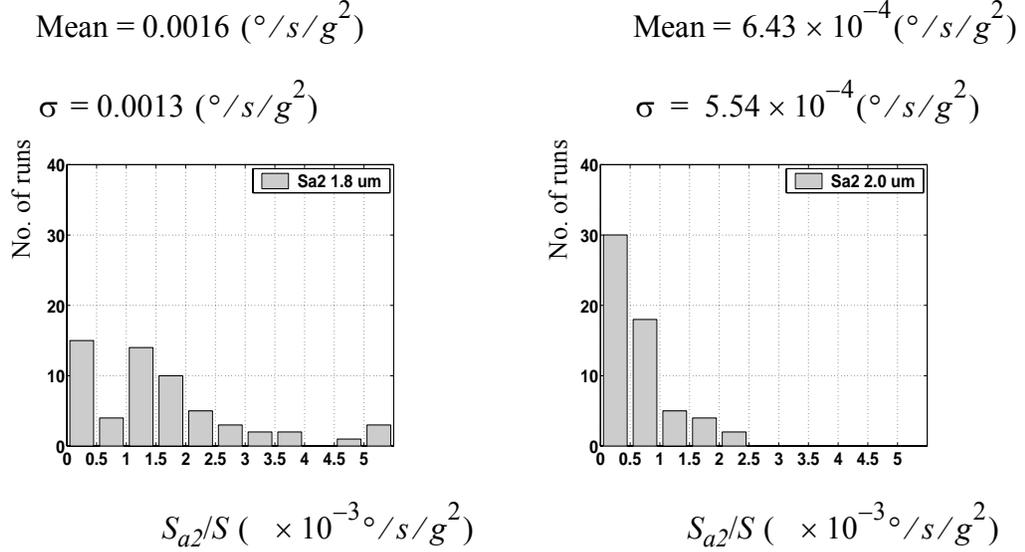


FIGURE 7.16. Acceleration-squared sensitivity histograms for widths and gaps = 1.8 μm and 2.0 μm from Monte-Carlo simulation.

tions increases, then the number of runs can be considered to be adequate. The standard deviation of the ZRO for the first 30 runs is $781^{\circ}/s$ and that for the remaining 29 runs is $787^{\circ}/s$. Since there is a difference of less than 1% in the two standard deviations, the number of simulations runs can be considered to be sufficient. Since the main contribution to ZRO is the asymmetric topology of the inner springs, it is not affected by larger width and gap. To reduce the ZRO alternate solutions have to be adopted. Simulations with symmetric inner springs resulted in ZRO reduction of about 90% and greatly reduced offsets due to lateral curling. The simulation values for cross-axis sensitivities are about 100 times smaller than the gyro sensitivity. Increased width and gaps do not have significant impact on cross-axis sensitivity because, the cross-axis sensitivity is mainly dependent on out-of-plane resonant modes and comb sensitivities. Now, the differences between the 1.8 μm and the 2.0 μm designs in acceleration sensitivity are discussed.

For equal drive displacements, the nominal gyro sensitivity S for the 2.0 μm design is smaller by about 35%. However, both the normalized acceleration sensitivity S_{ax}/S and the normalized acceleration-squared sensitivity S_{a2x}/S have also reduced significantly, as expected. The mean and standard deviation of S_{ax}/S reduce by about 45% from $0.35^{\circ}/s/g$ to $0.19^{\circ}/s/g$ and from $0.28^{\circ}/s/g$ to $0.16^{\circ}/s/g$ respectively. The absolute

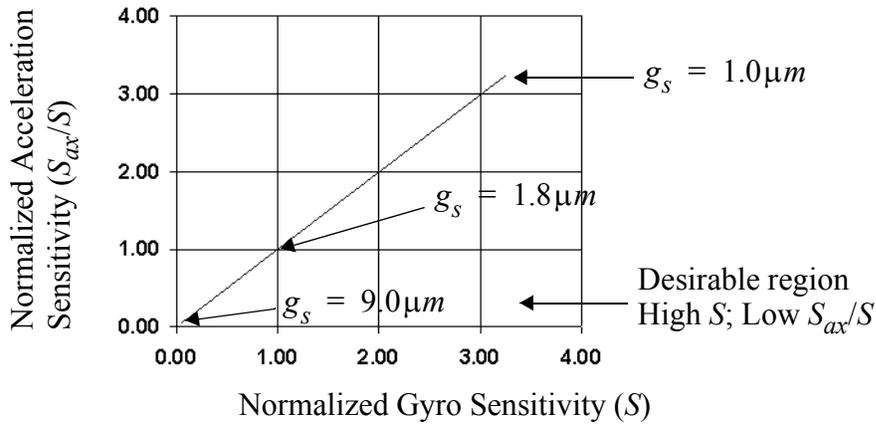


FIGURE 7.17. Trade-off between gyroscope sensitivity and acceleration sensitivity with varying gap

reduction in S_{ax} is greater than 60%. Thus, it is seen that increasing the widths and the gaps results in significant improvement in the acceleration and acceleration-squared sensitivity. The analytical equations derived earlier can be used to optimize the gyro sensitivity and the acceleration sensitivity to a required ratio for a given set of manufacturing variations. For instance, (7.68) indicates that, with all other things kept constant, the normalized acceleration sensitivity goes as $1/g_s^2$. It should, however, be noted that the gyro sensitivity also reduces with increasing gap. The ideal choice of gap for a specific application can be made using the trade-off between decreasing acceleration sensitivity and decreasing gyro sensitivity as shown in Figure 7.17. In the figure, the acceleration sensitivity is plotted against the gyroscope sensitivity, both being normalized to the respective values at the sense gap $g_s = 1.8\mu m$. The desirable gyroscope has high sensitivity to angular velocity and low sensitivity to linear acceleration. However, the plot shows that this cannot be achieved by changing the sense gap. Gyroscope sensitivity has to be sacrificed in order to obtain better rejection of linear acceleration due to non-linearity in the differential sense combs.

From the analysis and the simulations presented in the preceding sections the following conclusions for the ZRO and S_{ax} can be deduced:

Zero Rate Output:

1. Use symmetric springs only

2. Choose spring topology and position the springs to minimize elastic coupling

Acceleration Sensitivity:

1. Larger comb gaps are better
2. Higher sense mode resonant frequency and larger gaps are better for reduced acceleration-squared sensitivity,

Both the above approaches lead to reduced sensitivity. However, the acceleration rejection obtained is greater than the sensitivity loss. Furthermore, analysis (equation (7.72) in Section 7.6.2) suggests that decoupling of drive vibration from sense combs can reduce linear acceleration sensitivity significantly and potentially eliminate the need for dual anti-phase gyroscopes. Cross-axis sensitivities are found to be about 100 times smaller than the gyro sensitivity and are not strongly dependent on the beam widths and the gaps unlike the acceleration sensitivity.

7.9 Summary

Detailed analyses of microgyroscope non-idealities caused by three kinds of manufacturing effects have been presented in this chapter. The manufacturing effects considered are: beam width variation, comb gap variation and mask misalignment. The gyroscope non-idealities discussed are the Zero Rate Output, acceleration and acceleration-squared sensitivity and the cross-axis sensitivity. The analyses enable a qualitative comparison of CMOS-MEMS and single-layer gyroscope topologies. The necessity of complete symmetry in design (e.g., symmetric spring topologies) as well as operational aspects (e.g., symmetric gyroscope drive) has been clearly brought out. Mode-separation, which has always been considered important in MEMS design, has been quantitatively linked to gyroscope non-idealities. The equations derived to explain gyroscope non-idealities can serve as examples for future work in treatment of other manufacturing effects. Monte-Carlo simulations have been used to verify the analyses and enhance understanding of design trade-offs between nominal performance and the ability to reject non-ideal variations. They can also be used to estimate manufacturing yield for given beam width and gap variations and alignment tolerances.

Chapter 8. Summary and Future Work

8.1 Thesis Summary and Contributions

The thesis examines several important aspects of MEMS modeling, simulation and design on the way to understanding of gyroscope non-idealities. Elastic cross-axis coupling, temperature dependent curl macromodels and rapid computation of spring stiffness constants, modeling method for CMOS-MEMS combs and difficulties in MEMS behavioral simulations have been addressed. The chapter on gyroscope analysis and simulation utilizes the models and solutions described earlier to derive detailed equations correlating gyroscope non-idealities to manufacturing variations.

The contributions of this work can be broadly classified in three fundamental directions: modeling, simulation and design. Each direction is elaborated in the following subsections.

8.1.1 Modeling

Modeling of phenomena in two physical domains, elastic and electrostatic, has been done in this thesis. The two domains required different approaches. Elastic models were derived from a purely analytical viewpoint, with extensive verification through finite element analysis. Elastic models for micromechanical springs, have previously been mostly restricted to main axis stiffness terms. The analytical treatment presented in this thesis led to fundamental understanding of elastic cross-axis coupling effects at the system-level and for individual springs such as crab-leg, u-spring and serpentine shaped springs. The important result that in-plane translational cross-axis coupling can be eliminated through symmetric spring design was also proved formally. Equations for out-of-plane cross-axis coupling in CMOS-MEMS due to misaligned metal layers were derived. A computational technique for rapid computation of spring stiffness matrices for single-chain-of-beams springs has been presented. Extensions to arbitrary spring topologies as well as inclusion of viscous and inertial effects have been suggested. Temperature-dependent curl macro-model for CMOS-MEMS beams was developed and implemented in the NODAS frame-

work. Test structures were characterized experimentally to verify the vertical and lateral curvature predicted by the macromodel.

Modeling in the electrostatic domain has been tackled by using a combination of analytical and numerical techniques. The method used is similar to transistor models, wherein, each new process requires several device-level field solutions in order to fit the model parameters. Curling of comb fingers, multi-layer cross-section, corner effects and arbitrary motion of combs, which are almost impossible to capture analytically, have been captured in the model through this method. Furthermore, the model ensures energy conservation between different domains through combined solution for the capacitance and force fitting coefficients. Experimental measurements were made on variable capacitance structures for model verification.

8.1.2 Simulation

A fundamental understanding of transient analysis with MEMS component models encoded in an Analog Hardware Description Language (AHDL) was obtained by extending time-discretized interpretation of electrical components such as resistors, inductors and capacitors to the mechanical domain. Pointers for AHDL implementation with better simulation convergence and speed were deduced. Important guidelines to be considered during AHDL modeling are: minimize the number of additional internal states during model development, minimize the quantity and size of off-diagonal elements in the simulation matrix and use scaling in multi-domain simulations. The importance of the above pointers cannot be overemphasized; gyroscope simulations reported in Chapter 7 do not converge without making use of the above guidelines.

8.1.3 Gyroscope Design

A number of significant insights into gyroscope design were obtained following the analyses and simulations reported in Chapter 7, which represent the first detailed study of intra-die manufacturing variations and misalignments on the Zero Rate Output, acceleration sensitivity and cross-axis sensitivity of a microgyroscope. Geometrical and operational symmetry were found to be crucial in reducing gyroscope non-idealities. Importance of double-decoupling: i.e., sense mode decoupled from drive combs and drive mode decoupled from sense modes was also quantified. The nested gyroscope topology in

the multi-conductor CMOS-MEMS process particularly embodies the double-decoupling concept. The effect of lateral offsets in drive comb and sense comb, when combined with vertical coupling of drive motions was shown to cause all the three non-idealities under consideration. The mode-separation requirement was quantified by analytically linking it to gyroscope non-idealities. Monte-Carlo simulations demonstrated the trade-offs between nominal gyroscope performance and reduction of gyroscope non-idealities.

8.2 Future Directions of Work

Experimental evidence of the analysis presented in this thesis is required in order to establish leading causes of gyroscope non-idealities. Following measurements of non-idealities, design techniques to further reduce the non-idealities need to be explored. Existing commercial microgyroscopes [21][28][29] use two proof-masses which are driven anti-phase in order to cancel out acceleration sensitivity to the first order. This entails almost a 2X increase in the MEMS area of the chip. Gyroscope designs with double-decoupling may potentially eliminate the need for dual proof-masses in single-layer processes. Since the acceleration sensitivity arises due to sense comb non-linearity, linear sensing mechanisms such as a simple comb may also be used to suppress it, with the side-effect of reduced gyroscope sensitivity. Additionally, the low-frequency acceleration signal provided by the inner accelerometer in the gyroscope maybe used to provide electronic or electromechanical cancellation of the acceleration sensitivity. The area efficiency of such a solution remains to be compared with the use of dual anti-phase proof-masses.

Based on the simulations presented in this thesis, a set of basic benchmark simulations to be performed on any gyroscope design can be readily constructed. Optimizing gyroscope performance by trading off sensitivity for robustness will be facilitated by such a benchmark suite. This will not only be useful in comparing gyroscope topologies, but can also help in potentially automating parts of the optimization.

Automated simulation-based optimization has already been demonstrated commercially for analog systems [97] and in academia for micromachined accelerometers [60]. Having obtained a basic understanding of gyroscope non-idealities due to manufacturing effects, we can now look at automating gyroscope size optimization. Model-order reduction techniques proposed in Chapter 4 and by others [98], can be combined with the simu-

lation-based synthesis tools to facilitate rapid sizing of microgyroscopes. Synthesis tools can illuminate trade-offs between various performance metrics such as gyroscope area, operating frequency, sensitivity, resolution, bandwidth, Zero Rate Output, acceleration sensitivity.

References

- [1] M. Madou, *Fundamentals of Microfabrication*, CRC Press, New York, 1997
- [2] N. Yazdi, F. Ayazi, and K. Najafi, "Micromachined Inertial Sensors," *Proc. IEEE*, vol. 86, no. 8, pp. 1640-1659, Aug. 1998
- [3] *MEMS 1999: Emerging Applications and Markets*, System Planning Corporation, 1999
- [4] J. Soderkvist, "Micromachined Gyroscopes," *Sensors and Actuators A*, vol. 43, pp. 65-71, 1994
- [5] J. A. Geen, "A Path to Low Cost Gyroscopy," *Tech. Dig. Solid-State Sensors and Actuators*, 51-54, Hilton Head, SC, June 8-11, 1998
- [6] T. B. Gabrielson, "Mechanical-Thermal Noise in Micromachined Acoustic and Vibration Sensors," *IEEE Tran. Electron Devices*, vol. 40, no. 5, May 1993
- [7] G. K. Fedder, S. Santhanam, M. L. Reed, S. C. Eagle, D. F. Guillou, M. S.-C. Lu, and L. R. Carley, "Laminated High-Aspect-Ratio Microstructures in a Conventional CMOS Process," *Sensors and Actuators A*, vol. A57, no. 2, pp. 103-110, 1996
- [8] *IEEE Standard for Inertial Sensor Terminology*, IEEE Std 528-1994
- [9] A. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*, Progress in Astronautics and Aeronautics, volume 174, 1997
- [10] H. Luo, G. K. Fedder, L. R. Carley, "An Elastically Gimbaled Z-Axis CMOS-MEMS Gyroscope," *Int. Sym. Smart Structures & Microsystems*, Hong Kong, Oct. 19-21, 2000
- [11] T. Mukherjee, "CAD for Integrated MEMS Design," *Proc. Design, Test, Integration and Packaging of MEMS/MOEMS*, Paris, France, May 9-11, 2000
- [12] G. K. Fedder and Q. Jing, "A Hierarchical Circuit-level Design Methodology for Microelectromechanical Systems," *IEEE Tran. Cir. and Systems II*, vol. 46, no. 10, pp. 1309-1315, 1999
- [13] P. Greiff, B. Boxenhorn, T. King, L. Niles, "Silicon monolithic micromechanical gyroscope," *Tech. Dig. 6th Int. Conf. Solid-state Sensors and Actuators (TRANSDUCERS '91)*, 955-968, San Francisco, CA, USA; 24-27 June 1991

- [14] J. Bernstein, S. Cho, A. T. King, A. Kourepenis, P. Maciel, and M. Weinberg, "A Micromachined Comb-Drive Tuning Fork Rate Gyroscope," *Proceedings. IEEE MEMS Workshop (MEMS '93)*, Fort Lauderdale, FL, 143-148, 7-10 Feb. 1993
- [15] F. Gretillat, M-A Gretillat, N. F. de Rooij, "Improved Design of a Silicon Micromachined Gyroscope with Piezoresistive Detection and Electromagnetic Actuation," *IEEE Journal of Microelectromechanical Systems*, vol. 8, no. 3, pp. 243-250, September 1999
- [16] R. L. Kubena, D. J. Vickers-Kirby, R. J. Joyce, F. P. Stratton, and D. T. Chang, "A New Tunneling-based Sensor for Inertial Rotation Rate Measurements," *Sensors and Actuators A*, 109-117, vol. 83, 2000
- [17] O. Bochobza-Degani, D. J. Seter, E. Socher, and Y. Nemirovsky, "A Novel Micromachined Vibrating Rate-Gyroscope with Optical Sensing and Electrostatic Actuation," *Sensors and Actuators A*, 54-60, vol. 83, 2000
- [18] M. W. Putty and K. Najafi, "A Micromachined Vibrating Ring Gyroscope," *Tech. Dig. Solid-State Sensors and Actuators Workshop*, 213-220, Hilton Head, SC, June 13-16, 1994
- [19] W. A. Clark, R. T. Howe, and R. Horowitz, "Surface Micromachined Z-Axis Vibratory Rate Gyroscope," *Tech. Dig. Solid-State Sensors and Actuators Workshop*, 283-287, Hilton Head, SC, June 1996
- [20] K. Y. Park, C. W. Lee, Y. S. Oh and Y. H. Cho, "Laterally Oscillated and force-balanced Micro Vibratory Gyroscope Supported by Fish Hook Shape Springs," *Proc. IEEE MEMS Workshop (MEMS '97)*, 272-277, Japan, 1997
- [21] M. Lutz, W. Golderer, J. Gerstenmeier, J. Marek, B. Maihofer, S. Mahler, H. Munzel and U. Bischof, "A Precision Yaw Rate Sensor in Silicon Micromachining," *Tech. Dig. 9th Int. Conf. Solid-state Sensors and Actuators (Transducers '97)*, 847- 850, Chicago, Il, June 16-19, 1997
- [22] F. Ayazi and K. Najafi, "Design and Fabrication of a High-Performance Polysilicon Vibrating Ring Gyroscope," *Proc. 11th IEEE Int. Conf. MEMS (MEMS '98)*, 621-626, January 1998

- [23] S. S. Baek, Y. S. Oh, B. J. Ha, S. D. An, B. H. An, H. Song, and C. M. Song, "A Symmetrical Z-Axis Gyroscope with a High Aspect Ratio Using Simple and New Process," *Proc. 12th IEEE Int. Conf. MEMS (MEMS '99)*, 612-617, Orlando, Florida, January 17-21, 1999
- [24] H. Song, Y. S. Oh, I. S. Song, S. J. Kang, S. O. Choi, H. C. Kim, B. J. Ha, S. S. Baek, and C. M. Song, "Wafer Level Vacuum Packaged De-coupled Vertical Gyroscope by a New Fabrication Process," *Proc. 13th IEEE Int. Conf. MEMS (MEMS '00)*, 520-524, Miyazaki, Japan, January 23-27, 2000
- [25] X. Jiang, J. I. Seeger, M. Kraft, and B. E. Boser, "A Monolithic Surface Micromachined Z-Axis Gyroscope with Digital Output," *Tech. Dig. IEEE Symposium on VLSI Circuits*, 16-19, Honolulu, Hawaii, June 15-17, 2000
- [26] S. E. Alper and T. Akin, "A Symmetric Surface Micromachined Gyroscope With Decoupled Oscillation Modes," *Proc. 11th Int. Conf. Solid-State Sensors and Actuators (Transducers '01)*, Munich, Germany, June 10-14, 2001
- [27] A. A. Seshia, R. T. Howe, and S. Montague, "An Integrated Microelectromechanical Resonant Output Gyroscope," *Proc. 15th IEEE Int. Conf. MEMS (MEMS '02)*, 722-726, Las Vegas, NV, January 20-24, 2002
- [28] J. A. Geen, S. Sherman, J. F. Chang and S. R. Lewis, "Single-chip Surface-Micromachined Integrated Gyroscope with 50° / hour Root Allan variance," *J. Solid-State Circuits*, pp. 1860-1866, vol. 37, no. 12, December 2002
- [29] Analog Devices Inc. ADXRS150 gyroscope datasheet
- [30] H. Xie and G. K. Fedder, "A Lateral Axis CMOS-MEMS Gyroscope," *Proc. 14th Int. Conf. on MEMS (MEMS 2001)*, Interlaken, Switzerland, January 21-25, 2001
- [31] H. Xie, *Gyroscope and Micromirror Design using Vertical-Axis CMOS-MEMS Actuation and Sensing*, Ph. D. Thesis, ECE Department, Carnegie Mellon University, May 2002
- [32] M. Kranz, G. K. Fedder, "Micromechanical Vibratory Rate Gyroscopes Fabricated in Conventional CMOS," *Symposium on Gyro Technology*, Stuttgart, Germany, 3.0-3.8, 1997
- [33] P. Yoon, *Canonical Micro Gyroscope for Design Tool Verification*, M. S. Thesis, ECE Department, Carnegie Mellon University, May 2000

- [34] H. Luo, *Integrated Multiple Device CMOS-MEMS IMU Systems and RF MEMS Applications*, Ph. D. Thesis, ECE Department, Carnegie Mellon University, December 2002
- [35] X. Zhu, *Post-CMOS Micromachining of Surface and Bulk Structures*, Ph.D. Thesis, ECE Department, Carnegie Mellon University, May 2002
- [36] G. Zhang, H. Xie, L. E. Derosset, G. K. Fedder, "A Lateral Capacitive CMOS Accelerometer with Structural Curl Compensation," *Proc. 12th IEEE Conf. MEMS (MEMS '99)*, 606-611, Orlando, Florida, January 17-21, 1999
- [37] H. Lakdawala, *Temperature Control of CMOS Micromachined Sensors*, Ph. D. Thesis, ECE Department, Carnegie Mellon University, May 2002
- [38] C. H. Mastrangelo and W. C. Tang, "Semiconductor Sensor Technologies," in *Semiconductor Sensors*, Ed. S. M. Sze, John Wiley & Sons, New York, 1994
- [39] J. M. Gere and S. P. Timoshenko, *Mechanics of Materials*, PWS Publishing Company, Boston, 4th edition, 1997
- [40] Q. Jing, *Modeling and Simulation for Design of Suspended MEMS*, Ph. D. Thesis, ECE Department, Carnegie Mellon University, May 2003
- [41] G. K. Fedder, "Simulation of Microelectromechanical Systems," *Ph. D. Thesis*, EECS Department, University of California at Berkeley, September 1994
- [42] W. A. Johnson and L. K. Warne, "Electrophysics of Micromechanical Comb Actuators," *J. Microelectromechanical Systems*, pp. 49-59, vol. 4, no. 1, March 1995
- [43] Z. Nehari, *Conformal Mapping*, Dover Publications, New York, 1975
- [44] T. Mukherjee, Y. Zhou, and G. K. Fedder, "Automated Optimal Synthesis of Microaccelerometers," *Proc. 12th IEEE Int. Conf. MEMS (MEMS '99)*, 326-331, Orlando, Florida, January 1999,
- [45] W. O. Davis, and A. P. Pisano, "On the Vibrations of a MEMS Gyroscope," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '98)*, Santa Clara, CA, pp. 557-562, April 6-8 1998
- [46] Y. Ansel, Ph. Lerch, and Ph. Renaud, "Mode coupling aspects in a vibrating gyroscope," *Sensors and Actuators A* 62, pp. 576-581, 1997

- [47] Y. Mochida, M. Tamura, and K. Ohwada, "A Micromachined Vibrating Rate Gyroscope with Independent Beams for the Drive and Detection Modes," *Proc. IEEE MEMS Workshop (MEMS '99)*, 618-623, Orlando, Florida, January 17-21, 1999
- [48] Mathematica Reference Manual, Wolfram Research Inc., Champaign, Illinois, <http://www.wolfram.com>
- [49] ABAQUS Reference Manual, HKS Inc., Rhode Island
- [50] J. Vandemeer, M. S. Kranz, and G. K. Fedder, "Hierarchical Representation and Simulation of Micromachined Inertial Sensors," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '98)*, Santa Clara, CA, pp. 540-545, April 6-8 1998
- [51] G. Lorenz, *Network Simulation of Microelectromechanical Systems*, Ph. D. Thesis, University of Bremen, December 1999
- [52] A. Shkel and R. Howe, "Modeling and Simulation of Micromachined Gyroscopes in the Presence of Imperfections," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '99)*, 605-608, San Juan, Puerto Rico, April 19-21, 1999
- [53] S. Iyer, Y. Zhou, and T. Mukherjee, "Analytical Modeling of Cross-axis Coupling in Micromechanical Springs," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '99)*, 632-635, San Juan, Puerto Rico, April 19-21, 1999
- [54] Coventorware (previously MEMCAD) User's Manual, Coventor, Inc., Cary, NC 27513, <http://www.coventor.com>
- [55] D. Moulinier, P. Metsu, M.-P. Brutaills, S. Bergeon, and P. Nachtergaele, "MEMS-Master: A New Approach to Prototype MEMS," *Proc. Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP 2001)*, 165-174, Cannes, France, April 25-27, 2001
- [56] S. Timoshenko, "Analysis of Bi-metal Thermostats," *J. Optical Society of America*, 11, pp. 233-255, 1925
- [57] D. L. DeVoe, et al "Modeling and Optimal Design of Piezoelectric Cantilever Microactuators," *Journal of Microelectromechanical Systems*, vol 6, no. 3, Sept. 1997
- [58] H. Lakdawala and G. K. Fedder, "Analysis of Temperature-Dependent Residual Stress Gradients in CMOS Micromachined Structures," *Tech. Dig. 10th Int. Conf. Solid-State Sensors and Actuators (Transducers '99)*, 526-529, Sendai, June 1999

- [59] D. M. Freeman, A. J. Aranyosi, M. J. Gordon, and S. S. Hong, "Multidimensional motion analysis of MEMS using computer microvision," *Tech. Dig. Solid-State Sensor and Actuator Workshop*, Hilton Head, SC, 1998
- [60] V. Gupta, *Layout Synthesis of CMOS-MEMS Accelerometers*, M. S. Thesis, ECE Department, Carnegie Mellon University, May 2000
- [61] J. S. Przemieniecki, *Theory of Matrix Structural Analysis*, Dover Publishing Inc., New York, 1985
- [62] B. F. Romanowicz, *Methodology for the Modeling and Simulation of Microsystems*, Kluwer Academic Publishers, Boston, 1998
- [63] S. V. Iyer and T. Mukherjee, "Numerical Spring Models for Behavioral Simulation of MEMS Inertial Sensors," *Proc. Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP 2000)*, 55-62, Paris, France, May 9-11, 2000
- [64] B. Baidya, S.K. Gupta and T. Mukherjee, "MEMS Component Extraction," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '99)*, San Juan, Puerto Rico, April 19-21, 143-146, 1999
- [65] A. Odabasioglu, M. Celik, L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 8, pp. 645-654, Aug. 1998
- [66] D. Ramaswamy and J. White, "Automatic Generation of Small-signal Dynamic Macromodels from 3-D Simulation," *Proc. Int. Conf. Modeling and Simulation of Microsystems 2001 (MSM '01)*, 27-30, Hilton Head Island, SC, March 19-21, 2001
- [67] W. C. Tang, T. -C. H. Nguyen, M. W. Judy, and R. T. Howe, "Electrostatic-comb Drive of Lateral Polysilicon Resonators," *Tech. Dig. 5th Int. Conf. Solid-State Sensors and Actuators (Transducers '89)*, Montreux, vol. 2, pp. 328-331, June 25-30 1989
- [68] S. D. Senturia, "CAD Challenges for Microsensors, Microactuators, and Microsystems," *Proc. IEEE*, vol. 86, pp. 1611-1626, Aug. 1998
- [69] H. Lakdawala and G. K. Fedder, "Temperature Control of CMOS micromachined sensors," *Proc. 15th IEEE Int. Conf. Micro Electro Mechanical Systems ('MEMS 02)*, 324-327, Las Vegas, NV, 2002

- [70] H. Xie, L. Erdmann, Q. Jing, and G. K. Fedder, "Simulation and Characterization of a CMOS Z-axis Microactuator with Electrostatic Comb Drives," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '00)*, San Diego, CA, March 27-29, 2000
- [71] J. Wu and L. R. Carley, "Table-based Numerical Macromodeling for MEMS Devices," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '01)*, 68-71, Hilton Head Island, South Carolina, March 19-21, 2001
- [72] M. S-C. Lu and G. K. Fedder, "Parameterized Electrostatic Gap Models for Structured Design of Microelectromechanical Systems," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '99)*, 280-283, San Juan, Puerto Rico, April 19-21, 1999
- [73] L. D. Gabbay, J. E. Mehner, and S. D. Senturia, "Computer-Aided Generation of Nonlinear Reduced-order Dynamic Macromodels-I: Non-stress-stiffened case," *J. Microelectromechanical Systems*, vol. 9, no. 2, pp. 262-269, June 2000
- [74] S. V. Iyer, H. Lakdawala, G. K. Fedder, and T. Mukherjee, "Modeling Methodology for CMOS-MEMS Electrostatic Comb," *Proc. Design, Test, Integration and Packaging of MEMS/MOEMS (DTIP 2002)*, 55-62, Paris, Cannes, May 6-8, 2002
- [75] W. H. Chang, "Analytic IC Metal-line Capacitance Formulas," *IEEE Trans. Microwave Theory and Techniques*, pp. 608-611, September 1979
- [76] S. Iyer, H. Lakdawala, G. K. Fedder and T. Mukherjee, "Macromodeling Temperature-Dependent Curl in CMOS Micromachined Beams," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '01)*, 88-91, Hilton Head Island, South Carolina, March 19-21, 2001
- [77] N. R. Draper and H. Smith, *Applied Regression Analysis*, Second Edition, John Wiley and Sons, New York, 1981
- [78] W. Venables and B. D. Ripley, *Modern Applied Statistics with S-PLUS*, 2nd Edition, Springer Verlag, January 1998
- [79] S. Iyer and T. Mukherjee, "Simulation of Manufacturing Variations in a Z-axis CMOS-MEMS Gyroscope," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '02)*, 186-189, Puerto Rico, April 22-24, 2002

- [80] T. Mukherjee, G. K. Fedder, D. Ramaswamy, and J. White, "Emerging Simulation Approaches for Micromachined Devices," *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, pp. 1572-1588, vol. 19, no. 12, December 2000
- [81] P. Voigt, G. Schrag and G. Wachutka, "Electrofluidic full-system modelling of a flap valve micropump based on Kirchoffian Theory," *Sensors and Actuators A*, 66, pp. 9-14, 1998
- [82] D. Fitzpatrick, I. Miller, *Analog Behavioral Modeling With the Veriloga Language*, Kluwer Academic Publishers, November 1997
- [83] MAST Reference Manual, Release 4.0, Analogy Inc. Beaverton OR, 1995
- [84] S. V. Iyer, Q. Jing, G. K. Fedder, and T. Mukherjee, "Convergence and Speed Issues in Analog HDL Model Formulation for MEMS," *Proc. Fourth Int. Conf. Modeling and Simulation of Microsystems (MSM 2001)*, Hilton Head, SC, 590-593, March 19-21, 2001,
- [85] G. K. Fedder, "Mechanical Natures: A White Paper for the VHDL-AMS Interoperability Working Group," November 24, 1998
- [86] G. K. Fedder, Q. Jing and T. Mukherjee, *Personal Communications*
- [87] Spectre User Guide, Cadence Design Systems, 555 River Oaks Parkway, San Jose, CA, [http:// www.cadence.com](http://www.cadence.com)
- [88] Affirma Veriloga language Reference, Cadence Design Systems, San Jose CA
- [89] L. T. Pillage, R. E. Rohrer, and C. Visweswariah, *Electronic Circuit and System Simulation Methods*, McGraw-Hill Inc., New York 1995
- [90] E. Kreyszig, *Advanced Engineering Mathematics*, 8th edition, John Wiley and Sons, Singapore, 1999
- [91] S. Vemuri, G. K. Fedder, and T. Mukherjee, "Low-Order Squeeze Film Model for Simulation of MEMS Devices," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '00)*, San Diego, CA, March 27-29, 2000
- [92] C. Acar and A. Shkel, "A Design Approach for Robustness Improvement of Rate Gyroscopes," *Proc. Int. Conf. Modeling and Simulation of Microsystems (MSM '01)*, 80-83, Hilton Head, SC, March 19-21, 2001

- [93] W. O. Davis and A. P. Pisano, "Nonlinear Mechanics of Suspension Beams for a Micromachined Gyroscope," *Proc. Int. Conf. Modeling and Simulation of Microsystems 2001 (MSM '01)*, 270-273, Hilton Head Island, SC, March 19-21, 2001
- [94] W. O. Davis, *Mechanical Analysis and Design of Vibratory Micromachined Gyroscopes*, Ph. D. thesis, University of California, Berkeley, Spring 2001
- [95] Q. Jing, T. Mukherjee, and G. K. Fedder, "Large-Deflection Beam Model for Schematic-based Behavioral Simulation in NODAS," *Proc. Int. Conf. Modeling and Simulation of Microsystems 2002 (MSM '02)*, 136-139, San Juan, Puerto Rico, April 21-25, 2002
- [96] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons Inc.
- [97] NeoCircuit Reference Manual, Neolinear Inc., Pittsburgh, Pennsylvania
- [98] Z. Bai, K. S. Pister et al., "New Numerical Techniques and Tools in Sugar for 3D MEMS Simulation," *Proc. Int. Conf. Modeling and Simulation of Microsystems 2001 (MSM 01)*, 31-34, Hilton Head Island, SC, March 19-21, 2001

Appendix A1 Mathematica program to derive Crab-leg stiffness matrix

(* Computes the in-plane spring constants for the crab leg. *)

(* Lt is along the x direction,

Ls is along the y direction

Lt is attached to the plate *)

```
Clear[Ab, At, It, Is, Lt, Ls, Mb2, Ms, Mt, Ub1, Ub2, Ub3, Ly, Lx, tt, EY];
```

```
Clear[dtheta, dx, dy, Fx, Fy, M0];
```

```
Mt = M0 + Fx*Ly - Fy*Lx - Fy t;
```

```
Ms = M0 + Fx*Ly - Fy*Lx - Fy Lt + Fx t;
```

```
Ub1 = \[Integral]\_0\%Lt Mt^2/2 *EY * It \[DifferentialD]t,;
```

```
Ut = \[Integral]\_0\%Ls Ms^2/2 *EY * Is \[DifferentialD]t,;
```

```
Ucomp = Fx^2 Lt/2 EY Ab + Fy^2 Ls/2 EY At;
```

```
U = Ub1 + Ut;
```

```
dtheta = \[PartialD]\_M0 U,;
```

```
dx = \[PartialD]\_Fx U,;
```

```
dy = \[PartialD]\_Fy U;\n
```

(* ***** Computing kxx ***** *)

```
tmp = Solve[{dy == 0, dtheta == 0}, {Fy, M0}];
```

```
tmp1 = dx /. tmp;
```

```
Kx = Simplify[Fx/tmp1]
```

(***** End of kxx *****)

(***** Computing kyy *****)

tmp = Solve[{dx == 0, dtheta == 0}, {Fx, M0}];

tmp1 = dy /. tmp;

Ky = Simplify[Fy/tmp1]

(***** End of kyy *****)

(***** Computing ktt or kthetaz *****)

tmp = Solve[{dx == 0, dy == 0}, {Fx, Fy}];

tmp1 = dtheta /. tmp;

Kt = Simplify[M0/tmp1]

(***** End of ktt or kthetaz *****)

(***** Computing kxy *****)

tmp = Solve[{dx == 0, dtheta == 0}, {Fy, M0}];

tmp1 = dy /. tmp;

Kxy = Simplify[Fx/tmp1]

(***** End of kxy *****)

(***** Computing kyt or kythetaz *****)

tmp = Solve[{dx == 0, dy == 0}, {Fx, M0}];

tmp1 = dtheta /. tmp;

Kyt = Simplify[Fy/tmp1];

(***** End of kyt or kythetaz *****)

(***** Computing kxt or kxthetaz *****)

```

tmp1 = dtheta /. tmp;,
Kxt = Simplify[Fx/tmp1];
(***** End of kxthetaz *****)

(***** Computing kty or kthetazy *****)
tmp = Solve[{dx == 0, dtheta == 0}, {Fx, Fy}];
tmp1 = dy /. tmp;
Kty = Simplify[M0/tmp1]
(***** End of kty or kthetazy *****)

```

Appendix A2 Equations for out-of-plane off-diagonal stiffness constants

For crab-leg springs:

$$k_{z\phi_x} = \frac{6EI_{yt}I_{ys}(GJ_sL_t + EI_{yt}L_s)(2EI_{ys}L_tL_y + GJ_tL_s(L_s + 2L_y))}{D_{cop}} \quad (A2.1)$$

$$k_{z\phi_y} = \frac{-6EI_{yt}I_{ys}(EI_sL_t + GJ_tL_s)(2EI_{yt}L_s(L_t + L_x) + GJ_sL_t(L_t + 2L_x))}{D_{cop}} \quad (A2.2)$$

$$k_{\phi_x\phi_y} = \frac{-3EI_{yt}I_{ys}(2EI_{ys}L_tL_y + GJ_tL_s(2L_y + L_s))(2EI_{yt}L_s(L_t + L_x) + GJ_sL_t(L_t + 2L_x))}{D_{cop}} \quad (A2.3)$$

where,

$$D_{cop} = \left(\begin{aligned} &4E^2I_{yt}I_{ys}L_tL_s(I_{ys}L_t^3 + I_{yt}L_s^3) + G^2J_tJ_sL_tL_s(I_{ys}L_t^3 + I_{yt}L_s^3) + \\ &EG(I_{ys}^2J_sL_t^5 + I_{yt}^2J_tL_s^5 + 4I_{yt}I_{ys}L_t^2L_s^2(J_tL_t + J_sL_s)) \end{aligned} \right) \quad (A2.4)$$

is the common denominator for out-of-plane coupling coefficients for the crab-leg.

For u-springs

$$k_{z\phi_x} = \frac{\left(\begin{aligned} &6EGI_{yb}I_{yt}J_b(GJ_t(L_{b1} + L_{b2}) + EI_{yb}L_t) \\ &(GJ_bL_t(L_t + 2L_y) + 2EI_{yt}(L_{b1}L_y + L_{b2}(L_t + L_y))) \end{aligned} \right)}{D_{uop}} \quad (A2.5)$$

$$k_{z\phi_y} = \frac{\left(\begin{aligned} &-6EGI_bI_tJ_b(EI_{yt}(L_{b1} + L_{b2}) + GJ_bL_t) \\ &(2EI_{yb}L_t(L_{b1} + L_x) + GJ_t(L_{b1}^2 - L_{b2}(L_{b2} - 2L_x) + 2L_{b1}(L_{b2} + L_x))) \end{aligned} \right)}{D_{uop}} \quad (A2.6)$$

$$k_{\phi_x\phi_y} = \frac{\left(\begin{aligned} &-3EGI_{yb}I_{yt}J_b(GJ_bL_t(2L_y + L_t) + 2EI_{yt}(L_{b1}L_y + L_{b2}(L_y + L_t))) \times \\ &(2EI_{yb}L_t(L_{b1} + L_x) + GJ_t(L_{b1}^2 - L_{b2}(L_{b2} - 2L_x) + 2L_{b1}(L_{b2} + L_x))) \end{aligned} \right)}{D_{uop}} \quad (A2.7)$$

where,

$$D_{uop} = \left[\begin{array}{c} 12E^3 I_{yb}^2 I_{yt}^2 L_{b1} L_{b2} L_t^3 + \\ G^3 J_b^2 J_t L_t (I_{xt} (L_{b1}^4 + 4L_{b1}^3 L_{b2} - 6L_{b1}^2 L_{b2}^2 + 4L_{b1} L_{b2}^3 + L_{b2}^4) + I_{yb} (L_{b1} + L_{b2}) L_t^3) + \\ EG^2 J_b (I_{yt}^2 J_t (L_{b1}^5 + 5L_{b1}^4 L_{b2} - 2L_{b1}^3 L_{b2}^2 - 2L_{b1}^2 L_{b2}^3 + 5L_{b1} L_{b2}^4 + L_{b2}^5) + I_{yb}^2 J_b L_t^5) + \\ EG^2 J_b (4I_{xb} I_{xt} (L_{b1} + L_{b2}) L_t^2 (J_b L_{b1}^2 - J_b L_{b1} L_{b2} + J_b L_{b2}^2 + J_t L_{b1} L_t + J_t L_{b2} L_t)) + \\ 4E^2 G I_{yb} I_{yt} (L_{b1} + L_{b2}) L_t (I_{yb} J_b L_t^3 + I_{yt} (J_b L_{b1}^3 + J_b L_{b2}^3 + 3J_t L_{b1} L_{b2} L_t)) \end{array} \right] \quad (A2.8)$$

For serpentine springs for even n:

$$k_{z\phi_x} = \frac{6EG I_{ya} I_{yb} J_a J_b (2L_y + nL_a)}{D_{seop}} \quad (A2.9)$$

$$k_{z\phi_y} = \frac{-6EG I_{ya} I_{yb} J_a J_b (L_b + 2L_x)}{D_{seop}} \quad (A2.10)$$

$$k_{\phi_x \phi_y} = \frac{-3EG I_{ya} I_{yb} J_a J_b (L_b + 2L_x) (2L_y + nL_a)}{D_{seop}} \quad (A2.11)$$

where,

$$D_{seop} = (L_b^3 G I_{ya} J_a J_b (n-1) + 3n^2 L_a L_b^2 E I_{ya} I_{yb} J_b + n^3 L_a^3 G I_{yb} J_a J_b + n L_a^2 L_b I_{ya} I_{yb} J_a (n^2 - 3n + 2)) \quad (A2.12)$$

Serpentine springs with odd n:

$$k_{z\phi_x} = \frac{6EG I_{ya} I_{yb} J_a J_b (2L_y + nL_a) ((n-1) G J_a L_b + n E I_{yb} L_a)}{D_{soop}} \quad (A2.13)$$

$$k_{z\phi_y} = \frac{-6EG I_{ya} I_{yb} J_a J_b (G J_a (L_b + 2L_x) (n-1) L_b + E I_{yb} L_a (L_b (n-1) + 2n L_x))}{D_{soop}} \quad (A2.14)$$

$$k_{\phi_x \phi_y} = \frac{-3EG I_{ya} I_{yb} J_a J_b (2L_y + nL_a) (G J_a (L_b + 2L_x) (n-1) L_b + E I_{yb} L_a (L_b (n-1) + 2n L_x))}{D_{soop}} \quad (A2.15)$$

$$D_{soop} = \left[\begin{array}{c} (n-1)^2 G^2 I_{ya} J_a^2 J_b L_b^4 + 4n(n-1) E G I_{ya} I_{yb} J_a J_b L_a L_b^3 + n^4 E G I_{yb}^2 J_a J_b L_a^4 + \\ n^2 (n-1) I_{yb} J_a L_a^3 L_b (E^2 I_{ya} I_{yb} (n-2) + G^2 J_a J_b n) + \\ (n-1) E I_{ya} I_{yb} L_a^2 L_b^2 (3E I_{yb} J_b (n+1) + G J_a^2 n (n^2 - 3n + 2)) \end{array} \right] \quad (A2.16)$$

FEA verification results for out-of-spring stiffness constants are shown in Table 2.1

Table A2.1 Comparison of FEA and analytical stiffness (out-of-plane) values for the crab-leg spring

w	L_t	L_s	$k_{z\phi_x} (\times 10^{-6} \text{N})$			$k_{z\phi_y} (\times 10^{-6} \text{N})$			$k_{\phi_x\phi_y} (\times 10^{-12} \text{N-m})$		
			μm	μm	μm	A	S	Error (%)	A	S	Error (%)
2	40	40	320.5	332.5	-3.61	564.1	576.7	-2.18	2.66e4	2.74e4	-3.10
2	400	40	1.17	1.203	-2.74	8.061	8.226	-2.01	324.7	331.9	-2.17
2	40	400	6.166	6.394	-3.57	2.605	2.675	-2.62	553.2	573.4	-3.52
2	400	400	0.8598	0.9072	-5.22	2.704	2.742	-1.39	319.7	334.1	-4.31
5	40	40	877.8	929.1	-5.52	1457	1514	-3.76	7.12e4	7.48e4	-4.81
5	400	40	3.208	3.306	-2.96	21.61	22	-1.77	860.1	880.7	-2.34
5	40	400	17.88	18.4	-2.83	7.268	7.412	-1.94	1612	1652	-2.42
5	400	400	3.071	3.156	-2.69	7.289	7.369	-1.09	1059	1082	-2.13

for the crab-leg spring, Table 2.2 for the u-spring and Table 2.3 for the serpentine spring

Table A2.2 Comparison of FEA and analytical stiffness (out-of-plane) values for the U spring

w	L_{b1}	L_{b2}	L_t	$k_{z\phi_x} (\times 10^{-6} \text{N})$			$k_{z\phi_y} (\times 10^{-6} \text{N})$			$k_{\phi_x\phi_y} (\times 10^{-12} \text{N-m})$		
				μm	μm	μm	μm	A	S	Error (%)	A	S
2	40	40	10	480	502	-4.29	-727	-755	-3.71	-3.56e4	-3.71e4	-3.91
2	40	400	10	1.65	1.68	-1.85	2.62	2.69	-2.64	140.7	144.7	-2.76
2	400	40	10	1.353	1.37	-1.38	-8.40	-8.49	-1.08	-371.3	-375.8	-1.20
2	400	400	10	0.936	0.95	-1.48	-4.87	-4.93	-1.26	-238.8	-241.8	-1.24
2	40	40	50	121.4	130	-6.33	-142	-150	-5.66	-9784	-1.04e4	-5.92
2	40	400	50	2.155	2.21	-2.62	1.57	1.67	-5.69	139.6	148.1	-5.74
2	400	40	50	1.20	1.23	-2.68	-7.18	-7.30	-1.71	-354.1	-361.2	-1.97
2	400	400	50	1.05	1.08	-2.60	-4.08	-4.16	-2.09	-281.2	-287.3	-2.12
5	40	40	10	1207	1278	-5.56	-1825	-1913	-4.60	-8.94e4	-9.44e4	-5.26
5	40	400	10	4.24	4.33	-2.19	6.67	6.86	-2.84	365.6	378.6	-3.43
5	400	40	10	3.33	3.38	-1.33	-21.1	-21.3	-0.75	-912.1	-923	-1.18
5	400	400	10	2.36	2.4	-1.71	-12.3	-12.4	-1.29	-600.2	-609.4	-1.51
5	40	40	50	394	418	-5.71	-451	-471	-4.35	-3.11e4	-3.29e4	-5.33

Table A2.2 Comparison of FEA and analytical stiffness (out-of-plane) values for the U spring

w	L_{b1}	L_{b2}	L_t	$k_{z\phi_x} (\times 10^{-6}\text{N})$			$k_{z\phi_y} (\times 10^{-6}\text{N})$			$k_{\phi_x\phi_y} (\times 10^{-12}\text{N-m})$		
μm	μm	μm	μm	A	S	Error (%)	A	S	Error (%)	A	S	Error (%)
5	40	400	50	5.87	6.02	-2.36	4.90	5.078	-3.60	433.1	451.5	-4.08
5	400	40	50	3.29	3.35	-1.79	-19.0	-19.2	-0.94	-941.3	-956	-1.54
5	400	400	50	2.87	2.93	-1.98	-10.9	-11.1	-1.53	-752.8	-765.8	-1.70

Table A2.3 Comparison of FEA and analytical stiffness (out-of-plane) values for the serpentine spring

w	L_a	L_b	$k_{z\phi_x} (\times 10^{-6}\text{N})$			$k_{z\phi_y} (\times 10^{-6}\text{N})$			$k_{\phi_x\phi_y} (\times 10^{-12}\text{N-m})$		
μm	μm	μm	A	S	Error (%)	A	S	Error (%)	A	S	Error (%)
2	10	10	1513	1662	-8.97	-1038	-1140	-8.95	-7.26e4	-7.98e4	-8.97
2	10	100	35.08	37.66	-6.85	-46.6	-50.03	-6.86	-3262	-3502	-6.85
2	100	10	9.726	9.856	-1.32	-1.867	-1.892	-1.32	-466.8	-473.2	-1.35
2	100	100	5.405	5.628	-3.96	-2.011	-2.094	-3.96	-502.6	-523.5	-3.99
4	10	10	3658	4173	-12.3	-2561	-2802	-8.60	-1.79e5	-1.96e5	-8.57
4	10	100	83.52	91.16	-8.38	-112.1	-119.8	-6.43	-7850	-8383	-6.36
4	100	10	19.88	20.19	-1.54	-3.897	-3.794	2.71	-974.3	-946.9	2.89
4	100	100	13.06	13.39	-2.46	-4.912	-4.925	-0.26	-1228	-1231	-0.24

respectively.

Appendix A3 Mesh Refinement Steps

In this appendix the mesh refinement steps carried out inside AutoBEM are listed for three cases: simple comb capacitance, simple comb force runs (for numerical differentiation), differential comb capacitance

Mesh refinement for simple comb capacitance.

```
(* Before starting with the refinement procedure we have to form the
   ROTORTOTAL and STATORTOTAL regions *)
selection{type{panel},clear};

(* Make ROTORTOTAL *)
selection{type{panel}, add{region{"ROTOR"}}};
selection{type{panel}, add{region{"ROTOROXIDE"}}};
region{name{ "ROTORTOTAL" },panels{,,};
selection{type{panel},operation{add,region{"ROTORTOTAL"}}};
selection{type{panel},clear};
set{ property{ region{ visual{ name{ "ROTORTOTAL"}, value{
0.701960784314,0.0, 0.0, 1.00, 0.10, 10, 0.62, 0.59} } } } };

(* Make STATORTOTAL *)
selection{type{panel}, add{region{"STATOR"}}};
selection{type{panel}, add{region{"STATOROXIDE"}}};
region{ name{ "STATORTOTAL" },panels{,,};
selection{type{panel},operation{add,region{"STATORTOTAL"}}};
selection{type{panel},clear};
```

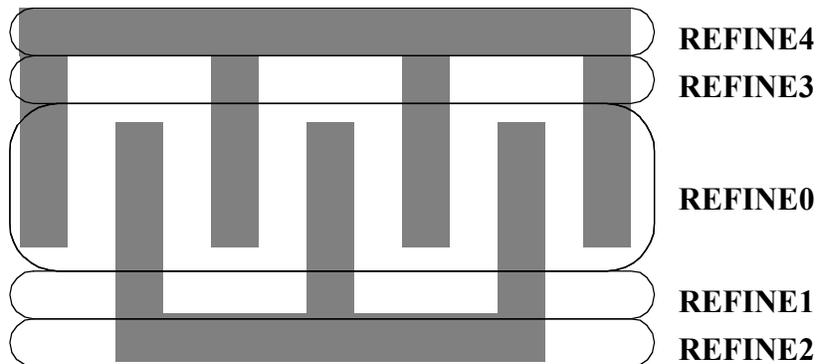


FIGURE A3.1. Regions defined in the boundary element model for mesh refinement. REFINE0 is the most finely meshed region, followed by REFINE1 and REFINE3 and finally REFINE2 and REFINE4 are the most coarsely meshed regions.

```
set{ property{ region{ visual{ name{ "STATORTOTAL"}, value{ 0.0,0.0,
0.701960784314, 1.00, 0.10, 10, 0.62, 0.59} } } } };
```

(* The following is the refinement algorithm:

Steps are as follows:

1. REFINE0:

smoothness>0.5 edge refinement with AR 1000 min diagonal of 0.2
*)

```
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.2},    includeRegion{"REFINE0"},
    },
    type{    edge{aspectRatio{1000}}    },
    absolute{ 0.5 },
  }
};
```

(*

2. REFINE3:

smoothness>0.5 edge refinement for AR > 1000 min diagonal of 0.2
*)

```
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.5},    includeRegion{"REFINE3"},
    },
    type{    edge{aspectRatio{1000}}    },
    absolute{ 0.5 },
  }
};
```

(*

3. REFINE1:

smoothness>0.5 edge refinement for AR > 1000 min diagonal of 0.4
*)

```
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.4},    includeRegion{"REFINE1"}
    },
    type{    edge{aspectRatio{1000}}    },
    absolute{ 0.5 },
  }
};
```

```

(*)
4. REFINE5 and REFINE6
   smoothness > 0.5 edge refinement for AR > 1000 min diagonal of 0.1
*)
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.1},    includeRegion{"REFINE5","REFINE6",}
    },
    type{    edge{aspectRatio{1000}}    },
    absolute{ 0.5 },
  }
};

```

Mesh refinement for simple comb force runs.

```

(* Before starting with the refinement procedure we have to form the
   ROTORTOTAL and STATORTOTAL regions *)
selection{type{panel},clear};

```

```

(* Make ROTORTOTAL *)
selection{type{panel}, add{region{"ROTOR"}}};
selection{type{panel}, add{region{"ROTOROXIDE"}}};
region{name{ "ROTORTOTAL" },panels{ },};
selection{type{panel},operation{add,region{"ROTORTOTAL"}}};
selection{type{panel},clear};
set{ property{ region{ visual{ name{ "ROTORTOTAL"}, value{
0.701960784314,0.0, 0.0, 1.00, 0.10, 10, 0.62, 0.59} } } } };

```

```

(* Make STATORTOTAL *)
selection{type{panel}, add{region{"STATOR"}}};
selection{type{panel}, add{region{"STATOROXIDE"}}};
region{ name{ "STATORTOTAL" },panels{ },};
selection{type{panel},operation{add,region{"STATORTOTAL"}}};
selection{type{panel},clear};
set{ property{ region{ visual{ name{ "STATORTOTAL"}, value{ 0.0,0.0,
0.701960784314, 1.00, 0.10, 10, 0.62, 0.59} } } } };

```

(* The following is the refinement algorithm:
Steps are as follows:

```

1. REFINE0:
   smoothness>0.5 edge refinement with AR 1000 min diagonal of 0.05
*)

```

```

refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.05},    includeRegion{"REFINE0",}
    }
  }
};

```

```

    },
    type{      edge{aspectRatio{1000}}      },
    absolute{ 0.5 },
  }
};

(*
2. REFINE0:
  smoothness>1.5 edge refinement with AR 1000 min diagonal of 0.02
*)

refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.02},    includeRegion{"REFINE0",}
    },
    type{      edge{aspectRatio{1000}}      },
    absolute{ 1.5 },
  }
};

(*3. REFINE3:
  smoothness>0.5 edge refinement for AR > 1000 min diagonal of 0.2
*)

refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.5},    includeRegion{"REFINE3",}
    },
    type{      edge{aspectRatio{1000}}      },
    absolute{ 0.5 },
  }
};

(*
4. REFINE1:
  smoothness>0.5 edge refinement for AR > 1000 min diagonal of 0.4
*)

refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.4},    includeRegion{"REFINE1"}
    },
    type{      edge{aspectRatio{1000}}      },
    absolute{ 0.5 },
  }
}

```

```

};

(*
5. REFINE5 and REFINE6
smoothness > 0.5 edge refinement for AR > 1000 min diagonal of 0.1
*)
refine{
simulationSpecification{"computation1"},
elementSmoothness{
option{
minimumElementDiagonal{0.1}, includeRegion{"REFINE5","REFINE6",}
},
type{ edge{aspectRatio{1000}} },
absolute{ 0.5 },
}
};

```

```

(*
6. All h-refinement
*)
refine {
simulationSpecification{computation1},
all{
type{ h },
}
};

```

Mesh refinement for differential comb capacitance.

(* Before starting with the refinement procedure we have to form the ROTORTOTAL and STATORTOTAL regions *)

```

selection{type{panel},clear};

(* Make ROTORTOTAL *)
selection{type{panel}, add{region{"ROTOR"}}};
selection{type{panel}, add{region{"ROTOROXIDE"}}};
region{name{ "ROTORTOTAL" },panels{ },};
selection{type{panel},operation{add,region{"ROTORTOTAL"}}};
selection{type{panel},clear};
set{ property{ region{ visual{ name{ "ROTORTOTAL"}, value{
0.701960784314,0.0, 0.0, 1.00, 0.10, 10, 0.62, 0.59} } } } };

```

```

(* Make STATORTOTAL *)
selection{type{panel}, add{region{"STATOR"}}};
selection{type{panel}, add{region{"STATOROXIDE"}}};
region{ name{ "STATORTOTAL" },panels{ },};
selection{type{panel},operation{add,region{"STATORTOTAL"}}};
selection{type{panel},clear};
set{ property{ region{ visual{ name{ "STATORTOTAL"}, value{ 0.0,0.0,
0.701960784314, 1.00, 0.10, 10, 0.62, 0.59} } } } };

```

(* The following is the refinement algorithm:

Steps are as follows:

1. REFINE0:

smoothness>0.5 edge refinement with AR 1000 min diagonal of 0.2
*)

```
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.2},    includeRegion{"REFINE0"},
    },
    type{    edge{aspectRatio{1000}}    },
      absolute{ 0.5 },
    }
  };
```

(*

2. REFINE3:

smoothness>0.5 edge refinement for AR > 1000 min diagonal of 0.2
*)

```
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.5},    includeRegion{"REFINE3"},
    },
    type{    edge{aspectRatio{1000}}    },
      absolute{ 0.5 },
    }
  };
```

(*

3. REFINE1:

smoothness>0.5 edge refinement for AR > 1000 min diagonal of 0.4
*)

```
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.4},    includeRegion{"REFINE1"}
    },
    type{    edge{aspectRatio{1000}}    },
      absolute{ 0.5 },
    }
  };
```

(*

4. REFINE5 and REFINE6

```

    smoothness > 0.5 edge refinement for AR > 1000 min diagonal of 0.1
*)
refine{
  simulationSpecification{"computation1"},
  elementSmoothness{
    option{
      minimumElementDiagonal{0.1},    includeRegion{"REFINE5","REFINE6",}
    },
    type{    edge{aspectRatio{1000}}    },
    absolute{ 0.5 },
  }
};

```

Appendix A4 Comb Model Listing

In this appendix the VerilogA[®] implementations of the curve-fit based capacitance models as described in Chapter 5 are listed. Models for the simple comb are presented first followed by models for the differential comb.

The simple comb model with curling effects consists of 4 files.

1. `comb_curl_cyy_1.va`: Main model file which “includes” the 3 other files
2. `efringe.va`: File containing functions for computing the fringe capacitance. The functions in this file are derived analytically
3. `cap.va`: File containing numerically curve-fit coefficients and equation for capacitance
4. `force.va`: File containing equation for force

Comb_curl_cyy_1.va

```
// VerilogA for "comb_1" "veriloga"
// Modified on January 1, 2002
// 1. Changed the calculation of capacitance from a function call
//    to simple inclusion of the file
// 2. Added another include file containing derivatives for computing
force

`include "../..//constants.h"
`include "../..//discipline.h"
`include "../..//process.h"
`include "../..//design.h"

// "a" side has more fingers
// "b" side has less fingers
// nominally a is to the left of b

module comb_curl_cyy_1(OMG, a, phia, phib, va, vb, xa, xb);
    // Pin definitions
    // Inertial Pins
    inout [0:2] OMG;
    rotational [0:2] OMG;
    inout [0:2] a;
    kinematic [0:2] a;

    // Position and voltage pins
    inout [0:2] phia;
    rotational [0:2] phia;
    inout [0:2] phib;
    rotational [0:2] phib;
    inout [0:3] va;
```

```

electrical [0:3] va;
inout [0:3] vb;
electrical [0:3] vb;
inout [0:2] xa;
kinematic [0:2] xa;
inout [0:2] xb;
kinematic [0:2] xb;

// Parameter Definitions;
parameter real finger_width = `default_comb_finger_width;
parameter real finger_length = `default_comb_finger_length;
parameter real overlap = `default_comb_overlap;
parameter real gap = `default_comb_gap;
parameter real fingers = `default_comb_fingers;
parameter real angle = `default_comb_angle;
parameter real poly_cut_in = `default_poly_cut_in;
parameter real Xc = `default_comb_Xc;
parameter real Yc = `default_comb_Yc ;
parameter real wing_length_a = `default_comb_wing_length_a;
parameter real wing_length_b = `default_comb_wing_length_b;
parameter real truss_width_a = `default_comb_truss_width_a;
parameter real truss_width_b = `default_comb_truss_width_b;

parameter real alpha = 0;
parameter real beta = 0;
parameter real gamma = 0;

parameter integer flip_about_y = 0 from [0:1]; // If flip is zero
that means that there is no flipping
parameter real rho_a = 0; // Radius of curvature of a side
parameter real rho_b = 0; // Radius of curvature of b side
parameter real za = 0;
parameter real zb = 0;

// These angles are the static angular displacements at the
// base of the a side and the b side respectively
parameter real angle_a = 0;
parameter real angle_b = 0;
//gap(air)
parameter real air_gap = `default_air_gap;

//Viscosity of air at Atmospheric pressure and at T=288K
parameter real visc_air = `default_visc_air;

parameter real ntv_ox_t = `default_ntv_ox_t;

parameter real delta = `default_delta;

parameter real E = `default_E_cmos;
parameter real den_metal=`default_den_metal;
parameter real den_poly=`default_den_poly;
parameter real den_oxide=`default_den_oxide;

```

```

parameter real t_m3 = `default_t_m3;
parameter real t_m2 = `default_t_m2;
parameter real t_m1 = `default_t_m1;
parameter real t_poly = `default_t_poly;
parameter real t_m3_m2 = `default_t_m3_m2;
parameter real t_m2_m1_overpoly = `default_t_m2_m1_overpoly;
parameter real t_m1_poly = `default_t_m1_poly;
parameter real t_poly_sub = `default_t_poly_sub;

parameter real t_cmos =
t_m3+t_m3_m2+t_m2+t_m2_m1_overpoly+t_m1+t_m1_poly+t_poly+t_poly_sub;
parameter real den_cmos =
(den_metal*(t_m3+t_m2+t_m1)+den_poly*t_poly+den_oxide*(t_m3_m2+t_m2_m1_o
verpoly+t_m1_poly+t_poly_sub))/t_cmos;
    // Call sub-modules
    mass_comb_3D # (.angle(angle),
.finger_width(finger_width),
    .finger_length(finger_length),
    .gap(gap),
    .fingers(fingers),
    .truss_width(truss_width_a),
    .wing_length(wing_length_a),
.thickness(t_cmos),
.density(den_cmos),
.flip(1),
.number(1)) mass_a(xa, phia, OMG, a);

    mass_comb_3D # (.angle(angle),
.finger_width(finger_width),
    .finger_length(finger_length),
    .gap(gap),
    .fingers(fingers),
    .truss_width(truss_width_b),
    .wing_length(wing_length_b),
.thickness(t_cmos),
.density(den_cmos),
.flip(-1),
.number(0)) mass_b(xb, phib, OMG, a);

    viscous_damping_comb #(.angle(angle),
.finger_width(finger_width),
.finger_length(finger_length),
.gap(gap),
.fingers(fingers),
.truss_width(truss_width_a),
.wing_length(wing_length_a),
.number(1),
.top_gap(delta),
.bottom_gap(delta)) damp_a(xa[0], xa[1], phia[2]);

    viscous_damping_comb #(.angle(angle),
.finger_width(finger_width),
.finger_length(finger_length),
.gap(gap),

```

```

.fingers(fingers),
.truss_width(truss_width_b),
.wing_length(wing_length_b),
.number(0),
.top_gap(delta),
.bottom_gap(delta)) damp_b(xb[0], xb[1], phib[2]);

coutte_damping #(.angle(angle),
.length(overlap),
.width(t_cmos),
.gap(gap),
.multiplier1(fingers),
.multiplier2(2)) side_wall_damp(xa[0], xa[1], phia[2], xb[0], xb[1],
phib[2]);

sq_damp_model_general #(.length(overlap),
.width(t_cmos),
.gap(gap),
.Pamb(1.0e5),
.multiplier1(fingers),
.multiplier2(2),
.angle(angle)) damp_sq_finger(xa[0], xa[1], xb[0], xb[1]);

`include "/afs/ece/usr/sita/.vol1/models/comb/curvefit/efringe.va"
// definition of user parameters (with default values)
// Euler angles (alpha, beta and gamma, in degrees) are used
// to specify beam orientation and to form coordination transformation
matrix
// Order of rotation:
// step1: rotate by gamma degree about z-axis
// step2: rotate by beta degree about y-axis
// step3: rotate by alpha degree about x-axis
// Following variables will be used in going from the chip frame to
the local frame of the model

real      cos_alpha, cos_beta, cos_gamma;
real      sin_alpha, sin_beta, sin_gamma;
real      l1, m1, n1 ;
real      l2, m2, n2 ;
real      l3, m3, n3 ;
real      inv_l1, inv_m1, inv_n1 ;
real      inv_l2, inv_m2, inv_n2 ;
real      inv_l3, inv_m3, inv_n3 ;

real      gamma_offset; // is 90 degrees because the simulations
were done
// with the combs rotated by 90 degrees
real      alpha_local_model; // This angle is required to convert
the displacements in the local
// frame to that in the model frame
// Variables for holding position in chip and local frames

```

```

real      l_phixa, l_phixb, l_phiya, l_phiza, l_phiyb, l_phizb;
real      l_phiz, l_phiy;
real      chip_phixa, chip_phiya, chip_phiza;
real      chip_phixb, chip_phiyb, chip_phizb;

real      l_xa, l_xb, l_ya, l_yb, l_za, l_zb;
real      l_dx, l_dy, l_dz;
real      dx, dy, dz;
real      chip_x, chip_y, chip_z;
real      chip_xa, chip_ya, chip_xb, chip_yb, chip_za, chip_zb;

// Variables for holding force and torque in local frame
real      l_Fxa, l_Fxb, l_Fya, l_Fyb, l_Fza, l_Fzb;
real      l_Tqxa, l_Tqxb, l_Tqya, l_Tqyb, l_Tqza, l_Tqzb;

// Variables for holding force and torque in chip frame
real      Fchipxa, Fchipxb, Fchipya, Fchipyb, Fchipza, Fchipzb;
real      Tqchipxa, Tqchipxb, Tqchipya, Tqchipyb, Tqchipza,
Tqchipzb;

real      w[0:9], thickness, Earray[0:9], tce[0:9]; // width, thick-
ness
integer   m3_id, m2_id, m1_id, poly_id, ox3_id, ox2_id, ox1_id,
ox0_id, ox_poly_left_id, ox_poly_right_id; // indices for array

real      sum_E_w_t_z, sum_E_w_t, sum_E_alpha_w_t_z, sum_E_I; // sum
variables for computing neutral axis

real      zc, z_top[0:9], new_z[0:9]; // z coordinates: neutral
axis, from the top, top - neutral axis

integer   i; // loop variable

real      cyy_per_degree_T, deltaT, cyy; // Temperature and tip
deflections
real      l_total_comb, l_rotor, l_base, l_stator, l_plate_to_rotor;
real      z_matched_rotor, z_matched_stator, phix_matched_rotor,
phix_matched_stator; // Curl matched coordinates

real      dz1; // varies from the curl-matched position
real      dphix, dphix1, dphiy, dphiz; // dphix is the total angle
of a in the model frame
// dphix1 is the mismatch angle i.e. dphix - phix_matched_rotor

real      cap, dCx, dCy, dCz, dCphix, dCphiy, dCphiz, vltg;
real      Fx, Fy, Fz;
real      moment_arm_a, moment_arm_b;
integer   flip;

// Variables for calculation of capacitance and force
real      fwidth, g, olp, dtx, dty, dtz, dtx1, cap_curl, eps0;
real      c0, c1, c2, c3, c4, c5, c6, c7, c8, c9;
real      c10, c11, c12, c13, c14, c15, c16, c17, c18, c19;

```

```

real      c20, c21, c22, c23, c24, c25, c26, c27, c28, c29;
real      c30, c31, c32, c33, c34, c35, c36, c37, c38, c39;
real      c40, c41, c42, c43, c44, c45, c46, c47, c48, c49;
real      c50, c51, c52, c53, c54, c55, c56, c57, c58, c59;
real      c60, c61, c62, c63, c64, c65, c66, c67, c68, c69;
real      c70, c71, c72, c73, c74, c75, c76, c77, c78, c79;
real      c80, c81, c82, c83, c84, c85, c86, c87, c88, c89;
real      c90, c91, c92, c93, c94, c95, c96, c97, c98, c99;
real      c100, c101, c102, c103, c104, c105, c106, c107, c108, c109;
real      c110, c111, c112, c113, c114, c115, c116, c117, c118, c119;
real      ppcapttotal, efcapttotal, defcapttotal, dppcapttotalx, dppcap-
totaly;

```

```

// internal states
electrical vlt;

```

```

analog begin
    @(initial_step) begin
eps0 = 8.85e-12;
// flip_about_y = 1 => flip = -1
// flip_about_y = 0 => flip = 1
flip = 1-2*flip_about_y;

cyy = (1.0/2.0)*(1/(2.0*rho_a) + 1/(2.0*rho_b));
// Can also calculate the tip deflection and the angle to be applied at
the end of the rotor
// so that in the nominal case the two combs are perfectly matched
// First calculate the position of the matched combs with rotor at a
fixed distance from the point of
// zero z deflection and angle

l_plate_to_rotor = 100.0e-6; // This is a value which was used in the
Coyote simulation runs
l_total_comb = 2*finger_length - overlap + truss_width_a +
truss_width_b;
l_rotor = l_plate_to_rotor;
z_matched_rotor = (l_rotor*l_rotor)*cyy*1e6; // in micrometers
phix_matched_rotor = (2*l_rotor*cyy)*180/'M_PI; // in degrees

l_stator = l_plate_to_rotor + l_total_comb;
z_matched_stator = (l_stator*l_stator)*cyy*1e6; // in micrometers
phix_matched_stator = (2*l_stator*cyy)*180/'M_PI; // in degrees

// Three angles need to be computed:
// 1. The angle of the stator - phix_matched_stator
//    This will tell us by what angle the entire comb reference frame
should be rotated
//    so that we are in the reference frame of the simulation data. Note
that this includes
//    rotation of displacements, computation of forces and finally,
reverse rotation of forces
// 2. rotor angle (after transformation by 1.) - phix_matched_rotor

```

```

//      This angle will give us the value corresponding to dtx1 in the
simulations
// 3. rotor angle itself (after transformation by 1.)
//      This angle will give us the value corresponding to dtx in the sim-
ulations
// // translational displacement in chip frame

// Compute the angle which has to be added to the value of beta to
// transform the displacements in the chip frame to the local frame

// For comb_1111_1 the "a" side will be considered equivalent to the
rotor
// and the "b" side will be considered equivalent to the stator

// angle between local frame and frame in which the model is built
alpha_local_model = (angle_a - phix_matched_stator); // in degrees
// rotation of simulation data
gamma_offset = -90; // i.e, gamma = 90 implies no rotation and
// gamma = 0 implies -90 rotation

// coordination transformation matrix based on euler angles
cos_alpha = cos((alpha + alpha_local_model)/180*'M_PI);
cos_beta = cos(beta /180*'M_PI);
cos_gamma = cos((gamma + gamma_offset)/180*'M_PI);
sin_alpha = sin((alpha + alpha_local_model)/180*'M_PI );
sin_beta = sin(beta /180*'M_PI);
sin_gamma = sin((gamma + gamma_offset)/180*'M_PI);

// transformation matrix from chip frame to local frame
l1 = cos_beta*cos_gamma;
m1 = cos_beta*sin_gamma;
n1 = -sin_beta;
l2 = sin_alpha*sin_beta*cos_gamma-cos_alpha*sin_gamma;
m2 = sin_alpha*sin_beta*sin_gamma+cos_alpha*cos_gamma;
n2 = sin_alpha*cos_beta;
l3 = cos_alpha*sin_beta*cos_gamma+sin_alpha*sin_gamma;
m3 = cos_alpha*sin_beta*sin_gamma-sin_alpha*cos_gamma;
n3 = cos_alpha*cos_beta;

// transformation matrix from local frame to chip frame
inv_l1 = cos_beta*cos_gamma;
inv_l2 = cos_beta*sin_gamma;
inv_l3 = -sin_beta;
inv_m1 = sin_alpha*sin_beta*cos_gamma-cos_alpha*sin_gamma;
inv_m2 = sin_alpha*sin_beta*sin_gamma+cos_alpha*cos_gamma;
inv_m3 = sin_alpha*cos_beta;
inv_n1 = cos_alpha*sin_beta*cos_gamma+sin_alpha*sin_gamma;
inv_n2 = cos_alpha*sin_beta*sin_gamma-sin_alpha*cos_gamma;
inv_n3 = cos_alpha*cos_beta;

cyy = cyy*1e-6;
end
chip_x = Pos(xa[0],xb[0]);
chip_y = Pos(xa[1],xb[1]);

```

```

chip_z = Pos(xa[2],xb[2]);
chip_xa = Pos(xa[0]);
chip_ya = Pos(xa[1]);
chip_za = Pos(xa[2]);
chip_xb = Pos(xb[0]);
chip_yb = Pos(xb[1]);
chip_zb = Pos(xb[2]);

// rotational displacement in chip frame all in radians
chip_phixa = 1e6*Theta(phia[0]);
chip_phiya = 1e6*Theta(phia[1]);
chip_phiza = 1e6*Theta(phia[2]);
chip_phixb = 1e6*Theta(phib[0]);
chip_phiyb = 1e6*Theta(phib[1]);
chip_phizb = 1e6*Theta(phib[2]);

// transform from chip frame into local frame
l_xa = l1*chip_xa + m1*chip_ya + n1*chip_za;
l_ya = l2*chip_xa + m2*chip_ya + n2*chip_za;
l_za = l3*chip_xa + m3*chip_ya + n3*chip_za;
l_xb = l1*chip_xb + m1*chip_yb + n1*chip_zb;
l_yb = l2*chip_xb + m2*chip_yb + n2*chip_zb;
l_zb = l3*chip_xb + m3*chip_yb + n3*chip_zb;

l_dx = l1*chip_x + m1*chip_y + n1*chip_z;
l_dy = l2*chip_x + m2*chip_y + n2*chip_z;
l_dz = l3*chip_x + m3*chip_y + n3*chip_z;

l_phixa = l1*chip_phixa + m1*chip_phiya + n1*chip_phiza;
l_phiya = l2*chip_phixa + m2*chip_phiya + n2*chip_phiza;
l_phiza = l3*chip_phixa + m3*chip_phiya + n3*chip_phiza;

l_phixb = l1*chip_phixb + m1*chip_phiyb + n1*chip_phizb;
l_phiyb = l2*chip_phixb + m2*chip_phiyb + n2*chip_phizb;
l_phizb = l3*chip_phixb + m3*chip_phiyb + n3*chip_phizb;

// phix and z displacements total and from curl-matched positions
// za and zb are the changes from the curl-matched positions of the
a comb and the b comb respectively
// Note that dz1 and dphix1 to be passed to comb_curl are for the
base of the truss
// beam which is assumed to be 10 um in the Coyote simulations, but
here they are at
// the position of the "truss_width_a and truss_width_b" respec-
tively
// This is not being taken into account right now

dz1 = zb - za - l_dz*1e6 - (z_matched_rotor - z_matched_stator); /
/ in micrometers
dz = dz1 + z_matched_rotor; // Total relative z displacement in
micrometers

```

```

    // angle_a and angle_b are the changes from the curl-matched angle
of the
    // bases of the a comb and the b comb respectively
    // They are multiplied by 1e-6*'M_PI/180 to convert them to Megara-
dians
    // so that they can be added to l_phixa - l_phixb

    dphix1 = -(angle_a - angle_b) - (phix_matched_rotor -
phix_matched_stator) * 'M_PI/180 - (l_phixa - l_phixb); // in radians
    dphix = phix_matched_rotor * 'M_PI/180 + dphix1; // in radians
    dphiy = l_phiya - l_phiyb;
    dphiz = l_phiza - l_phizb;

    moment_arm_a = truss_width_a + finger_length - (overlap + l_dy)/
2.0;
    moment_arm_b = truss_width_b + finger_length - (overlap + l_dy)/
2.0;
    // Note that 8.85e-12 is used here instead of 'P_EPS0 because
8.85e-12 is used in the perl script to
    // scale down the coefficients
    // 1e-6 is required because cap_curl assumes all dimensions are
meters and therefore returns a capacitance
    // value for all dimensions equal to meters
    // In other words it returns uF and we have to multiply that by 1e-
6 to get F

    // cap = 8.85e-12*1e-6*((fingers)/(3.0))*cap_curl(finger_width*1e6,
gap*1e6, overlap*1e6, cyy*1e-6,
    // l_dx*1e6, l_dy*1e6, dz*1e6,
    // -dphix*1e6*180/'M_PI, dphiy*1e6*180/'M_PI, dphiz*1e6*180/'M_PI,
    // dz1*1e6, -dphix1*1e6*180/'M_PI);
    fwidth = finger_width*1e6; // Needs to be in um
    g = gap*1e6;
    olp = overlap*1e6;
    dx = (l_dx - dphiz*moment_arm_a)*1e6;
    dy = l_dy*1e6;
    dz = dz;
    dtx = -dphix*180/'M_PI;
    dty = dphiy*180/'M_PI;
    dtz = dphiz*180/'M_PI;
    dz1 = dz1;
    dtx1 = -dphix1*180/'M_PI;
#include "cap.va"
    cap = eps0*1e-6*(fingers/3.0)*cap_curl;

    // Note the - sign before dphix and dphix1 in the above function
call, have to keep in mind that
    // the actual matching of rotation in this frame and the model
frame will be more involved
    // and has to be worked out later

```

```

// Current computation
V(vlt) <+ V(va[0], vb[0]);
I(va[0],vb[0]) <+ ddt(cap*V(vlt));

vltg = V(vlt)*V(vlt);

// Force and torque computation in local frame
Fx = 0;
Fy = 0;
Fz = 0;
dCphix = 0;
dCphiy = 0;
dCphiz = 0;

`include "force.va"

Fx = 0.5*eps0*(fingers/3.0)*Fx;
Fy = 0.5*eps0*(fingers/3.0)*Fy;
Fz = 0.5*eps0*(fingers/3.0)*Fz;
//      $display("Fx=%g Fy=%g Fz=%g\n", Fx, Fy, Fz);
//      $display("w=%g g=%g o=%g cyy=%g dx=%g dy=%g dz=%g dz1=%g dtx=%g
dtx1=%g\n",
//      fwidth, g, olp, cyy, dx, dy, dz, dz1, dtx, dtx1);
l_Fxa = Fx*vltg;
l_Fya = Fy*vltg;
l_Fza = -Fz*vltg;

l_Fxb = -Fx*vltg;
l_Fyb = -Fy*vltg;
l_Fzb = Fz*vltg;

l_Tqxa = dCphix*vltg + l_Fza*1*moment_arm_a;
l_Tqya = dCphiy*vltg;
l_Tqza = dCphiz*vltg + l_Fxa*1*moment_arm_a;

l_Tqxb = -dCphix*vltg - l_Fzb*1*moment_arm_b;
l_Tqyb = -dCphiy*vltg;
l_Tqzb = -dCphiz*vltg - l_Fxb*1*moment_arm_b;

//bending forces/moments transformed from local frame back to chip
frame
Fchipxb = inv_l1*1_Fxb + inv_m1*1_Fyb + inv_n1*1_Fzb;
Fchipyb = inv_l2*1_Fxb + inv_m2*1_Fyb + inv_n2*1_Fzb;
Fchipzb = inv_l3*1_Fxb + inv_m3*1_Fyb + inv_n3*1_Fzb;

Fchipxa = inv_l1*1_Fxa + inv_m1*1_Fya + inv_n1*1_Fza;
Fchipya = inv_l2*1_Fxa + inv_m2*1_Fya + inv_n2*1_Fza;
Fchipza = inv_l3*1_Fxa + inv_m3*1_Fya + inv_n3*1_Fza;

Tqchipxb = inv_l1*1_Tqxb + inv_m1*1_Tqyb + inv_n1*1_Tqzb;
Tqchipyb = inv_l2*1_Tqxb + inv_m2*1_Tqyb + inv_n2*1_Tqzb;
Tqchipzb = inv_l3*1_Tqxb + inv_m3*1_Tqyb + inv_n3*1_Tqzb;

```

```

Tqchipxa = inv_l1*1_Tqxa + inv_m1*1_Tqya + inv_n1*1_Tqza;
Tqchipya = inv_l2*1_Tqxa + inv_m2*1_Tqya + inv_n2*1_Tqza;
Tqchipza = inv_l3*1_Tqxa + inv_m3*1_Tqya + inv_n3*1_Tqza;

// bending forces/torques applied to the comb terminals
F(xa[0]) <+ Fchipxa;
F(xa[1]) <+ Fchipya;
F(xa[2]) <+ Fchipza;

F(xb[0]) <+ Fchipxb;
F(xb[1]) <+ Fchipyb;
F(xb[2]) <+ Fchipzb;

Tau(phia[2]) <+ Tqchipza;
Tau(phib[2]) <+ Tqchipzb;
Tau(phia[1]) <+ Tqchipya;
Tau(phib[1]) <+ Tqchipyb;
Tau(phia[0]) <+ Tqchipxa;
Tau(phib[0]) <+ Tqchipxb;

end

endmodule // comb_curl

```

efringe.va

```

function real efringe;
    input width, gap1, dx1;
    real    width, gap1, dx1;

    real Pi, gplus, gminus, alpha1, gamma1, q, p, fringe0, fringe1;
    begin
        Pi = 3.14159;
        gplus = (gap1 + dx1)/2.0;
        gminus = (gap1 - dx1)/2.0;
        alpha1 = (width + gap1)/gminus;
        gamma1 = gplus/gminus;

        q = (1.0/2.0)*(alpha1*alpha1 - gamma1*gamma1 - 1 +
sqrt((alpha1*alpha1 - gamma1*gamma1 - 1)*(alpha1*alpha1 - gamma1*gamma1
- 1) -
    4*gamma1*gamma1));
        p = q*q/(gamma1*gamma1);

        // gap1 minus dx1
        // Keep in mind that this contains the fringe for both the top and the
        bottom
        fringe0 = (1.0/Pi)*(2*alpha1*atanh(sqrt(p+q)/sqrt(p*(1+q))) -
    2*gamma1*atanh(1/sqrt(p)) -
    ln(4*p/(p-1)) +
    ln(q));

        // gap1 plus dx1
        fringe1 = (1.0/Pi)*((1.0/gamma1)*(2*alpha1*atanh(sqrt((1+q)/(p+q))) +

```

```

    gamma1*ln((p-1)/(4)) -
    2*atanh(1/sqrt(p)) - ln(q));
efringe = fringe0 + fringel;

// $display("width=%g gap=%g dx=%g alpha1 = %g gamma1 = %g q = %g p = %g
fringe0 = %g fringel = %g\n",
// width, gap1, dx1, alpha1, gamma1, q, p, fringe0, fringel);

    end

endfunction // efringe

function real defringe;
    input width, gap1, dx1;
    real    width, gap1, dx1;

    real Pi, gplus, gminus, alpha1, gamma1, q, p, fringe0, fringel;
    real    dalpha1, dgamma1, dalpha1_gamma1, dq, dp, dfringe0,
dfringel;
    begin
    Pi = 3.14159;
    gplus = (gap1 + dx1)/2.0;
    gminus = (gap1 - dx1)/2.0;
    alpha1 = (width + gap1)/gminus;
    gamma1 = gplus/gminus;

    q = (1.0/2.0)*(alpha1*alpha1 - gamma1*gamma1 - 1 +
sqrt((alpha1*alpha1 - gamma1*gamma1 - 1)*(alpha1*alpha1 - gamma1*gamma1
- 1) -
    4*gamma1*gamma1));
    p = q*q/(gamma1*gamma1);

    dalpha1 = 2*(gap1+width)/((gap1-dx1)*(gap1-dx1));
    dgamma1 = (gap1+dx1)/((gap1-dx1)*(gap1-dx1)) + 1/(gap1-dx1);
    dalpha1_gamma1 = -2*(gap1+width)/((gap1+dx1)*(gap1+dx1));
    dq = (1.0/2.0)*(2*alpha1*dalpha1 - 2*gamma1*dgamma1 +
(1.0/2.0)*(2*(alpha1*alpha1 - gamma1*gamma1 - 1)*(2*alpha1*dalpha1 -
2*gamma1*dgamma1) -
    8*gamma1*dgamma1)/
sqrt((alpha1*alpha1 - gamma1*gamma1 - 1)*(alpha1*alpha1 - gamma1*gamma1
- 1)-4*gamma1*gamma1));

    dp = 2*q*dq/(gamma1*gamma1) - 2*q*q*dgamma1/(gamma1*gamma1*gamma1);

    // gap1 minus dx1
    // Keep in mind that this contains the fringe for both the top and the
bottom
    fringe0 = (1.0/Pi)*(2*alpha1*atanh(sqrt(p+q)/sqrt(p*(1+q))) -
                2*gamma1*atanh(1/sqrt(p)) -
                ln(4*p/(p-1)) +
                ln(q));
    dfringe0 = (1.0/Pi)*(2.0*dalpha1*atanh(sqrt(p+q)/sqrt(p*(1+q))) +
2.0*alpha1*((p+p*q)/(-q+p*q))*(1.0/2.0)*

```

```

(sqrt(p*(1+q))/sqrt(p+q))* (p*(1+q)*(dp+dq) - (p+q)*(dp+p*dq+q*dp))/
(p*p*(1+q)*(1+q)) -
2.0*dgamma1*atanh(1/sqrt(p)) -
2.0*gamma1*(p/(p-1))*(-1.0/2.0)*pow(p,-3.0/2.0)*dp +
dp/(p-1) - dp/p +
dq/q);

// gap1 plus dx1
fringe1 = (1.0/Pi)*((1.0/gamma1)*(2*alpha1*atanh(sqrt((1+q)/(p+q))) +
gamma1*ln((p-1)/4)) -
2*atanh(1/sqrt(p))) - ln(q));

dfringe1 = (1.0/Pi)*((1.0/gamma1)*(2*dalpha1*atanh(sqrt((1+q)/(p+q))) +
2*alpha1*((p+q)/(p-1))*(1.0/2.0)*
(sqrt((p+q)/(1+q))*((p+q)*dq - (1+q)*(dp+dq))/((p+q)*(p+q))) +
dgamma1*ln((p-1)/4) + gamma1*(4/(p-1))*dp/4 -
2*p/(p-1)*(-1.0/2.0)*dp/(p*sqrt(p))) -
(dgamma1/(gamma1*gamma1))*(2*alpha1*atanh(sqrt((1+q)/(p+q))) +
gamma1*ln((p-1)/4)) - 2*atanh(1/sqrt(p))) -
dq/q);

defringe = dfringe0 + dfringe1;

// $display("alpha1 = %g gamma1 = %g q = %g p = %g fringe0 = %g fringe1 =
%g dfringe0 = %g dfringe1 = %g\n",
// alpha1, gamma1, q, p, fringe0, fringe1, dfringe0, dfringe1);

end

endfunction // defringe

function real pptotal;
    input thickness,overlap,gap1,dx1,dy;
    real thickness,overlap,gap1,dx1,dy;

    begin
        pptotal = thickness*(overlap + dy)*(2*gap1/(gap1*gap1 - dx1*dx1));
    end
endfunction // pptotal

function real dpptotalx;
    input thickness,overlap,gap1,dx1,dy;
    real thickness,overlap,gap1,dx1,dy;

    begin
        dpptotalx = thickness*(overlap + dy)*(4*gap1*dx1/((gap1*gap1 -
dx1*dx1)*(gap1*gap1 - dx1*dx1)));
    end
endfunction // dpptotalx

```

```

function real dpptotaly;
    input thickness,overlap,gap1,dx1,dy;
    real thickness,overlap,gap1,dx1,dy;

    begin

    dpptotaly = thickness*(1/(gap1 - dx1) + 1/(gap1 + dx1));
    end
endfunction // dpptotaly

// Single sided functions
function real efringep;
    input width, gap1, dx1;
    real width, gap1, dx1;

    real Pi, gplus, gminus, alpha1, gamma1, q, p, fringe0, fringe1;
    begin
    Pi = 3.14159;
    gplus = (gap1 + dx1)/2.0;
    gminus = (gap1 - dx1)/2.0;
    alpha1 = (width + gap1)/gminus;
    gamma1 = gplus/gminus;

    // $strobe("width=%g gap=%g dx=%g alpha1 = %g gamma1 = %g q = %g p = %g
    fringe0 = %g fringe1 = %g\n",
    // width, gap1, dx1, alpha1, gamma1, q, p, fringe0, fringe1);
    q = (1.0/2.0)*(alpha1*alpha1 - gamma1*gamma1 - 1 +
    sqrt((alpha1*alpha1 - gamma1*gamma1 - 1)*(alpha1*alpha1 - gamma1*gamma1
    - 1) -
    4*gamma1*gamma1));
    p = q*q/(gamma1*gamma1);

    // gap1 minus dx1
    // Keep in mind that this contains the fringe for both the top and the
    bottom
    fringe0 = (1.0/Pi)*(2*alpha1*atanh(sqrt(p+q)/sqrt(p*(1+q))) -
    2*gamma1*atanh(1/sqrt(p)) -
    ln(4*p/(p-1)) +
    ln(q));

    // gap1 plus dx1
    fringe1 = (1.0/Pi)*((1.0/gamma1)*(2*alpha1*atanh(sqrt((1+q)/(p+q))) +
    gamma1*ln((p-1)/(4)) -
    2*atanh(1/sqrt(p))) - ln(q));

    // Return gap + dx value
    efringep = fringe1;

    end

```

```

endfunction // efringep

function real defringep;
    input width, gap1, dx1;
    real    width, gap1, dx1;

    real Pi, gplus, gminus, alpha1, gamma1, q, p, fringe0, fringe1;
    real    dalpha1, dgamma1, dalpha1_gamma1, dq, dp, dfringe0,
dfringe1;
    begin
    Pi = 3.14159;
    gplus = (gap1 + dx1)/2.0;
    gminus = (gap1 - dx1)/2.0;
    alpha1 = (width + gap1)/gminus;
    gamma1 = gplus/gminus;

    q = (1.0/2.0)*(alpha1*alpha1 - gamma1*gamma1 - 1 +
sqrt((alpha1*alpha1 - gamma1*gamma1 - 1)*(alpha1*alpha1 - gamma1*gamma1
- 1) -
    4*gamma1*gamma1));
    p = q*q/(gamma1*gamma1);

    dalpha1 = 2*(gap1+width)/((gap1-dx1)*(gap1-dx1));
    dgamma1 = (gap1+dx1)/((gap1-dx1)*(gap1-dx1)) + 1/(gap1-dx1);
    dalpha1_gamma1 = -2*(gap1+width)/((gap1+dx1)*(gap1+dx1));
    dq = (1.0/2.0)*(2*alpha1*dalpha1 - 2*gamma1*dgamma1 +
(1.0/2.0)*(2*(alpha1*alpha1 - gamma1*gamma1 - 1)*(2*alpha1*dalpha1 -
2*gamma1*dgamma1) -
    8*gamma1*dgamma1)/
sqrt((alpha1*alpha1 - gamma1*gamma1 - 1)*(alpha1*alpha1 - gamma1*gamma1
- 1)-4*gamma1*gamma1));

    dp = 2*q*dq/(gamma1*gamma1) - 2*q*q*dgamma1/(gamma1*gamma1*gamma1);

    // gap1 minus dx1
    // Keep in mind that this contains the fringe for both the top and the
bottom
    fringe0 = (1.0/Pi)*(2*alpha1*atanh(sqrt(p+q)/sqrt(p*(1+q))) -
                2*gamma1*atanh(1/sqrt(p)) -
                ln(4*p/(p-1)) +
                ln(q));
    dfringe0 = (1.0/Pi)*(2.0*dalpha1*atanh(sqrt(p+q)/sqrt(p*(1+q))) +
2.0*alpha1*((p+p*q)/(-q+p*q))*(1.0/2.0)*
(sqrt(p*(1+q))/sqrt(p+q))*(p*(1+q)*(dp+dq) - (p+q)*(dp+p*dq+q*dp))/
(p*p*(1+q)*(1+q)) -
2.0*dgamma1*atanh(1/sqrt(p)) -
2.0*gamma1*(p/(p-1))*(-1.0/2.0)*pow(p,-3.0/2.0)*dp +
dp/(p-1) - dp/p +
dq/q);

    // gap1 plus dx1
    fringe1 = (1.0/Pi)*((1.0/gamma1)*(2*alpha1*atanh(sqrt((1+q)/(p+q))) +
gamma1*ln((p-1)/(4)) -

```

```

2*atanh(1/sqrt(p)) - ln(q));

dfringe1 = (1.0/Pi)*((1.0/gamma1)*(2*dalpha1*atanh(sqrt((1+q)/(p+q))) +
2*alpha1*((p+q)/(p-1))*(1.0/2.0)*
(sqrt((p+q)/(1+q))*((p+q)*dq - (1+q)*(dp+dq))/((p+q)*(p+q))) +
dgamma1*ln((p-1)/4) + gamma1*(4/(p-1))*dp/4 -
2*p/(p-1)*(-1.0/2.0)*dp/(p*sqrt(p))) -
(dgamma1/(gamma1*gamma1))*(2*alpha1*atanh(sqrt((1+q)/(p+q))) +
gamma1*ln((p-1)/(4)) - 2*atanh(1/sqrt(p))) -
dq/q);

// Return gap + dx value
defringep = dfringe1;

// $display("alpha1 = %g gamma1 = %g q = %g p = %g fringe0 = %g fringe1 =
%g dfringe0 = %g dfringe1 = %g\n",
// alpha1, gamma1, q, p, fringe0, fringe1, dfringe0, dfringe1);

end

endfunction // defringep

function real pptotalp;
input thickness,overlap,gap1,dx1,dy;
real thickness,overlap,gap1,dx1,dy;

begin
pptotalp = thickness*(overlap + dy)*(-1/((gap1 + dx1)*(gap1 + dx1)));
end
endfunction // pptotalp

function real dpptotalpx;
input thickness,overlap,gap1,dx1,dy;
real thickness,overlap,gap1,dx1,dy;

begin
dpptotalpx = thickness*(overlap + dy)*(-1/((gap1 + dx1)*(gap1 + dx1)));
end
endfunction // dpptotalpx

function real dpptotalpy;
input thickness,overlap,gap1,dx1,dy;
real thickness,overlap,gap1,dx1,dy;

begin

dpptotalpy = thickness*(1/(gap1 + dx1));
end
endfunction // dpptotalpy

```

cap.va

```

thickness = 4.5;
ppcapttotal = thickness*(olp + dy)*(1/(g + dx) + 1/(g - dx));
dppcapttotalx = dppttotalx(thickness,olp,g,dx,dy);
dppcapttotaly = dppttotaly(thickness,olp,g,dx,dy);
efcapttotal = efringe(fwidth, g, dx);
defcapttotal = defringe(fwidth, g, dx);
@(initial_step("ac","dc","tran")) begin
    c0 = 0.0276373107344633;
    c1 = -0.036654;
    c2 = -0.0363877;
    c3 = -49.4764;
    c4 = 114.19;
    c5 = -4586.57;
    c6 = -0.00139927;
    c7 = 0.926732;
    c8 = 0.094436;
    c9 = -0.0572805;
    c10 = 1.31469;
    c11 = -1.88933;
    c12 = 0.00653463;
    c13 = 97.8598;
    c14 = -1248.49;
    c15 = -4.8071;
    c16 = -0.00433067;
    c17 = 0.254919;
    c18 = -0.215126;
    c19 = -0.00244782;
    c20 = -0.00601334;
    c21 = -0.00252484;
    c22 = -4.89106;
    c23 = 0.0343773;
    c24 = 136.489;
    c25 = -0.0350893;
    c26 = -0.011512;
    c27 = 0.0103864;
    c28 = -0.12185;
    c29 = -0.0306053;
    c30 = 369779;
    c31 = 0.0019406;
    c32 = -304.803;
    c33 = -0.177552;
    c34 = 0.012253;
    c35 = -0.000267055;
    c36 = 0.00242414;
    c37 = -2742.33;
    c38 = 850666;
    c39 = 879.931;
    c40 = 10.7562;
    c41 = -0.000721616;
    c42 = -0.0073036;
    c43 = 0.0177324;
    c44 = -11096.8;
    c45 = 0.0285413;
    c46 = -0.24904;

```

c47 = 202.283;
c48 = 0.012807;
c49 = 0.000136373;
c50 = 100.469;
c51 = -1.69623;
c52 = 63.838;
c53 = 0.445681;
c54 = -206898;
c55 = 0.0276114;
c56 = 13.0536;
c57 = 0.959911;
c58 = -0.00741532;
c59 = 1.19818;
c60 = 0.0803255;
c61 = 32432.5;
c62 = 5.49038;
c63 = -5.23022;
c64 = 2.99633;
c65 = -0.0239581;
c66 = -0.230001;
c67 = 153.515;
c68 = 1.78854;
c69 = -0.0455665;
c70 = 1.14643;
c71 = 0.036121;
c72 = -858.224;
c73 = -7.32389;
c74 = -24.7476;
c75 = -0.0874021;
c76 = 90599.6;
c77 = -0.00403123;
c78 = -5.02762e-06;
c79 = -6.24865e+07;
c80 = 0.000903951;
c81 = 0.0135729;
c82 = 0.000232695;
c83 = 0.00104478;
c84 = -0.00324096;
c85 = 0.184985;
c86 = -32.5984;
c87 = 0.084973;
c88 = 0.01552;
c89 = -26.5471;
c90 = -33.4237;
c91 = 0.0022785;
c92 = -50.8039;
c93 = -51.879;
c94 = -3746.48;
c95 = 11108.5;
c96 = -0.000548988;
c97 = -1.17128;
c98 = 4149.19;
c99 = -362530;
c100 = 0.00312093;

```

c101 = -0.0539101;
c102 = 0.276848;
c103 = 0.00182284;
c104 = 17.6506;
c105 = 0.0136148;
end
cap_curl = (c0 * ((olp+dy)*efcapttotal)*(dtx1)) +
(c1 * (dy*dz1)) +
(c2 * ppcapttotal*((g/fwidth)*dz1)) +
(c3 * ((olp+dy)*efcapttotal)*((g/fwidth)*dz1*cy)) +
(c4 * ((olp+dy)*efcapttotal)*(cy*dtx1)) +
(c5 * ((olp*cy*dz1)/(g-dx)+(olp*cy*dz1)/(g+dx))) +
(c6 * ((olp+dy)*efcapttotal)*(olp*dz1*dtx1)) +
(c7 * (dz1*dtx1)) +
(c8 * (olp*dz1)) +
(c9 * ((olp+dy)*efcapttotal)*((g/fwidth)*dz1*dtx1)) +
(c10 * ppcapttotal) +
(c11 * (dy)) +
(c12 * (olp*fwidth*dz1)) +
(c13 * ((cy*dy*dy*dz1)/(g-dx)+(cy*dy*dy*dz1)/(g+dx))) +
(c14 * ((g/fwidth)*dz1*dtx1*cy)) +
(c15 * ppcapttotal*((g/fwidth)*dz1*cy)) +
(c16 * ((olp+dy)*efcapttotal)*(dz1*dtx1)) +
(c17 * (dtx1*dtx1)) +
(c18 * ((olp+dy)*efcapttotal)*(fwidth)) +
(c19 * (dy*dy)*((olp+dy)*efcapttotal)) +
(c20 * ((olp+dy)*efcapttotal)*(dy*dz1)) +
(c21 * ((olp+dy)*efcapttotal)*(olp*dtx1)) +
(c22 * ppcapttotal*(cy*(dy*dy))) +
(c23 * (olp*dz1*dtx1)) +
(c24 * ((olp+dy)*efcapttotal)*((g/fwidth)*dz1*dtx1*cy)) +
(c25 * ppcapttotal*(dtx1*dtx1)) +
(c26 * (olp*dtx1)) +
(c27 * (dy*fwidth*dz1)) +
(c28 * (g*fwidth*dtx1)) +
(c29 * ((olp+dy)*efcapttotal)*(g*dz1*dtx1)) +
(c30 * (cy)) +
(c31 * ((olp+dy)*efcapttotal)*(g*fwidth*dz1)) +
(c32 * ((olp*cy*dtx1*dy)/(g-dx)+(olp*cy*dtx1*dy)/(g+dx))) +
(c33 * (g*dz1*dtx1)) +
(c34 * (dy*dy)) +
(c35 * ppcapttotal*(g*fwidth*dtx1)) +
(c36 * ppcapttotal*(olp*dz1*dtx1)) +
(c37 * ((olp+dy)*efcapttotal)*(cy*(g*g))) +
(c38 * (cy*cy)*((olp+dy)*efcapttotal)) +
(c39 * ppcapttotal*(cy)) +
(c40 * ((olp+dy)*efcapttotal)) +
(c41 * (g*fwidth*dy*dz1)) +
(c42 * ppcapttotal*(dy)) +
(c43 * (dtx1*dtx1)*((olp+dy)*efcapttotal)) +
(c44 * ((olp+dy)*efcapttotal)*(cy)) +
(c45 * ppcapttotal*(g*dz1)) +
(c46 * ppcapttotal*(g*g)) +
(c47 * ppcapttotal*(cy*(g*g))) +

```

```

(c48 * (g*fwidth*dz1)) +
(c49 * ppcapttotal*(olp*dz1)) +
(c50 * (g)) +
(c51 * (dz1)) +
(c52 * ((olp*cyy*dz1*dz1*dtx1*dy)/(g-dx)+(olp*cyy*dz1*dz1*dtx1*dy)/
(g+dx))) +
(c53 * (fwidth*fwidth)) +
(c54 * ppcapttotal*(cyy*cyy)) +
(c55 * ((olp+dy)*efcapttotal)*(dy)) +
(c56 * ppcapttotal*(cyy*dy)) +
(c57 * ppcapttotal*(g)) +
(c58 * ppcapttotal*(dz1)) +
(c59 * (g*g)*((olp+dy)*efcapttotal)) +
(c60 * ((olp+dy)*efcapttotal)*(dz1)) +
(c61 * ((cyy*dz1)/(g-dx)+(cyy*dz1)/(g+dx))) +
(c62 * ((olp+dy)*efcapttotal)*(cyy*(dy*dy))) +
(c63 * ((olp+dy)*efcapttotal)*(g)) +
(c64 * ((olp*cyy*dy*dy*dz1)/(g-dx)+(olp*cyy*dy*dy*dz1)/(g+dx))) +
(c65 * ((olp+dy)*efcapttotal)*(g*dz1)) +
(c66 * ppcapttotal*(dz1*dz1)) +
(c67 * (((olp+dy)*(olp+dy))*cyy*dz1)/(g-
dx)+(((olp+dy)*(olp+dy))*cyy*dz1)/(g+dx))) +
(c68 * ppcapttotal*(cyy*dtx1)) +
(c69 * (olp*fwidth*dz1*dz1)) +
(c70 * (dtx1)) +
(c71 * ((olp+dy)*efcapttotal)*((g/fwidth)*dz1)) +
(c72 * ppcapttotal*(cyy*g)) +
(c73 * ppcapttotal*(cyy*dz1)) +
(c74 * (g*g)) +
(c75 * (g*dz1)) +
(c76 * (cyy*(g*g))) +
(c77 * ppcapttotal*(g*fwidth*dz1)) +
(c78 * ppcapttotal*(olp*dtx1)) +
(c79 * (cyy*cyy)) +
(c80 * ppcapttotal*(dy*dy)) +
(c81 * ppcapttotal*(g*dz1*dtx1)) +
(c82 * ppcapttotal*(g*fwidth*dy*dz1)) +
(c83 * ppcapttotal*(dtx1)) +
(c84 * ((olp+dy)*efcapttotal)*(olp*dz1)) +
(c85 * (dx*dx)*((olp+dy)*efcapttotal)) +
(c86 * ((olp+dy)*efcapttotal)*(cyy*dy)) +
(c87 * ppcapttotal*(dz1*dtx1)) +
(c88 * ppcapttotal*((g/fwidth)*dz1*dtx1)) +
(c89 * (cyy*(dy*dy))) +
(c90 * ((olp*cyy*dz1*dtx1*dy)/(g-dx)+(olp*cyy*dz1*dtx1*dy)/(g+dx))) +
(c91 * ((olp+dy)*efcapttotal)*(g*fwidth*dtx1)) +
(c92 * ((olp+dy)*efcapttotal)*(cyy*dz1)) +
(c93 * ppcapttotal*((g/fwidth)*dz1*dtx1*cyy)) +
(c94 * (cyy*dtx1)) +
(c95 * ((olp+dy)*efcapttotal)*(cyy*g)) +
(c96 * ((olp+dy)*efcapttotal)*(g*fwidth*dy*dz1)) +
(c97 * (dz1*dz1)) +
(c98 * (cyy*dz1)) +
(c99 * (cyy*g)) +

```

```

(c100 * ppcapttotal*(fwidth)) +
(c101 * (dy*fwidth*dz1*dz1)) +
(c102 * (dz1*dz1)*((olp+dy)*efcapttotal)) +
(c103 * ppcapttotal*(dy*dz1)) +
(c104 * (fwidth)) +
(c105 * ppcapttotal*(dx*dx));

```

force.va

```

Fx =
c61*((cyy*dz1)/((-dx + g)*(-dx + g)) - (cyy*dz1)/((dx + g)*(dx + g))) +
c13*((cyy*((dy)*(dy))*dz1)/((-dx + g)*(-dx + g)) -
(cyy*((dy)*(dy))*dz1)/((dx + g)*(dx + g))) +
c32*((cyy*dtx1*dy*olp)/((-dx + g)*(-dx + g)) - (cyy*dtx1*dy*olp)/((dx +
g)*(dx + g))) +
c5*((cyy*dz1*olp)/((-dx + g)*(-dx + g)) - (cyy*dz1*olp)/((dx + g)*(dx +
g))) +
c90*((cyy*dtx1*dy*dz1*olp)/((-dx + g)*(-dx + g)) -
(cyy*dtx1*dy*dz1*olp)/((dx + g)*(dx + g))) +
c64*((cyy*((dy)*(dy))*dz1*olp)/((-dx + g)*(-dx + g)) -
(cyy*((dy)*(dy))*dz1*olp)/((dx + g)*(dx + g))) +
c52*((cyy*dtx1*dy*((dz1)*(dz1))*olp)/((-dx + g)*(-dx + g)) -
(cyy*dtx1*dy*((dz1)*(dz1))*olp)/((dx + g)*(dx + g))) +
c67*((cyy*dz1*((dy + olp)*(dy + olp)))/((-dx + g)*(-dx + g)) -
(cyy*dz1*((dy + olp)*(dy + olp)))/((dx + g)*(dx + g))) +
2*c85*dx*(dy + olp)*efcapttotal +
2*c105*dx*ppcapttotal +
c40*(dy + olp)*defcapttotal +
c44*cyy*(dy + olp)*defcapttotal +
c38*((cyy)*(cyy))*(dy + olp)* defcapttotal +
c0*dtx1*(dy + olp)*defcapttotal +
c4*cyy*dtx1*(dy + olp)*defcapttotal +
c43*((dtx1)*(dtx1))*(dy + olp)* defcapttotal +
c85*((dx)*(dx))*(dy + olp)*defcapttotal +
c55*dy*(dy + olp)*defcapttotal +
c86*cyy*dy*(dy + olp)*defcapttotal +
c19*((dy)*(dy))*(dy + olp)*defcapttotal +
c62*cyy*((dy)*(dy))*(dy + olp)* defcapttotal +
c60*dz1*(dy + olp)*defcapttotal +
c92*cyy*dz1*(dy + olp)*defcapttotal +
c16*dtx1*dz1*(dy + olp)*defcapttotal +
c20*dy*dz1*(dy + olp)*defcapttotal +
c102*((dz1)*(dz1))*(dy + olp)* defcapttotal +
c18*fwidth*(dy + olp)*defcapttotal +
c63*g*(dy + olp)*defcapttotal +
c95*cyy*g*(dy + olp)*defcapttotal +
c65*dz1*g*(dy + olp)*defcapttotal +
c29*dtx1*dz1*g*(dy + olp)* defcapttotal +
(c71*dz1*g*(dy + olp)*defcapttotal)/ fwidth +
(c3*cyy*dz1*g*(dy + olp)* defcapttotal)/fwidth +
(c9*dtx1*dz1*g*(dy + olp)* defcapttotal)/fwidth +
(c24*cyy*dtx1*dz1*g*(dy + olp)* defcapttotal)/fwidth +
c91*dtx1*fwidth*g*(dy + olp)* defcapttotal +
c31*dz1*fwidth*g*(dy + olp)* defcapttotal +
c96*dy*dz1*fwidth*g*(dy + olp)* defcapttotal +

```

```

c59*((g)*(g))*(dy + olp)* defcapttotal +
c37*cy*(g)*(g))*(dy + olp)* defcapttotal +
c21*dtx1*olp*(dy + olp)* defcapttotal +
c84*dz1*olp*(dy + olp)*defcapttotal +
c6*dtx1*dz1*olp*(dy + olp)* defcapttotal +
c10*dppcapttotalx +
c39*cy*dppcapttotalx +
c54*((cy)*(cy))*dppttotalx(thickness,olp,g,dx, dy) +
c83*dtx1*dppttotalx(thickness,olp,g,dx, dy) +
c68*cy*dtx1*dppttotalx(thickness,olp,g, dx,dy) +
c25*((dtx1)*(dtx1))* dppcapttotalx +
c105*((dx)*(dx))*dppttotalx(thickness,olp,g,dx, dy) +
c42*dy*dppttotalx(thickness,olp,g,dx, dy) +
c56*cy*dy*dppttotalx(thickness,olp,g, dx,dy) +
c80*((dy)*(dy))*dppttotalx(thickness, olp,g,dx,dy) +
c22*cy*((dy)*(dy))* dppcapttotalx +
c58*dz1*dppcapttotalx +
c73*cy*dz1*dppcapttotalx +
c87*dtx1*dz1*dppcapttotalx +
c103*dy*dz1*dppcapttotalx +
c66*((dz1)*(dz1))*dppttotalx(thickness,olp,g,dx, dy) +
c100*fwidth*dppttotalx(thickness,olp,g, dx,dy) +
c57*g*dppttotalx(thickness,olp,g, dx,dy) +
c72*cy*g*dppttotalx(thickness,olp, g,dx,dy) +
c45*dz1*g*dppttotalx(thickness, olp,g,dx,dy) +
c81*dtx1*dz1*g* dppcapttotalx +
(c2*dz1*g*dppcapttotalx)/ fwidth +
(c15*cy*dz1*g*dppttotalx(thickness, olp,g,dx,dy))/fwidth +
(c88*dtx1*dz1*g*dppttotalx(thickness,olp,g, dx,dy))/fwidth +
(c93*cy*dtx1*dz1*g* dppcapttotalx)/fwidth +
c35*dtx1*fwidth*g*dppttotalx(thickness,olp,g, dx,dy) +
c77*dz1*fwidth*g* dppcapttotalx +
c82*dy*dz1*fwidth*g*dppttotalx(thickness,olp, g,dx,dy) +
c46*((g)*(g))* dppcapttotalx +
c47*cy*((g)*(g))*dppttotalx(thickness,olp,g, dx,dy) +
c78*dtx1*olp*dppttotalx(thickness, olp,g,dx,dy) +
c49*dz1*olp* dppcapttotalx +
c36*dtx1*dz1*olp*dppttotalx(thickness,olp,g, dx,dy) ;
Fy =
c11 +
2*c34*dy +
2*c89*cy*dy +
c1*dz1 +
c27*dz1*fwidth +
c101*((dz1)*(dz1))*fwidth +
c41*dz1*fwidth*g +
c13*((2*cy*dy*dz1)/(-dx + g) + (2*cy*dy*dz1)/(dx + g)) +
c32*((cy*dtx1*olp)/(-dx + g) + (cy*dtx1*olp)/(dx + g)) +
c90*((cy*dtx1*dz1*olp)/(-dx + g) + (cy*dtx1*dz1*olp)/(dx + g)) +
c64*((2*cy*dy*dz1*olp)/(-dx + g) + (2*cy*dy*dz1*olp)/(dx + g)) +
c52*((cy*dtx1*((dz1)*(dz1))*olp)/(-dx + g) +
(cy*dtx1*((dz1)*(dz1))*olp)/(dx + g)) +
c67*((2*cy*dz1*(dy + olp))/(-dx + g) + (2*cy*dz1*(dy + olp))/(dx + g))
+

```

c40*efcapttotal +
c44*cyy*efcapttotal +
c38*((cyy)*(cyy))*efcapttotal +
c0*dtx1*efcapttotal +
c4*cyy*dtx1*efcapttotal +
c43*((dtx1)*(dtx1))*efcapttotal +
c85*((dx)*(dx))*efcapttotal +
c55*dy*efcapttotal +
c86*cyy*dy*efcapttotal +
c19*((dy)*(dy))*efcapttotal +
c62*cyy*((dy)*(dy))*efcapttotal +
c60*dz1*efcapttotal +
c92*cyy*dz1*efcapttotal +
c16*dtx1*dz1*efcapttotal +
c20*dy*dz1*efcapttotal +
c102*((dz1)*(dz1))*efcapttotal +
c18*fwidth*efcapttotal +
c63*g*efcapttotal +
c95*cyy*g*efcapttotal +
c65*dz1*g*efcapttotal +
c29*dtx1*dz1*g*efcapttotal +
(c71*dz1*g*efcapttotal)/fwidth +
(c3*cyy*dz1*g*efcapttotal)/fwidth +
(c9*dtx1*dz1*g*efcapttotal)/fwidth +
(c24*cyy*dtx1*dz1*g*efcapttotal)/fwidth +
c91*dtx1*fwidth*g*efcapttotal +
c31*dz1*fwidth*g*efcapttotal +
c96*dy*dz1*fwidth*g*efcapttotal +
c59*((g)*(g))*efcapttotal +
c37*cyy*((g)*(g))*efcapttotal +
c21*dtx1*olp*efcapttotal +
c84*dz1*olp*efcapttotal +
c6*dtx1*dz1*olp*efcapttotal +
c55*(dy + olp)*efcapttotal +
c86*cyy*(dy + olp)*efcapttotal +
2*c19*dy*(dy + olp)*efcapttotal +
2*c62*cyy*dy*(dy + olp)*efcapttotal +
c20*dz1*(dy + olp)*efcapttotal +
c96*dz1*fwidth*g*(dy + olp)*efcapttotal +
c42*ppcapttotal +
c56*cyy*ppcapttotal +
2*c80*dy*ppcapttotal +
2*c22*cyy*dy*ppcapttotal +
c103*dz1*ppcapttotal +
c82*dz1*fwidth*g*ppcapttotal +
c10*dppcapttotaly +
c39*cyy*dppcapttotaly +
c54*((cyy)*(cyy))*dppttotaly(thickness,olp,g,dx, dy) +
c83*dtx1*dppttotaly(thickness,olp,g,dx, dy) +
c68*cyy*dtx1*dppttotaly(thickness,olp,g, dx,dy) +
c25*((dtx1)*(dtx1))* dppttotaly +
c105*((dx)*(dx))*dppttotaly(thickness,olp,g,dx, dy) +
c42*dy*dppttotaly(thickness,olp,g,dx, dy) +
c56*cyy*dy*dppttotaly(thickness,olp,g, dx,dy) +

```

c80*((dy)*(dy))*dpptotaly(thickness,  olp,g,dx,dy) +
c22*cyy*((dy)*(dy))*  dppcaptotaly +
c58*dz1*dppcaptotaly +
c73*cyy*dz1*dppcaptotaly +
c87*dtx1*dz1*dppcaptotaly +
c103*dy*dz1*dppcaptotaly +
c66*((dz1)*(dz1))*dpptotaly(thickness,olp,g,dx,  dy) +
c100*fwidth*dpptotaly(thickness,olp,g,  dx,dy) +
c57*g*dpptotaly(thickness,olp,g,  dx,dy) +
c72*cyy*g*dpptotaly(thickness,olp,  g,dx,dy) +
c45*dz1*g*dpptotaly(thickness,  olp,g,dx,dy) +
c81*dtx1*dz1*g*  dppcaptotaly +
(c2*dz1*g*dppcaptotaly)/  fwidth +
(c15*cyy*dz1*g*dpptotaly(thickness,  olp,g,dx,dy))/fwidth +
(c88*dtx1*dz1*g*dpptotaly(thickness,olp,g,  dx,dy))/fwidth +
(c93*cyy*dtx1*dz1*g*  dppcaptotaly)/fwidth +
c35*dtx1*fwidth*g*dpptotaly(thickness,olp,g,  dx,dy) +
c77*dz1*fwidth*g*  dppcaptotaly +
c82*dy*dz1*fwidth*g*dpptotaly(thickness,olp,  g,dx,dy) +
c46*((g)*(g))*  dppcaptotaly +
c47*cyy*((g)*(g))*dpptotaly(thickness,olp,g,  dx,dy) +
c78*dtx1*olp*dpptotaly(thickness,  olp,g,dx,dy) +
c49*dz1*olp*  dppcaptotaly +
c36*dtx1*dz1*olp*dpptotaly(thickness,olp,g,  dx,dy) ;
Fz =
c51 +
c98*cyy +
c7*dtx1 +
c1*dy +
2*c97*dz1 +
c27*dy*fwidth +
2*c101*dy*dz1*fwidth +
c75*g +
c33*dtx1*g +
(c14*cyy*dtx1*g)/fwidth +
c48*fwidth*g +
c41*dy*fwidth*g +
c61*(cyy/(-dx + g) + cyy/(dx + g)) +
c13*((cyy*((dy)*(dy)))/(-dx + g) + (cyy*((dy)*(dy)))/(dx + g)) +
c8*olp +
c23*dtx1*olp +
c12*fwidth*olp +
2*c69*dz1*fwidth*olp +
c5* ((cyy*olp)/(-dx + g) + (cyy*olp)/(dx + g)) +
c90*((cyy*dtx1*dy*olp)/(-dx + g) + (cyy*dtx1*dy*olp)/(dx + g)) +
c64*((cyy*((dy)*(dy))*olp)/(-dx + g) + (cyy*((dy)*(dy))*olp)/(dx + g))
+
c52*((2*cyy*dtx1*dy*dz1*olp)/(-dx + g) + (2*cyy*dtx1*dy*dz1*olp)/(dx +
g)) +
c67*((cyy*((dy + olp)*(dy + olp)))/(-dx + g) + (cyy*((dy + olp)*(dy +
olp)))/(dx + g)) +
c60*(dy + olp)*efcapttotal +
c92*cyy*(dy + olp)*efcapttotal +
c16*dtx1*(dy + olp)*efcapttotal +

```

```

c20*dy*(dy + olp)*efcapttotal +
2*c102*dz1*(dy + olp)*efcapttotal +
c65*g*(dy + olp)*efcapttotal +
c29*dtx1*g*(dy + olp)*efcapttotal +
(c71*g*(dy + olp)*efcapttotal)/fwidth +
(c3*cyy*g*(dy + olp)*efcapttotal)/fwidth +
(c9*dtx1*g*(dy + olp)*efcapttotal)/fwidth +
(c24*cyy*dtx1*g*(dy + olp)*efcapttotal)/fwidth +
c31*fwidth*g*(dy + olp)*efcapttotal +
c96*dy*fwidth*g*(dy + olp)*efcapttotal +
c84*olp*(dy + olp)*efcapttotal +
c6*dtx1*olp*(dy + olp)*efcapttotal +
c58*ppcapttotal +
c73*cyy*ppcapttotal +
c87*dtx1*ppcapttotal +
c103*dy*ppcapttotal +
2*c66*dz1*ppcapttotal +
c45*g*ppcapttotal +
c81*dtx1*g*ppcapttotal +
(c2*g*ppcapttotal)/fwidth +
(c15*cyy*g*ppcapttotal)/fwidth +
(c88*dtx1*g*ppcapttotal)/fwidth +
(c93*cyy*dtx1*g*ppcapttotal)/fwidth +
c77*fwidth*g*ppcapttotal +
c82*dy*fwidth*g*ppcapttotal +
c49*olp*ppcapttotal +
c36*dtx1*olp*ppcapttotal ;

```

Appendix A5 Comparison of Rotational and Translational Modal Frequencies

In this appendix we show through approximate analysis that the rotational mode frequency is usually higher or of the order of the higher of the two in-plane translational modes. To the first order we can write the overall rotational stiffness of a rectangular plate suspended by four springs at its corners as:

$$K_{\phi_z\phi_z} = K_{xx}L_y^2 + K_{yy}L_x^2 + 4k_{\phi_z\phi_z} \quad (\text{A5.1})$$

where, K_{xx} is the total stiffness of the four springs in the x direction, K_{yy} is the total stiffness of the four springs in the y direction and $k_{\phi_z\phi_z}$ is the stiffness of each individual spring for rotation about the z axis. The total moment of inertia of the plate is given by:

$$I_{zz} = M \left(\frac{L_x^2 + L_y^2}{3} \right) \quad (\text{A5.2})$$

where, M is the total mass of the plate. Without loss of generality we can assume that the springs are much stiffer in x than in y and that the lengths L_x and L_y are comparable. Then we can write a simplified equation for the resonant frequency of the first ϕ_z mode as:

$$\omega_{\phi_z}^2 = \frac{K_{xx}L_y^2 + K_{yy}L_x^2 + 4k_{\phi_z\phi_z}}{M \left(\frac{L_x^2 + L_y^2}{3} \right)} \approx \frac{K_{xx}}{M \left(\frac{2}{3} \right)} \geq \omega_x^2 \quad (\text{A5.3})$$

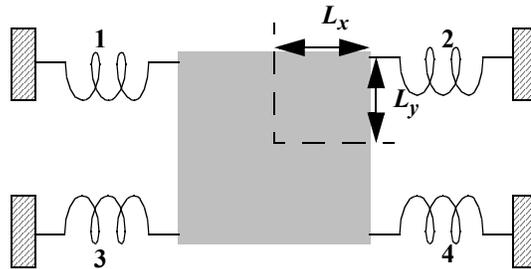


FIGURE A5.1. Rectangular plate suspended by four springs

Note that since we assumed the springs to be stiffer in x than in y , $\omega_x > \omega_y$. Thus we see that the rotational mode is of the same order of the higher translational mode. Normally there will be comb fingers attached to the plate, as is the case of the gyroscope drive mode. However, if the comb fingers are not very long compared to the plate, our result is still approximately valid.

Appendix A6 Beam Widths and Comb Gaps for Monte-Carlo Analysis

Table A6.1 Fractional variation of beam widths and gaps used for Monte-Carlo simulations

No.	w_{o1}	w_{o2}	w_{o3}	w_{o4}	w_{i1}	w_{i2}	w_{i3}	w_{i4}	g_{dt}	g_{db}	g_{st}	g_{sb}
1	-0.433	-0.588	-0.692	1.191	-2.171	0.000	-0.377	0.212	0.781	-0.947	0.129	-0.990
2	-0.546	-0.079	0.133	-0.465	1.957	0.086	0.944	0.587	0.485	0.262	-0.011	-1.021
3	-0.548	-0.896	-0.275	-0.020	-2.449	-0.957	-0.981	1.635	0.702	-0.637	0.672	1.381
4	-0.209	-0.964	0.229	0.652	-2.424	-1.863	-0.146	1.751	1.259	-0.570	-1.301	0.330
5	-0.351	-0.249	1.142	-0.006	1.682	0.249	0.485	0.723	0.018	-0.711	1.151	-0.243
6	0.533	-1.723	0.755	0.411	0.285	-0.607	-0.583	0.900	0.399	0.406	0.760	-0.803
7	0.339	-1.430	0.289	0.441	-0.453	-1.384	0.709	-0.598	-1.008	1.259	-1.414	0.065
8	-1.227	-0.151	-0.911	-0.908	-0.703	1.008	-0.258	0.448	-0.339	-0.720	-0.535	0.256
9	-0.395	-1.333	1.958	-0.650	-1.666	2.183	0.858	-1.202	-0.059	-0.318	-0.296	0.238
10	0.569	-0.374	0.656	1.340	-0.847	1.535	1.593	0.371	0.505	-2.005	-2.120	-0.251
11	-0.005	1.213	-0.001	-0.402	0.261	0.135	2.213	0.279	0.473	1.293	-0.688	-1.256
12	-0.488	-1.003	-0.508	1.320	0.756	-2.379	-0.959	-0.377	-0.224	-0.454	0.248	0.753
13	0.044	-1.499	-0.605	0.599	0.892	-0.150	1.552	0.524	0.594	0.102	0.599	0.039
14	0.818	-1.170	-0.608	0.173	1.033	1.263	0.508	-1.307	0.826	0.687	-0.896	-2.123
15	1.116	1.216	-1.713	-1.665	0.290	-0.150	1.165	0.565	1.399	0.460	-0.748	-0.482
16	-0.665	0.858	-0.384	-1.758	-0.190	0.316	-1.631	-1.044	0.563	0.158	-1.372	0.581
17	0.730	0.758	3.203	-0.191	-0.144	-0.724	-0.073	0.623	0.125	-0.136	1.254	-0.020
18	-1.011	1.095	-1.475	-1.008	-0.822	-1.186	-1.168	0.290	-0.246	-0.606	1.018	0.728
19	1.865	-0.493	-0.645	0.480	-0.276	-0.275	-0.249	0.174	-0.013	-0.139	1.509	1.058
20	0.117	0.441	1.339	-0.214	1.862	-0.186	0.856	-0.909	0.376	-0.838	-0.146	-0.661
21	0.058	-0.652	-0.077	0.065	-0.314	-0.050	-1.489	0.147	1.578	-1.258	1.384	1.364
22	0.790	-0.041	-0.086	1.541	0.702	1.065	0.806	0.923	-1.052	-0.600	0.132	0.384
23	-0.008	0.020	0.249	0.285	0.620	1.448	1.537	-0.901	0.662	-0.505	0.806	-0.694
24	-0.462	0.629	0.229	0.983	0.558	-2.105	-0.458	1.687	-0.302	1.344	-0.359	0.374
25	-0.050	-0.587	-1.268	0.857	0.294	-1.643	0.439	-0.079	-2.334	-0.664	0.939	-1.335
26	0.288	0.114	-1.594	-0.157	0.614	-1.874	-0.234	-0.742	-0.266	-1.056	-0.461	1.479
27	0.663	-1.347	-1.580	2.112	-0.340	0.462	-0.704	0.668	1.276	-0.133	0.397	-0.116
28	-0.580	-1.163	-1.945	0.622	-0.591	1.281	-0.909	-0.199	1.107	-1.054	0.269	-2.306
29	-1.345	0.257	0.745	0.249	-0.425	0.103	1.738	-0.293	0.227	0.553	0.559	-0.101

Table A6.1 Fractional variation of beam widths and gaps used for Monte-Carlo simulations

No.	w_{o1}	w_{o2}	w_{o3}	w_{o4}	w_{i1}	w_{i2}	w_{i3}	w_{i4}	g_{dt}	g_{db}	g_{st}	g_{sb}
30	-1.108	0.313	-0.758	0.482	0.105	-2.248	0.325	-1.701	-0.231	-0.680	0.217	-0.179
31	0.362	-2.064	0.280	0.500	0.858	1.064	-1.490	-0.733	-0.288	-1.025	-1.610	0.588
32	-0.581	-1.729	-1.356	0.834	0.033	0.380	-0.223	1.762	-1.189	-0.361	-0.291	0.327
33	0.957	-2.238	-0.398	0.902	1.164	1.574	-0.895	-0.266	-0.849	-1.057	-1.150	0.700
34	-1.358	0.198	-0.080	1.048	-1.146	1.067	-1.441	-1.604	0.508	0.428	0.118	1.082
35	-1.188	1.472	-0.262	1.138	-0.854	0.469	-0.079	-1.357	-1.140	-0.321	-1.018	-0.078
36	1.863	-1.271	-0.264	1.064	2.136	1.184	-1.681	-1.751	-0.655	-0.498	-0.413	0.307
37	-1.228	-0.072	0.625	1.789	1.482	-0.184	-0.890	-0.384	-0.203	-0.221	1.622	0.083
38	0.997	0.083	-0.277	-2.635	-0.026	2.690	0.443	-0.787	-0.159	-0.511	-0.335	-1.034
39	-0.114	-1.726	-0.373	-0.522	-0.037	0.111	-0.983	-0.511	0.775	-1.341	0.314	-0.773
40	-1.372	0.205	1.110	0.554	0.888	-0.417	0.121	-2.237	0.799	-1.013	-0.853	1.427
41	-0.775	0.554	-0.301	0.716	-0.533	1.293	-1.161	1.279	0.659	-0.517	0.589	-0.417
42	-2.534	-0.252	0.887	-0.797	-1.816	-1.795	-0.801	0.863	1.191	0.059	0.571	0.257
43	1.692	0.896	0.315	-0.131	-2.202	0.056	-1.213	-0.684	-1.201	-0.904	-0.682	-1.023
44	-0.211	1.237	-0.182	0.889	-0.523	-1.664	-1.664	-0.245	-0.258	-0.015	-0.574	0.697
45	-1.081	-1.119	-0.506	-0.572	-0.670	0.279	-1.047	0.391	0.033	-0.168	0.139	-0.528
46	-1.513	-0.279	0.626	0.766	1.216	1.578	-1.294	0.028	-1.111	0.290	0.911	0.752
47	0.871	0.249	-0.322	-0.764	0.128	0.813	-0.832	1.432	-1.228	1.488	-0.944	0.235
48	1.306	0.480	-2.025	0.152	-0.686	0.589	-1.110	-0.415	0.172	-0.615	-0.222	1.098
49	0.897	-0.347	0.346	0.912	0.271	-1.556	-1.589	1.599	-0.901	-0.378	-1.110	-0.128
50	-1.550	1.228	1.843	-0.206	-2.379	-1.298	-0.283	-0.801	1.108	-1.388	0.309	-0.642
51	1.189	-0.096	-0.400	-1.056	0.591	0.731	1.444	0.390	0.986	-1.217	-1.319	-1.292
52	-0.120	0.036	-1.025	1.038	1.190	-0.631	1.521	2.309	0.103	-0.704	-1.029	-1.518
53	-1.410	0.536	-0.186	0.811	-0.048	0.808	1.620	-0.978	1.341	1.373	1.536	0.020
54	1.870	-0.117	-0.236	0.055	-1.126	-0.734	0.092	2.237	-0.543	-0.331	-0.888	-0.876
55	0.751	-1.423	-1.074	-0.167	-0.195	0.369	-0.382	2.176	-0.799	1.442	0.287	-0.870
56	-0.275	0.053	-0.013	-0.598	1.232	-1.634	0.529	-0.337	0.332	-0.264	0.385	0.062
57	0.849	0.721	0.572	-0.002	0.138	0.442	0.110	0.327	1.535	-0.207	1.094	-2.065
58	-0.893	0.003	0.291	0.613	-3.029	1.584	1.348	-0.174	-0.346	1.233	1.035	-0.008
59	-0.142	0.198	1.052	0.660	-0.038	-0.832	0.690	1.415	-0.644	0.578	-0.351	0.088
60	-0.519	-0.041	0.931	-0.073	-0.065	-0.628	-1.234	-0.390	-1.116	-2.325	-0.038	0.525

Appendix A7 OCEAN Scripts and Sample Netlist for Gyroscope Monte-Carlo Simulations

A7.1 OCEAN Script

```
simulator( "spectre" )
runname = "./sigma_0.05/run1";
design(strcat(runname "/netlist/netlist"))
  resultsDir( runname )
  desVar( "temper" 0 )
  desVar( "m2ma_x" 0.1u )
  desVar( "m1ma_x" 0.1u )
  desVar( "m2ma_y" 0.1u )
  desVar( "m1ma_y" 0.1u )
  desVar( "zframe" 0 )
  desVar( "zanchor" 0u )
  desVar( "zpm" 0 )
  desVar( "ifacbr" 1.0020 )
  desVar( "ifacbl" 0.9965 )
  desVar( "g_s_r" 1.7835u )
  desVar( "g_s_l" 1.8021u )
  desVar( "g_dr_r" 1.7842u )
  desVar( "g_dr_l" 1.8130u )
  desVar( "overetch" 0 )
  desVar( "mism" 0 )
  desVar( "fdrive" 8.4k )
  desVar( "fext" 500 )
  desVar( "offs" 0.0 )
  desVar( "w_i" 1.8000u )
  desVar( "w_o" 1.8000u )
  desVar( "ofacbr" 1.0110 )
  desVar( "_gpar3" 0 )
  desVar( "ax_sin" 0 )
  desVar( "ax_dc" 0 )
  desVar( "ay_sin" 0 )
  desVar( "ay_dc" 0 )
  desVar( "az_sin" 0 )
  desVar( "az_dc" 0 )
  desVar( "phi_dc" 0 )
  desVar( "phi_amp" 0 )
  desVar( "phix_sin" 0 )
  desVar( "phiy_sin" 0 )
  desVar( "fx_fr_sin" 0 )
```

```

desVar( "tphi_fr_sin" 0 )
desVar( "tphi_fr_ac" 0 )
desVar( "tphi_frame" 0 )
desVar( "tphi_disp" 0 )
desVar( "fz_pm_ac" 0 )
desVar( "fz_fr_ac" 0 )
desVar( "fy_pm_dc" 300.0000n )
desVar( "fy_pm_ac" 0 )
desVar( "fy_fr_ac" 0 )
desVar( "fx_pm_ac" 0 )
desVar( "fx_fr_ac" 0 )
desVar( "drive_sin" 2 )
desVar( "drive_dc" 0 )
desVar( "drive_sin_r" -2)

desVar("fx_fr_ac" 1u)
saveOption( 'save "selected")
analysis('ac ?start "1k" ?stop 100k ?dec 500 )
run()

selectResult( "ac-ac" )
x_fr_freq1 = xmax(vm("I0.x_frame_left_0"), 1)
fd = 50*round(x_fr_freq1/50)

analysis('tran ?maxstep 1u ?stop 60m ?errpreset "conservative")
desVar( "fx_fr_ac" 0u)
desVar( "fdrive" fd )
desVar( "ay_sin" 98 )
desVar( "fy_pm_dc" 300.0000n )
run()

fa = 500;
fdmfa = fd - fa;
fdpfa = fd + fa;
fdm2fa = fd - 2*fa;
fdp2fa = fd + 2*fa;
selectResult( "tran-tran" )
drive_wave = dft(v("I0.x_frame_left_0") 40m 60m 2048 "Rectangular" )
drive = value( drive_wave fd)
dft_wave = dft( (v("inp") - v("inn")) 40m 60m 2048 "Rectangular" )
lsb = value(dft_wave fdmfa)
carrier = value(dft_wave fd)
rsb = value(dft_wave fdpfa)
l2sb = value(dft_wave fdm2fa)
r2sb = value(dft_wave fdp2fa)
printf("RESULT: %s %6d %9.4g %4.2f %9.4g %4.2f %9.4g %4.2f %9.4g %4.2f
%9.4g %4.2f %9.4g %4.2f\n", runname fd mag(drive) phase(drive) mag(l2sb)
phase(l2sb) mag(lsb) phase(lsb) mag(carrier) phase(carrier) mag(rsb)
phase(rsb) mag(r2sb) phase(r2sb))

```

A7.2 Gyroscope 2D netlist

```
// Library name: bikram_gyro
// Cell name: gyro_mems
// View name: schematic_simple
// Inherited view list: spectres cdsSpice spice verilog behavioral
//functional hdl system verilogNetlist schematic cmos.sch cmos_sch
//veriloga ahd1 auLvs
subckt gyro_mems ax_ext ay_ext az_ext inn inp phi_disp
    phi_ext phix_ext phiy_ext
    v_act_bias v_act_l
    v_act_r vdc_neg_n vdc_neg_p vdc_pos_n vdc_pos_p x_disp y_disp
I83 (0 net0693) angle_source dc_value=0 ac_mag=0 ac_phase=0 offset=0
    amplitude=752f frequency=8.4k phase=0
I60 (0 phi_frame) torque_source dc_value=0 ac_mag=tphi_fr_ac
    ac_phase=0 offset=0 amplitude=tphi_fr_sin frequency=8.4k phase=0
I61 (0 phi_disp) torque_source dc_value=0 ac_mag=tphi_disp ac_phase=0
    offset=0 amplitude=0 frequency=0 phase=0
I65 (0 x_proof_left) force_source dc_value=0 ac_mag=fx_pm_ac
    ac_phase=0 offset=0 amplitude=0 frequency=0 phase=0
I67 (0 x_frame_left) force_source dc_value=0 ac_mag=fx_fr_ac
    ac_phase=0 offset=0 amplitude=fx_fr_sin frequency=8.4k phase=0
I72 (0 y_frame_bot) force_source dc_value=fy_fr_dc ac_mag=fy_fr_ac
ac_phase=0
    offset=fy_fr_dc amplitude=0 frequency=0 phase=0
I75 (net01619 net1202) force_source dc_value=fy_pm_dc
ac_mag=fy_pm_ac
    ac_phase=0 offset=fy_pm_dc amplitude=0 frequency=0 phase=0
I56 (net679 net695 net686 net1830 net693 net1832 net698 net682 net692
    net691 net696 net677 net683 net676 net678 net689 net681 net1831
    net675 net697 net700 net685 net699 net687 net684 net674 net694
    net680 net713 net702 net703 net704 net705 net706 net707 net708
    net709 net710 net711 net712) plate_1000 unitl=7.6u unitw=9.9u
    unitnum_x=24 unitnum_y=3 angle=0 joint_offset=7.05u
    percentage_holes=0.295853 fraction_m3=1 fraction_m2=0.00604
    fraction_m1=0.00604 fraction_poly=0 Xc=0 Yc=0 neighbor_x=1
    neighbor_y=0
I52 (phi_ext ax_ext ay_ext phi_frame net929 x_frame_left net738
net722
    net732 net731 net736 net717 net723 net716 net718 net729 net920
    y_frame_left net715 net737 net740 net725 net739 net727 net724
    net714 net734 net720 net753 v_act_bias net743 net744 net951
net746
    net747 net748 net749 net750 net751 net752) plate_1000
    unitl=44.9e-6 unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
    joint_offset=0u percentage_holes=0.284072 fraction_m3=1
    fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
    neighbor_x=0 neighbor_y=2
I40 (phi_ext ax_ext ay_ext phi_frame net773 net770 net778 net762
    net772 net771 net776 net757 net763 net756 net758 net769 net761
    net781 net755 net777 net780 net765 net779 net767 net764 net754
    net774 net760 net793 net782 net783 net784 net785 net786 net787
    net788 net789 net790 net791 net792) plate_1000 unitl=37.65u
    unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0 joint_offset=10.45u
```

percentage_holes=0.284072 fraction_m3=1 fraction_m2=0.15583
fraction_m1=0.16524 fraction_poly=0 Xc=0 Yc=0 neighbor_x=0
neighbor_y=2

I41 (phi_ext ax_ext ay_ext phi_frame net849 net810 net818 net802
net812 net811 x_frame_right net0865 net803 net1791 net798 net0720
net840 net821 net795 net817 net820 net805 y_frame_right net0875
net804 net1789 net814 net0708 net833 net822 net823 net824 net871
net826 net827 net828 net829 net830 net831 net832) plate_1000
unitl=44.9u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
joint_offset=22.45u percentage_holes=0.284072 fraction_m3=1
fraction_m2=0.15583 fraction_m1=0.16524 fraction_poly=0 Xc=0 Yc=0
neighbor_x=0 neighbor_y=2

I48 (phi_ext ax_ext ay_ext phi_frame net1249 net850 net858 net842
net852 net851 net856 net837 net843 net836 net838 net849 net1241
net861 net835 net857 net860 net845 net859 net847 net844 net834
net854 net840 net873 net862 net863 net864 net1271 net866 net867
net868 net869 net870 net871 net872) plate_1000 unitl=127.35u+offs
unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0 joint_offset=34.45u
percentage_holes=0.284072 fraction_m3=1 fraction_m2=0.15583
fraction_m1=0.16524 fraction_poly=0 Xc=0 Yc=0 neighbor_x=0
neighbor_y=2

I50 (phi_ext ax_ext ay_ext phi_frame net969 net890 net898 net882
net892 net891 net896 net877 net883 net876 net878 net889 net960
net901 net875 net897 net900 net885 net899 net887 net884 net874
net894 net880 net913 net902 net903 net904 net991 net906 net907
net908 net909 net910 net911 net912) plate_1000 unitl=9.4125u
unitw=7.6u unitnum_x=3 unitnum_y=4 angle=0 joint_offset=15.15u
percentage_holes=0.284072 fraction_m3=1 fraction_m2=0.18459
fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0 neighbor_x=0
neighbor_y=2

I44 (phi_ext ax_ext ay_ext phi_frame net1329 net930 net938 net922
net932 net931 net936 net917 net923 net1856 net918 net929 net1321
net941 net915 net937 net940 net925 net939 net927 net924 net1857
net934 net920 net953 net942 net1862 net944 net1351 net946 net947
net948 net949 net950 net951 net952) plate_1000 unitl=127.35u+offs
unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0 joint_offset=34.45u
percentage_holes=0.284072 fraction_m3=1 fraction_m2=0.18459
fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0 neighbor_x=0
neighbor_y=2

I51 (phi_ext ax_ext ay_ext phi_frame net1689 net970 net978 net962
net972 net971 net976 net957 net963 net1844 net958 net969 net1680
net981 net955 net977 net980 net965 net979 net967 net964 net1845
net974 net960 net993 net982 net1843 net984 net1711 net986 net987
net988 net989 net990 net991 net992) plate_1000 unitl=9.4125e-6
unitw=7.6u unitnum_x=3 unitnum_y=4.0 angle=0 joint_offset=10.45u
percentage_holes=0.284072 fraction_m3=1 fraction_m2=0.18459
fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0 neighbor_x=0
neighbor_y=2

I74 (phi_ext ax_ext ay_ext phi_frame net0720 net0717 net0725 net0709
net0719 net0718 net0723 net0704 net0710 net0703 net0705 net0716
net0708 net0728 net0702 net0724 net0727 net0712 net0726 net0714
net0711 net0701 net0721 net0707 net0740 net0729 net0730 net0731
net0732 net0733 net0734 net0735 net0736 net0737 net0738 net0739)
plate_1000 unitl=82.45u-offs unitw=7.6u unitnum_x=3 unitnum_y=1

```

angle=0 joint_offset=22.45u percentage_holes=0.284072
fraction_m3=1 fraction_m2=0.15583 fraction_m1=0.16524
fraction_poly=0 Xc=0 Yc=0 neighbor_x=0 neighbor_y=2
I14 (phi_ext ax_ext ay_ext phi_frame net1013 net1616 net1018 net1003
net1012 net1011 net1016 net998 net1004 net997 net999 net1253
net1002 net1619 net996 net1017 net1020 net1006 net1019 net1008
net1005 net995 net1014 net1242 net1033 net1628 net1023 net1024
net1025 net1026 net1027 net1028 net1029 net1030 net1265 net1032)
plate_1000 unitl=22.8u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
joint_offset=0u percentage_holes=0.284072 fraction_m3=1
fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
neighbor_x=1 neighbor_y=1
I13 (phi_ext ax_ext ay_ext phi_frame net1053 net1050 net1058 net1043
net1052 net1051 net1056 net1038 net1044 net1037 net1039 net1049
net1042 net1061 net1036 net1057 net1060 net1046 net1059 net1048
net1045 net1035 net1054 net1041 net1073 net1062 net1063 net1064
net1065 net1066 net1067 net1068 net1069 net1070 net1071 net1072)
plate_1000 unitl=22.8u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
joint_offset=0u percentage_holes=0.284072 fraction_m3=1
fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
neighbor_x=1 neighbor_y=1
I15 (phi_ext ax_ext ay_ext phi_frame net1093 net1656 net1098 net1083
net1092 net1091 net1096 net1078 net1084 net1077 net1079 net1089
net1082 net1659 net1076 net1097 net1100 net1086 net1099 net1088
net1085 net1075 net1094 net1081 net1113 net1668 net1103 net1104
net1105 net1106 net1107 net1108 net1109 net1110 net1111 net1112)
plate_1000 unitl=22.8u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
joint_offset=0u percentage_holes=0.284072 fraction_m3=1
fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
neighbor_x=1 neighbor_y=1
I16 (phi_ext ax_ext ay_ext phi_frame net1369 net1130 net1138 net1123
net1132 net1131 net1136 net1118 net1124 net1117 net1119 net1129
net1361 net1141 net1116 net1137 net1140 net1126 net1139 net1128
net1125 net1115 net1134 net1121 net1153 net1142 net1143 net1144
net1391 net1146 net1147 net1148 net1149 net1150 net1151 net1152)
plate_1000 unitl=22.8u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
joint_offset=0u percentage_holes=0.284072 fraction_m3=1
fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
neighbor_x=1 neighbor_y=1
I6 (phi_ext ax_ext ay_ext phi_disp net1569 net1170 net1178 net1163
net1172 net2204 net1176 net1158 net1164 net1157 net1159 net1169
net1561 net1181 net1156 net1177 net1180 net2205 net1179 net1168
net1165 net1155 net1174 net1161 net1193 net1182 net1183 net1184
net1591 net1186 net1187 net1188 net1189 net1190 net1191 net1192)
plate_1000 unitl=9.55u unitw=9.3u unitnum_x=12 unitnum_y=1
angle=0
joint_offset=0.9u percentage_holes=0.208716 fraction_m3=1
fraction_m2=0.59043 fraction_m1=0.53222 fraction_poly=0 Xc=0 Yc=0
neighbor_x=0 neighbor_y=1
I5 (phi_ext ax_ext ay_ext phi_disp net1213 net1210 net1218 net1203
net1212 net1211 net1216 net1198 net1204 net2084 net1199 net1209
net1202 net1221 net1196 net1217 net1220 net1206 net1219 net1208
net1205 net2085 net1214 net1201 net1233 net1222 net1223 net1224
net1225 net1226 net1227 net1228 net1229 net1230 net1231 net1232)

```

```

    plate_1000 unitl=9.55u unitw=9.3u unitnum_x=12 unitnum_y=1
angle=0
    joint_offset=0.9u percentage_holes=0.208716 fraction_m3=1
    fraction_m2=0.59043 fraction_m1=0.53222 fraction_poly=0 Xc=0 Yc=0
    neighbor_x=0 neighbor_y=1
I967 (phi_ext ax_ext ay_ext phi_frame net1253 net1250 net1258 net1243
    net1252 net1251 net1256 net1238 net1244 net1237 net1239 net1249
    net1242 net1261 net1236 net1257 net1260 net1246 net1259 net1248
    net1245 net1235 net1254 net1241 net1273 net1262 net1263 net1264
    net1265 net1266 net1267 net1268 net1269 net1270 net1271 net1272)
    plate_1000 unitl=55.4u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
    joint_offset=15.15u percentage_holes=0.284072 fraction_m3=1
    fraction_m2=0.15583 fraction_m1=0.16524 fraction_poly=0 Xc=0 Yc=0
    neighbor_x=0 neighbor_y=2
I968 (phi_ext ax_ext ay_ext phi_frame net1729 net1290 net1298 net1283
    net1292 net1291 net1296 net1278 net1284 net1277 net1279 net1093
    net1720 net1301 net1276 net1297 net1300 net1286 net1299 net1288
    net1285 net1275 net1294 net1082 net1313 net1302 net1303 net1304
    net1751 net1306 net1307 net1308 net1309 net1310 net1105 net1312)
    plate_1000 unitl=25u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
    joint_offset=0u percentage_holes=0.284072 fraction_m3=1
    fraction_m2=0.15583 fraction_m1=0.16524 fraction_poly=0 Xc=0 Yc=0
    neighbor_x=0 neighbor_y=2
I971 (phi_ext ax_ext ay_ext phi_frame net1049 net1330 net1338 net1323
    net1332 net1331 net1336 net1318 net1324 net1317 net1319 net1329
    net1041 net1341 net1316 net1337 net1340 net1326 net1339 net1328
    net1325 net1315 net1334 net1321 net1353 net1342 net1343 net1344
    net1071 net1346 net1347 net1348 net1349 net1350 net1351 net1352)
    plate_1000 unitl=55.4u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
    joint_offset=15.15u percentage_holes=0.284072 fraction_m3=1
    fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
    neighbor_x=0 neighbor_y=2
I972 (phi_ext ax_ext ay_ext phi_frame net889 net1370 net1378 net1363
    net1372 net1371 net1376 net1358 net1364 net1357 net1359 net1369
    net880 net1381 net1356 net1377 net1380 net1366 net1379 net1368
    net1365 net1355 net1374 net1361 net1393 net1382 net1383 net1384
    net911 net1386 net1387 net1388 net1389 net1390 net1391 net1392)
    plate_1000 unitl=25u unitw=7.6u unitnum_x=3 unitnum_y=1 angle=0
    joint_offset=0u percentage_holes=0.284072 fraction_m3=1
    fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
    neighbor_x=0 neighbor_y=2
I973 (phi_ext ax_ext ay_ext phi_disp x_proof_top net1410 net1418
    net1403 net1412 net1411 net1416 net1398 net1404 net1397 net1399
    net1409 y_proof_top net1421 net1396 net1417 net1420 net1406
    net1419 net1408 net1405 net1395 net1414 net1401 net1433 net1422
    net1423 net1424 net1471 net1426 net1427 net1428 net1429 net1430 0
    net1432) plate_1000 unitl=5.7u-offs unitw=9.3u unitnum_x=12
    unitnum_y=1 angle=0 joint_offset=0u percentage_holes=0.208716
    fraction_m3=1 fraction_m2=0.59043 fraction_m1=0.53222
    fraction_poly=0 Xc=0 Yc=0 neighbor_x=0 neighbor_y=2
I974 (phi_ext ax_ext ay_ext phi_disp x_proof_bot x_proof_left net1458
    net1443 x_disp net1451 x_proof_right net1438 net1444 net1437
    net1439 x_proof_top y_proof_bot y_proof_left net1436 net1457
    y_disp net1446 y_proof_right net1448 net1445 net1435 net1454

```

```

y_proof_top net1473 net1462 net1463 net1464 net1511 net1466
net1467 net1468 net1469 net1470 net1471 net1472) plate_1000
unitl=284.7u unitw=9.3u unitnum_x=12 unitnum_y=1 angle=0
joint_offset=0u percentage_holes=0.208716 fraction_m3=1
fraction_m2=0.59043 fraction_m1=0.53222 fraction_poly=0 Xc=0 Yc=0
neighbor_x=0 neighbor_y=2
I975 (phi_ext ax_ext ay_ext phi_disp net1529 net1490 net1498 net1483
net1492 net1491 net1496 net1478 net1484 net1477 net1479
x_proof_bot net1521 net1501 net1476 net1497 net1500 net1486
net1499 net1488 net1485 net1475 net1494 y_proof_bot net1513
net1502 net1503 net1504 net1551 net1506 net1507 net1508 net1509
net1510 net1511 net1512) plate_1000 unitl=5.7u+offs unitw=9.3u
unitnum_x=12 unitnum_y=1 angle=0 joint_offset=0u
percentage_holes=0.208716 fraction_m3=1 fraction_m2=0.59043
fraction_m1=0.53222 fraction_poly=0 Xc=0 Yc=0 neighbor_x=0
neighbor_y=2
I976 (phi_ext ax_ext ay_ext phi_disp net1209 net1530 net1538 net1523
net1532 net1531 net1536 net1518 net1524 net1517 net1519 net1529
net1201 net1541 net1516 net1537 net1540 net1526 net1539 net1528
net1525 net1515 net1534 net1521 net1553 net1542 net1543 net1544
net1231 net1546 net1547 net1548 net1549 net1550 net1551 net1552)
plate_1000 unitl=9.575u unitw=9.3u unitnum_x=12 unitnum_y=3
angle=0 joint_offset=16.95u percentage_holes=0.208716
fraction_m3=1 fraction_m2=0.59043 fraction_m1=0.53222
fraction_poly=0 Xc=0 Yc=0 neighbor_x=0 neighbor_y=2
I977 (phi_ext ax_ext ay_ext phi_disp net1409 net1570 net1578 net1563
net1572 net1571 net1576 net1558 net1564 net1557 net1559 net1569
net1401 net1581 net1556 net1577 net1580 net1566 net1579 net1568
net1565 net1555 net1574 net1561 net1593 net1582 net1583 net1584 0
net1586 net1587 net1588 net1589 net1590 net1591 net1592)
plate_1000 unitl=9.575u unitw=9.3u unitnum_x=12 unitnum_y=3
angle=0 joint_offset=16.95u percentage_holes=0.208716
fraction_m3=1 fraction_m2=0.59043 fraction_m1=0.53222
fraction_poly=0 Xc=0 Yc=0 neighbor_x=0 neighbor_y=2
I982 (phi_ext ax_ext ay_ext phi_frame x_frame_bot net1056 net1618
net1603 net1959 net1611 net1616 net1598 net1604 net1597 net1599
net01627 y_frame_bot net1059 net1596 net1617 net1961 net1606
net1619 net1608 net1605 net1595 net1614 net01619 net1633 net1068
net1623 net1624 net1625 net1626 net1627 net1628 net1629 net1630
net1631 net1968) plate_1000 unitl=7.6u unitw=9.9u unitnum_x=24
unitnum_y=3 angle=0 joint_offset=7.05u percentage_holes=0.295853
fraction_m3=1 fraction_m2=0.14949 fraction_m1=0 fraction_poly=0
Xc=0 Yc=0 neighbor_x=2 neighbor_y=0
I1028 (phi_ext ax_ext ay_ext phi_frame net01671 net1136 net1658
net1643 net1652 net1651 net1656 net1638 net1644 net1637 net2324
x_frame_top net01660 net1139 net1636 net1657 net1660 net1646
net1659 net1648 net1645 net1635 net2325 y_frame_top net1673
net1148 net1663 net2330 net1665 net1666 net1667 net1668 net1669
net1670 net1671 net1672) plate_1000 unitl=7.6u unitw=9.9u
unitnum_x=24 unitnum_y=3 angle=0 joint_offset=7.05u
percentage_holes=0.295853 fraction_m3=1 fraction_m2=0.14949
fraction_m1=0 fraction_poly=0 Xc=0 Yc=0 neighbor_x=2 neighbor_y=0
I53 (phi_ext ax_ext ay_ext phi_frame net729 net1690 net1698 net1682
net1692 net1691 net1696 net1677 net1683 net1676 net1678 net1689

```

```

net720 net1701 net1675 net1697 net1700 net1685 net1699 net1687
net1684 net1674 net1694 net1680 net1713 net1702 net1703 net1704
net751 net1706 net1707 net1708 net1709 net1710 net1711 net1712)
plate_1000 unitl=82.45u-offs unitw=7.6u unitnum_x=3 unitnum_y=1
angle=0 joint_offset=0u percentage_holes=0.284072 fraction_m3=1
fraction_m2=0.18459 fraction_m1=0.13085 fraction_poly=0 Xc=0 Yc=0
neighbor_x=0 neighbor_y=2
I147 (phi_ext ax_ext ay_ext phi_frame net769 net1730 net1738 net1722
net1732 net1731 net1736 net1717 net1723 net1716 net1718 net1729
net760 net1741 net1715 net1737 net1740 net1725 net1739 net1727
net1724 net1714 net1734 net1720 net1753 net1742 net1743 net1744
net791 net1746 net1747 net1748 net1749 net1750 net1751 net1752)
plate_1000 unitl=9.4125u unitw=7.6u unitnum_x=3 unitnum_y=4
angle=0 joint_offset=15.15u percentage_holes=0.284072
fraction_m3=1 fraction_m2=0.15583 fraction_m1=0.16524
fraction_poly=0 Xc=0 Yc=0 neighbor_x=0 neighbor_y=2
I1047 (phi_ext ax_ext ay_ext phi_frame net1762 v_act_bias v_act_bias
v_act_bias v_act_bias v_act_r v_act_r v_act_r v_act_r
x_frame_right net1757 y_frame_right net1764) comb_1111_2
finger_width=2.7u finger_length=11.4u overlap=3.3u gap=g_dr_r
fingers=23 angle=0 Xc=0 Yc=0 wing_length_a=0u wing_length_b=0u
truss_width_a=0u truss_width_b=0u
I1046 (phi_ext ax_ext ay_ext net1836 phi_frame v_act_l v_act_l
v_act_l
v_act_l v_act_bias v_act_bias v_act_bias v_act_bias net1838
x_frame_left net1837 y_frame_left) comb_1111_1 finger_width=2.7u
finger_length=11.4u overlap=3.3u gap=g_dr_l fingers=23 angle=0
Xc=0 Yc=0 wing_length_a=0u wing_length_b=0u truss_width_a=0u
truss_width_b=0u
I1045 (phi_ext ax_ext ay_ext phi_disp phi_frame inn inn inn inn
vdc_pos_n vdc_pos_n vdc_pos_n vdc_pos_n vdc_neg_n vdc_neg_n
vdc_neg_n vdc_neg_n x_proof_right net810 y_proof_right net821)
comb_diff_1111_2 finger_width_a=5.7u finger_width_b=3.9u
finger_length=61.5u overlap=60u gap=g_s_r fingers=21 Xc=0 Yc=0
angle=0 wing_length_a=0u wing_length_b=0u truss_width_a=0u
truss_width_b=0u
I1044 (phi_ext ax_ext ay_ext phi_frame phi_disp inp inp inp inp
vdc_neg_p vdc_neg_p vdc_neg_p vdc_neg_p vdc_pos_p vdc_pos_p
vdc_pos_p vdc_pos_p net736 x_proof_left net739 y_proof_left)
comb_diff_1111_1 finger_width_a=3.9u finger_width_b=5.7u
finger_length=61.5u overlap=60u gap=g_s_l fingers=21 Xc=0 Yc=0
angle=0 wing_length_a=0u wing_length_b=0u truss_width_a=0u
truss_width_b=0u
I157 (net1830 net1832 net1831) anchor
I978 (net1762 net1757 net1764) anchor l=32.4u w=27u
I1032 (net1836 net1838 net1837) anchor l=32.4u w=27u
I10 (net2166 phi_frame net2163 net1844 net2165 net1845 nt2172 net1843
net2164 net1843 net2171 net1843) beam_1110 l=6.0u w=5.1u angle=0
I7 (net1930 phi_frame net1928 net1856 net1929 net1857 net1934 net1862
net1933 0 net1927 0) beam_1110 l=6.0u w=5.1u angle=0
I9 (phi_frame net1870 net757 net1868 net767 net1869 net787 net1874
net787 net1873 net787 net1867) beam_1110 l=6.0u w=5.1u angle=0
I8 (phi_frame net1882 net837 net1880 net847 net1881 net867 net1886
net867 net1885 net867 net1879) beam_1110 l=6.0u w=5.1u angle=0

```

I981 (net1914 net1894 net1911 net1892 net1913 net1893 net1920 net1898
 net1912 net1897 net1919 net1891) beam_1110 l=5.1u+1.8u w=5.1u
 angle=0

I983 (net1938 net1906 net1935 net1904 net1937 net1905 net1944 net1910
 net1936 net1909 net1943 net1903) beam_1110 l=5.1u+1.8u w=5.1u
 angle=0

I984 (net1914 net2110 net1911 net2108 net1913 net2109 net1920 net2114
 net1912 net2113 net1919 net2107) beam_1110 l=105u+5.1u
 w=ofacbl*1.8u-overetch angle=90

I985 (net1894 net1930 net1892 net1928 net1893 net1929 net1898 net1934
 net1897 net1933 net1891 net1927) beam_1110 l=105u+5.1u
 w=ofacbl*1.8u-overetch angle=90

I986 (net1938 net1882 net1935 net1880 net1937 net1881 net1944 net1886
 net1936 net1885 net1943 net1879) beam_1110 l=105u+5.1u
 w=ofacbr*1.8u-overetch angle=90

I987 (net1906 net1954 net1904 net1952 net1905 net1953 net1910 net1958
 net1909 net1957 net1903 net1951) beam_1110 l=105u+5.1u
 w=ofacbr*1.8u-overetch angle=90

I988 (phi_frame net1966 net1959 net1964 net1961 net1965 net1968
 net1970 0 net1969 0 net1963) beam_1110 l=5.1u+0.9u w=5.1u
 angle=90

I989 (phi_frame net2002 net1618 net2000 net1596 net2001 net1630
 net2006 0 net2005 0 net1999) beam_1110 l=5.1u+0.9u w=5.1u
 angle=90

I990 (net1966 net1990 net1964 net1988 net1965 net1989 net1970 net1994
 net1969 net1993 net1963 net1987) beam_1110 l=102u w=1.8u*ifacbl
 angle=0

I991 (net2022 net2002 net2019 net2000 net2021 net2001 net2028 net2006
 net2020 net2005 net2027 net1999) beam_1110 l=102u w=1.8u*ifacbr
 angle=0

I992 (net1990 net2038 net1988 net2036 net1989 net2037 net1994 net2042
 net1993 net2041 net1987 net2035) beam_1110 l=5.1u+1.8u w=5.1u
 angle=90

I993 (net2022 net2026 net2019 net2024 net2021 net2025 net2028 net2030
 net2020 net2029 net2027 net2023) beam_1110 l=5.1u+1.8u w=5.1u
 angle=90

I994 (net2058 net2038 net2055 net2036 net2057 net2037 net2064 net2042
 net2056 net2041 net2063 net2035) beam_1110 l=102u w=1.8u*ifacbl
 angle=0

I995 (net2026 net2050 net2024 net2048 net2025 net2049 net2030 net2054
 net2029 net2053 net2023 net2047) beam_1110 l=102u w=1.8u*ifacbr
 angle=0

I996 (net2058 net2062 net2055 net2060 net2057 net2061 net2064 net2066
 net2056 net2065 net2063 net2059) beam_1110 l=5.1u+1.8u w=5.1u
 angle=90

I997 (net2050 net2098 net2048 net2096 net2049 net2097 net2054 net2102
 net2053 net2101 net2047 net2095) beam_1110 l=5.1u+1.8u w=5.1u
 angle=90

I998 (net2062 phi_disp net2060 net2084 net2061 net2085 net2066
 net2090
 net2065 net2089 net2059 net2083) beam_1110 l=53.4u+2.55u w=1.8u
 angle=0

I999 (phi_disp net2098 net1198 net2096 net1208 net2097 net2100
 net2102

```

net2092 net2101 net2099 net2095) beam_1110 l=53.4u+2.55u w=1.8u
angle=0
I1002 (net1836 net2110 net1838 net2108 net1837 net2109 net2112
net2114
net2104 net2113 net2111 net2107) beam_1110 l=5.1u+0.9u w=5.1u
angle=0
I1005 (net1954 net1762 net1952 net1757 net1953 net1764 net1958
net2126
net1957 net2125 net1951 net2119) beam_1110 l=5.1u+0.9u w=5.1u
angle=0
I1006 (net1836 net2134 net1838 net2132 net1837 net2133 net2136
net2138
net2128 net2137 net2135 net2131) beam_1110 l=5.1u+0.9u w=5.1u
angle=0
I1009 (net2190 net1762 net2187 net1757 net2189 net1764 net2196
net2150
net2188 net2149 net2195 net2143) beam_1110 l=5.1u+0.9u w=5.1u
angle=0
I1010 (net2134 net2158 net2132 net2156 net2133 net2157 net2138
net2162
net2137 net2161 net2131 net2155) beam_1110 l=105u+5.1u
w=ofactl*1.8u-overetch angle=90
I1011 (net2166 net2350 net2163 net2348 net2165 net2349 net2172
net2354
net2164 net2353 net2171 net2347) beam_1110 l=105u+5.1u
w=ofactl*1.8u-overetch angle=90
I1012 (net1870 net2182 net1868 net2180 net1869 net2181 net1874
net2186
net1873 net2185 net1867 net2179) beam_1110 l=105u+5.1u
w=ofactr*1.8u-overetch angle=90
I1013 (net2190 net2362 net2187 net2360 net2189 net2361 net2196
net2366
net2188 net2365 net2195 net2359) beam_1110 l=105u+5.1u
w=ofactr*1.8u-overetch angle=90
I1016 (net2226 phi_disp net2223 net2204 net2225 net2205 net2232 0
net2224 0 net2231 0) beam_1110 l=53.4u+2.55u w=1.8u angle=0
I1017 (phi_disp net2218 net1163 net2216 net1177 net2217 0 net2222 0
net2221 0 net2215) beam_1110 l=53.4u+2.55u w=1.8u angle=0
I1018 (net2226 net2230 net2223 net2228 net2225 net2229 net2232
net2234
net2224 net2233 net2231 net2227) beam_1110 l=5.1u+1.8u w=5.1u
angle=90
I1019 (net2218 net2266 net2216 net2264 net2217 net2265 net2222
net2270
net2221 net2269 net2215 net2263) beam_1110 l=5.1u+1.8u w=5.1u
angle=90
I1020 (net2230 net2254 net2228 net2252 net2229 net2253 net2234
net2258
net2233 net2257 net2227 net2251) beam_1110 l=102u w=1.8u*ifactl
angle=0
I1021 (net2286 net2266 net2283 net2264 net2285 net2265 net2292
net2270
net2284 net2269 net2291 net2263) beam_1110 l=102u w=1.8u*ifactr
angle=0

```

```

I1022 (net2254 net2302 net2252 net2300 net2253 net2301 net2258
net2306
    net2257 net2305 net2251 net2299) beam_1110 l=5.1u+1.8u w=5.1u
    angle=90
I1023 (net2286 net2290 net2283 net2288 net2285 net2289 net2292
net2294
    net2284 net2293 net2291 net2287) beam_1110 l=5.1u+1.8u w=5.1u
    angle=90
I1024 (net2322 net2302 net2319 net2300 net2321 net2301 net2328
net2306
    net2320 net2305 net2327 net2299) beam_1110 l=102u w=1.8u*ifactl
    angle=0
I1025 (net2290 net2314 net2288 net2312 net2289 net2313 net2294
net2318
    net2293 net2317 net2287 net2311) beam_1110 l=102u w=1.8u*ifactr
    angle=0
I1026 (net2322 phi_frame net2319 net2324 net2321 net2325 net2328
net2330 net2320 net2329 net2327 net2323) beam_1110 l=5.1u+0.9u
w=5.1u angle=90
I1027 (net2314 phi_frame net2312 net1644 net2313 net1645 net2318
net1666 net2317 net2341 net2311 net2335) beam_1110 l=5.1u+0.9u
w=5.1u angle=90
I1029 (net2158 net2350 net2156 net2348 net2157 net2349 net2162
net2354
    net2161 net2353 net2155 net2347) beam_1110 l=5.1u+1.8u w=5.1u
    angle=0
I1030 (net2182 net2362 net2180 net2360 net2181 net2361 net2186
net2366
    net2185 net2365 net2179 net2359) beam_1110 l=5.1u+1.8u w=5.1u
    angle=0
ends gyro_mems
// End of subcircuit definition.

// Library name: bikram_gyro
// Cell name: gyro_sim_no_ckt
// View name: schematic
// Inherited view list: spectreS cdsSpice spice verilog behavioral
//functional hdl system verilogNetlist schematic cmos.sch cmos_sch
//veriloga ahdl auLvs
I6 (0 ax_ext) position_source dc_value=ax_dc ac_mag=0 ac_phase=0
    offset=ax_dc amplitude=ax_sin frequency=100 phase=0
I7 (0 ay_ext) position_source dc_value=ay_dc ac_mag=0 ac_phase=0
    offset=ay_dc amplitude=ay_sin frequency=fext phase=0
Az (0 az_ext) position_source dc_value=az_dc ac_mag=0 ac_phase=0
    offset=az_dc amplitude=az_sin frequency=fext phase=0
C4 (inp dr_bias) capacitor c=mism*23.4501e-15
C5 (inn dr_bias) capacitor c=mism*1.09296e-16
C0 (inp 0) capacitor c=500.0f
C1 (inn 0) capacitor c=500.0f
I17 (0 phi_ext) angle_source dc_value=phi_dc ac_mag=0 ac_phase=0 off-
set=0
    amplitude=phi_amp frequency=fext phase=0
Phix (0 phix_ext) angle_source dc_value=0 ac_mag=0 ac_phase=0 offset=0
    amplitude=phix_sin frequency=fext phase=0

```

```

Phiy (0 phiy_ext) angle_source dc_value=0 ac_mag=0 ac_phase=0 offset=0
    amplitude=phiy_sin frequency=fext phase=0
I0 (ax_ext ay_ext az_ext inn inp phi_disp phi_ext phix_ext phiy_ext
    dr_bias drive_l drive_r net90
    net90 net92 net92 x_disp y_disp) gyro_mems
V8 (drive_r 0) vsource dc=-drive_dc mag=10 phase=0 type=sine delay=0
    sinedc=0 ampl=drive_sin_r freq=fdrive
V0 (drive_l 0) vsource dc=drive_dc mag=10 phase=0 type=sine sinedc=0
    ampl=drive_sin freq=fdrive
V7 (net086 0) vsource dc=0 mag=0 phase=0 type=dc
V6 (net088 0) vsource dc=0 mag=0 phase=0 type=dc
V2 (net90 0) vsource dc=-2.5 mag=0 phase=0 type=dc
V3 (net92 0) vsource dc=2.5 mag=0 phase=0 type=dc
V1 (dr_bias 0) vsource dc=18 mag=0 phase=0 type=dc

save I0.phi_frame ay_ext x_disp y_disp phi_disp I0.x_frame_left
    I0.y_frame_left I0.x_frame_bot I0.y_frame_bot I0.x_frame_right
    I0.y_frame_right I0.x_frame_top I0.y_frame_top I0.x_proof_left
    I0.y_proof_left I0.x_proof_bot I0.y_proof_bot I0.x_proof_right
    I0.y_proof_right I0.x_proof_top I0.y_proof_top I0:4 I0:3 inp inn
ax_ext

```