Approaches to Synthesis of a CMOS Accelerometer

Vishal Gupta

A report submitted to the graduate school in partial fulfillment
of the requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Advisor: Dr. Tamal Mukherjee
Second Reader: Prof. Rob A. Rutenbar

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania
August, 2000

Abstract

A synthesis framework for automated design of CMOS MEMS accelerometers from high-level behavioral specifications is presented. Two approaches to synthesis are discussed. The equation-based synthesis approach requires lumped parameter model equations to characterize the behavior of the device being synthesized. With these equations and constraints on the device behavior, the synthesis problem is formulated into a non-linear constrained optimization problem, where objective functions are also used to guide the search for a solution in the design space. The second approach, simulation-based synthesis, uses a nodal simulator (NODAS) to evaluate the behavior of a proposed design during an annealing-based search for an optimal design. This approach formulates the synthesis problem into an unconstrained optimization problem. For both of these approaches, the algorithms, the implementations, the results and the relative pros and cons are discussed in detail. A comparison of synthesized designs with manual designs suggest significant optimization possibilities for MEMS design by using CAD tools. Experimental measurements are also presented in order to verify the synthesis results.

Acknowledgments

I wish to express my sincere gratitude and indebtedness to my advisor, Dr. Tamal Mukherjee without whose active interest and constructive criticism this work would not have been possible. I would like to thank Prof. Rob Rutenbar for his advice and encouragement during this research, and for his patience to read my thesis.

My colleagues at Carnegie Mellon have contributed a lot to the growth of this research during numerous discussions. Specifically, I want to thank Sitaraman Iyer, Huikai Xie, Qi Jing, Hao Luo, Michael Krasnicki, Hasnain Lakdawala, Xu Zhu, Bikram Baidya and Satish Vemuri for their support. Finally, I wish to thank my parents for their love and encouragement.

This research effort was sponsored by the Defence Advanced Research Projects Agency (DARPA) Composite CAD program, under grant number F30602-96-2-0304.

Table of Contents

I. Introduction

Traditionally, MEMS have been designed in a bottom-up manner, from sensor design to system design, evidently due to the inherent complexity of MEMS design. However, the advent of VLSI MEMS poses the need for a suitable CAD flow to support top-down design of MEMS. There is increasing interest towards integrated microsystems, like an inertial measurement unit (IMU), which use an array of similar topology devices (like accelerometers and gyroscopes) with different behavioral specifications for enhanced performance. Cell-level synthesis tools in a CAD flow automate device design from high-level behavioral specifications. This provides higher levels of abstraction, leading to a much more structured and faster approach to MEMS design.

This work focuses on investigating alternative approaches to synthesis for automated design of CMOS accelerometers. An IMU is an example of an integrated system that can potentially make use of such synthesis tools. Two approaches to synthesis, *equation-based synthesis* and *simulation-based synthesis*, are presented here. Both these approaches to MEMS synthesis derive from engineering design synthesis techniques [1][2][3][4]. Equation-based synthesis of polysilicon resonators [5] and accelerometers [6] has previously been demonstrated, while simulation-based synthesis approach has previously been used to synthesize analog circuits in [8]. This work extends the MEMS synthesis paradigm to a more intricate technology (CMOS surface micromachining [7]), as well as for simulation-based synthesis [8].

The first approach to synthesis, *equation-based synthesis* is highly knowledge intensive, and requires codification of designer expertise into analytical lumped parameter behavioral equations in order to characterize a particular device design. The second approach, *simulation-based synthesis*, is more flexible than equation-based synthesis. It uses a nodal simulator (NODAS [9]), and a suite of evaluation scripts built on top of simulation runs to characterize a device design, and hence is much easier to use. Both of these approaches are extensible to other types of cells for MEMS design, but simulation-based synthesis is more accurate and easy to use as compared to equation-based synthesis. On the other hand, it takes much longer computing time than equation-based synthesis.

Results from experiments and simulations demonstrate that the use of a synthesis tool for cell-level design provides significant optimization over manual design. It also allows us to explore comprehensive system design trade-offs, and lead to system design optimization in addition to device optimization.

Thesis Organization

Chapter 2 of this thesis discusses the formulation of CMOS accelerometer design problem into a synthesis problem. It presents the issues involved in the design of a CMOS accelerometer from a synthesis perspective. The generic details of the synthesis strategy are also presented.

In Chapter 3, the details of *equation-based synthesis* are presented. The issues involved in modeling the accelerometer behavior in terms of lumped parameter equations are discussed. The synthesis results from a prototype tool are compared with a manual design, and finally, some insights provided by trade-off analysis using synthesis are presented in this chapter.

Chapter 4 discusses the framework for *simulation-based synthesis* of CMOS accelerometers. The evaluation-jigs for the nodal simulation of an accelerometer are also discussed. Finally, the results from a prototype simulation-based synthesis tool are presented.

In Chapter 5, a comparison of device behavior evaluation using the lumped parameter models (used in the *equation-based synthesis* approach) and that using NODAS is presented in order to validate the models. Accelerometers synthesized using *equation-based synthesis* approach were fabricated. Measurements from these chips, and the conclusions to be drawn from these measurements are also presented.

Chapter 6 presents the conclusions to be drawn from this work, and possible directions for further research.

II. CMOS Accelerometer Synthesis

This chapter discusses the formulation of CMOS MEMS accelerometer design into a synthesis problem. At this point of time, most of this formulation has been done manually but, in future, in an advanced synthesis framework, one would like to automate the process of synthesis formulation as well.

II.1 CMOS Accelerometer Overview

The CMOS micromachining technology [7] uses a 0.5µm three metal layer CMOS process. After the normal CMOS foundry fabrication, an anisotropic etch is used to etch away the unmasked oxide layers to define the sidewalls of the microstructure. Finally, an isotropic silicon etch is performed to release the microstructures, as shown in *Fig. II.1*. The top metal layer acts as a mask for both the post-processing steps. This technology allows multiple signal routes within the same mechanical structure. Also, the microstructures are approximately 20µm above the substrate leading to designs with low parasitic capacitance.

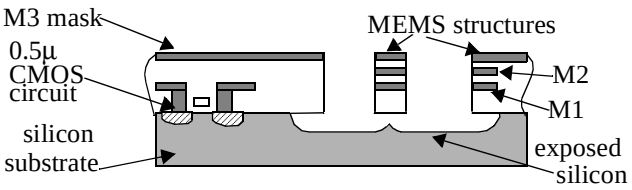


Fig. II.1: A CMOS-MEMS chip cross-section

The topology of the accelerometer used here is that of a single-axis, common-centroid, fully-differential, capacitive-sensing lateral accelerometer [10][11]. The schematic is shown in *Fig. II.2*. The proof mass is suspended using four serpentine springs attached to its corners. A limit stop is also used (not shown in the figure) to stop the movement of the spring under large accelerations, and hence, prevent the comb-fingers

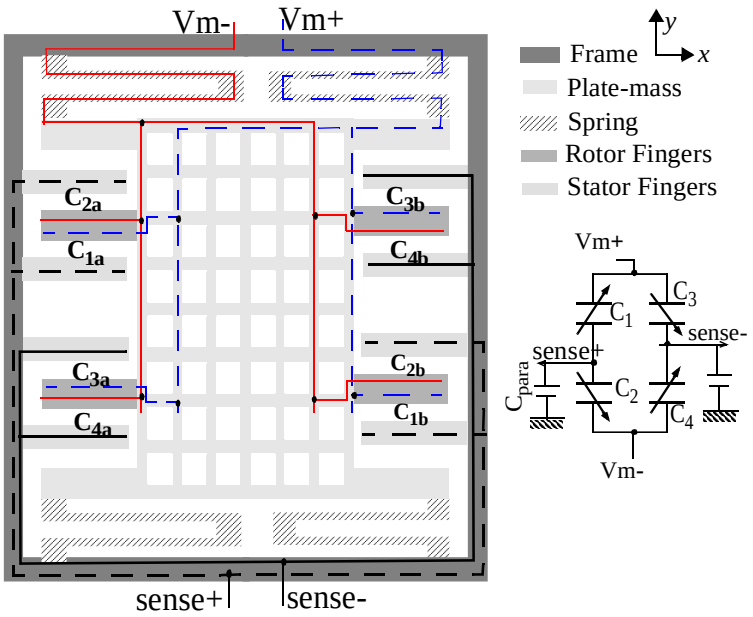


Fig. II.2: Accelerometer topology schematic

from crashing into each other. Interdigitated comb-drives are used for differential capacitive sensing (shown in the schematic), as well as for force feedback balancing (not shown). Each rotor finger consists of two electrical nodes (using multiple routes through the plate-mass), one each for the two capacitors on the capacitive half bridge; and the sense nodes are located on the stator fingers. This is used to create a common-centroid configuration, which is not possible in polysilicon MEMS [12]. In order to counter out-of-plane curl mismatch between the comb-fingers, the stator fingers are attached to a peripheral frame rather than being anchored to the substrate. The transducer area is also limited to *x-size* X *y-size*.

The synthesis methodology discussed here uses the above-mentioned fixed topology of the sensor, and optimally sizes the constituent elements of the sensor to meet user specifications like sensitivity, noise, *etc*. The optimization of sensor topology is beyond the scope of this work.

II.2 Synthesis Approach

Synthesis is usually an iterative process of search and evaluation. The search algorithm provides a candidate solution to meet the user specifications, and the evaluation engine then provides a measure of goodness (cost) of that solution. The aim of this iterative exercise is to find a minimal cost (best possible) solution for the given specifications.

Therefore, the synthesis problem is equivalent to a numerical optimization problem, where the aim of optimization is to find the global minima of the cost function. Then, the global optimization problem can be formally stated as:

$$\begin{aligned} &\underset{X \in \Re^n}{minimize} && \textbf{objective\_function}(X) \\ &subject\ to && l \le \begin{Bmatrix} X \\ C_{linear} X \\ C_{non-linear}(X) \end{Bmatrix} \le u \end{aligned} \qquad \textbf{Eq. (II.1)}$$

where X is vector of layout dimensions to be optimized, C_{linear} is the linear constraint formulation matrix and $C_{non\text{-}linear}$ is an array of non-linear constraints to be met.

In this report, we will discuss two approaches to synthesis of MEMS. The first approach, equation-based synthesis, uses a conjugate gradient-based search algorithm [13] and lumped parameter analytical equations

[6] as its evaluation engine. Simulation-based synthesis, on the other hand, uses a stochastic search algorithm, simulated annealing [14], and nodal simulation in NODAS [9] as the evaluation engine.

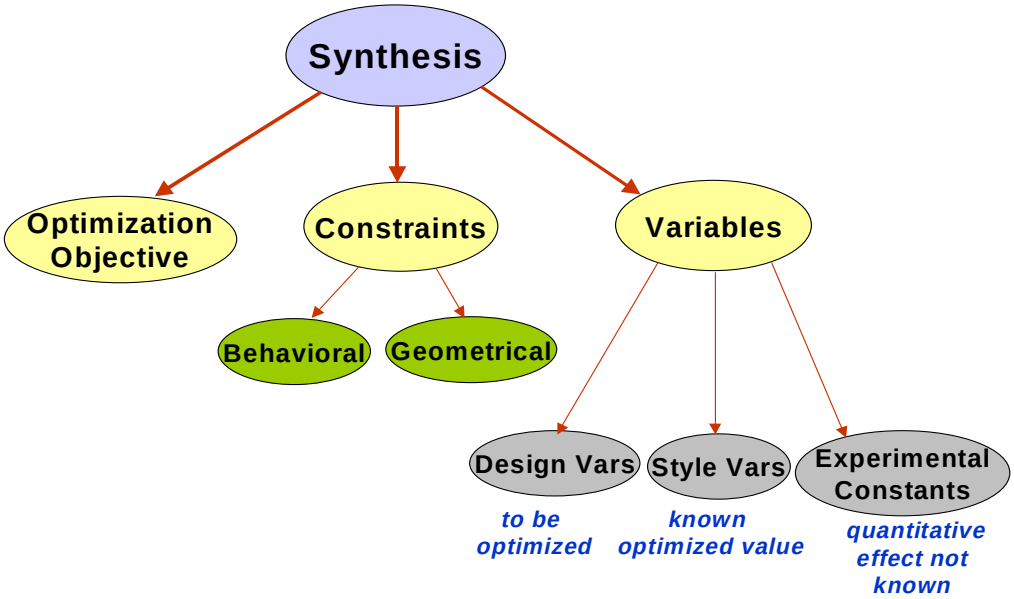


Fig. II.3: Modules in Synthesis

This generic formulation of the synthesis problem into a numerical optimization problem consists of an optimization objective, a number of design variables and design constraints (refer *Fig. II.3*). The optimization objective guides the search for an optimal solution. For an accelerometer, the objective can be either to maximize sensitivity, minimize noise or to minimize the sensor area.

The accelerometer design is behaviorally constrained to meet the user specifications of sensitivity, noise, cross-axis sensitivity, bandwidth and total sensor area. In addition, there are geometrical constraints that ensure the design rule correctness and physical validity of the layout, *e.g.*, all the sense and force comb-finger units must fit on the plate-mass. Also, for CMOS micromachining there are constraints on gaps around different structures based on their release characteristics, *e.g.*, a thick beam needs more space around it for proper release as compared to a thin beam. All such constraints constitute the linear and non-linear constraint matrices in Eq. (II.1).

Finally, the variable space of the synthesis problem is constituted by all the dimensions of the sensor layout. This variable space can be divided into following three categories:

(a) design variables: these are the dimensions to be varied in order to optimize the sensor design, *e.g.*, the length & width of the spring beam

(b) style variables: these dimensions don't affect the sensor behavior directly and hence, have a fixed optimized value for a given topology, *e.g.*, the optimal dimension of the gap between the rotor finger and the frame is its minimum possible (design rule correct) value

(c) experimental constants: these are the dimensions whose effect on the sensor behavior is not completely quantified as yet, and so these dimensions are fixed to values that have worked in previous experimental (manual) designs, *e.g.*, the size of etch-holes in the plate-mass

II.3 Design Considerations for a CMOS Accelerometer

The behavior of an accelerometer is characterized in terms of the following terms:

- *sensitivity*: the output voltage of the sensor per unit acceleration,
- *noise*: the minimum acceleration detectable by the accelerometer,
- *offset*: the acceleration required to make the output voltage zero,
- *range*: the maximum acceleration detectable by the device,
- *cross-axis sensitivity*: the output voltage of the accelerometer purely due to forces in directions orthogonal to its primary sensing axis, and
- *bandwidth*: the range of frequency of acceleration for which the accelerometer works.

For a good accelerometer, one would like to have a high sensitivity, a low noise, close to zero offset, a high range, a high bandwidth, and close to zero cross-axis sensitivity. Thus, synthesis of an accelerometer involves searching the design space for a design that meets these characteristics as specified by the user.

The primary constituent elements of an accelerometer are: the plate-mass, the springs and the comb-fingers. The design and style variables and experimental constants involved with design of each of these constituents are discussed here. Also, the constraints associated with the synthesis formulation of a CMOS accelerometer are introduced in this section.

II.3.1 Plate-mass Design

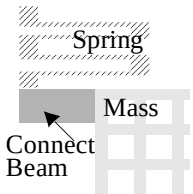
The plate-mass in a CMOS accelerometer consists of unit squares with etch holes for release. Ideally, the synthesis methodology would like to optimize the size of these squares and mass density. However, this unit square is chosen to have a fixed size since, the exact *non-linear* relationship between the size of etch holes required to release a particular area of plate-mass is not yet determined. Plate-mass with unit squares of this fixed size (9μm wide square with 6μm etch hole) has been repeatedly released in earlier experiments. In other words, the hole size and the unit square size are experimental constants for this formulation. Thus, the design variables for the plate-mass are reduced to: the number of unit squares in *x* and *y* directions, referred to as W_{pmass} and L_{pmass} respectively (see *Fig. II.4(a)*).

The plate-mass is defined by M3 layer, which is always grounded. A fixed topology routing network uses M1 and M2 to conduct signals to the rotor fingers, as shown in *Fig. II.2*. The plate-mass area not being used for routing is filled with grounded M1 and M2, all metal layers contributing to increasing the density and lowering the vertical curl in the structure [15].

II.3.2 Spring Design

A serpentine spring with odd number of truss beams is used. As shown in [16], such a spring topology has very low cross-axis sensitivity, assuming only beam width variations. The lengths and widths of the truss and the main beam are the design variables for this spring (see *Fig. II.4(b)*). The number of truss beams is also chosen to be a design variable in equation-based synthesis formulation ($NumTrussBeams \in \{3, 5, 7, 9\}$). However, in simulation-based synthesis, changing the number of truss beams changes the topology of the spring (which requires a change in the simulation netlists), and hence, has been avoided ($NumTrussBeams = 5$).

A very stiff beam is used at each corner to connect the plate-mass to the springs. In order to ensure that this connecting beam does not affect the cross-axis properties of the serpentine spring, a constraint is placed on its stiffness to be higher than ten times the spring stiffness ($K_{y(connect\ beam)} > 10*K_{y(spring)}$), with the width of this beam used as a design variable.



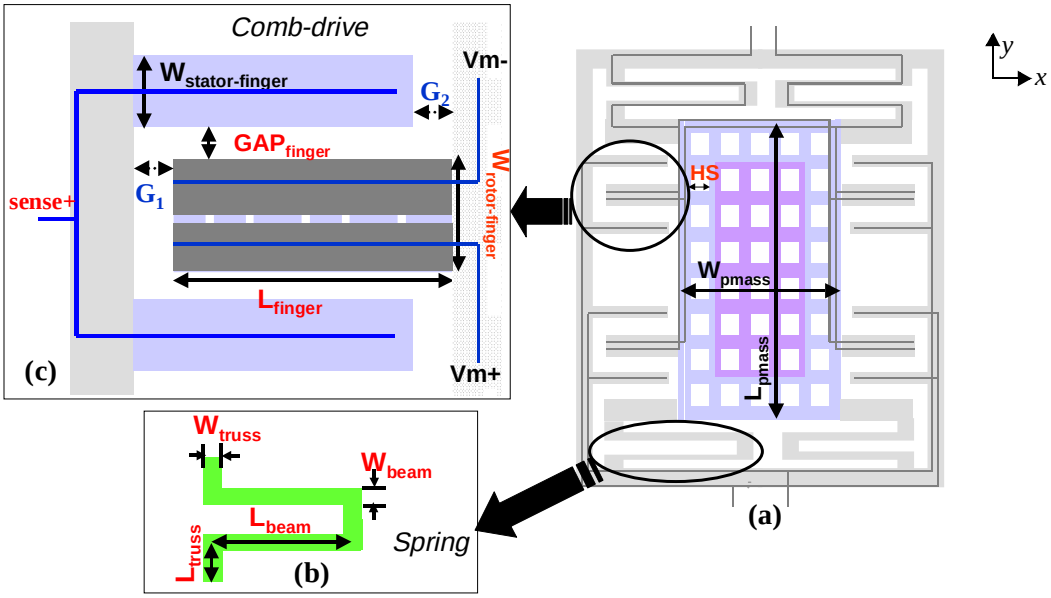


Fig. II.4 Design and Style Variables for CMOS Accelerometer Synthesis

For reduced lateral curl, the spring beams consist of all three, equally wide metal layers; though least possible metallization in the spring beams would be preferred for more flexibility.

II.3.3 Comb-finger Design

Each rotor finger has two electrical nodes as shown in *Fig. II.4(c)*. The primary design variables here are: the length (L_{finger}) and width ($W_{rotor\text{-}finger}$) of the rotor finger, and the gap between the rotor finger and stator finger (GAP_{finger}). For maximum sidewall capacitance, all the layers (poly, M1, M2 and M3) are shorted on each electrical node. Since, two electrical nodes have to be separated on a single mechanical finger, the process design rules for electrical separation define the lower limit on the width of the finger. This width (4µm for a 0.5µm process) is equally shared by the two electrical nodes. The upper limit on finger-width is determined by the release constraints. The maximum width of finger that can be released with minimum (design-rule correct) gap around it has been experimentally determined to be 5.7µm [11].

The upper limit on length of the rotor finger is set by out-of-plane curl limitations, an unavoidable feature of multi-layered structures. Instead of the absolute out-of-plane curl, we are limited by the relative curl in the rotor and stator fingers. Though, the peripheral frame matches the curl between rotor fingers and stator fingers

to first order, it does not eliminate the curl mismatch completely. In the absence of the peripheral frame, a 138μm long rotor-finger would curl out of plane by 5μm (for an average radius of curvature of 1.9mm [15]), so that there would be no overlap between the rotor and stator fingers at the end of the rotor finger (average thickness of stator finger ~ 5μm). Thus, a limit of 120μm is set for maximum rotor finger length. Further investigations to characterize the curl mismatch relationship as a function of finger length in the presence of a frame, and use of this relationship in synthesis will lead to more optimal designs.

The width of stator finger ($W_{stator\text{-}finger}$), the gap between stator finger and plate-mass (G_2), and the gap between the rotor-finger and the frame (G_1) are all *style variables*. These layout dimensions do not affect the accelerometer behavior directly, and their minimum possible (design-rule correct) values are the optimal values for them. However, for preliminary curl match in the fingers, the stator finger width is fixed to be the lower limit on the rotor finger width (4μm for a 0.5μm process).

II.4 CMOS Accelerometer Design and Style Variables

The following tables list the design and style variables for this implementation of synthesis of a CMOS accelerometer, as discussed in the previous three sub-sections. For a graphical representation of these variables, refer *Fig. II.4* and *Appendix I*. The minimum and maximum values of the design variables and the optimal values for the style variables are mostly determined by design rule correctness. The area limits on the transducer are set based on the manual design in [11] (where *x-size*=270μm and *y-size*=500μm). The maximum width of a beam that can be released easily is 20μm [17].

Table 1: Design Variables for a CMOS Accelerometer

Var	Description	Min Value	Max Value
L_{beam}	length of spring beam	1.5μm	*x-size*/2
W_{beam}	width of spring beam	1.8μm	20μm
L_{truss}	length of spring truss	2.4μm	*y-size*/4
W_{truss}	width of spring truss	5μm	20μm

Table 1: Design Variables for a CMOS Accelerometer

Var	Description	Min Value	Max Value
$Num_{trussbeams}$*	# truss beams in the serpentine spring (odd)	3	9
L_{finger}	length of rotor finger	5μm	*x-size*/2
$W_{rotor\text{-}finger}$	width of rotor finger	4μm	5.7μm
G_{limit}	gap between spring and limit-stop	0.9μm	10μm
GAP_{finger}	gap between stator & rotor fingers *in the sensing unit*	1.5μm	20μm
$W_{strongbeam}$	Width of the beam connecting the spring to plate-mass	1.5μm	10μm
L_{pmass}	# unit squares in plate-mass in *y*-direction	5	*y-size*/(9μm)
W_{pmass}	# unit squares in plate-mass in *x*-direction	5	*x-size*/(9μm)
$Num_{sfinger}$	# sense comb fingers	1	100
$Num_{ffinger}$	# force comb fingers	1	100

*. not a design variable for simulation-based synthesis

Table 2: Style Variables for a CMOS Accelerometer

Symbol	Description	Value
$G_{force\text{-}comb}$	gap between stator & rotor fingers *in the force unit*	G_{limit}+0.3μm
G_{spring}	horizontal gap between two springs at top and bottom	3μm
G_1	gap between rotor finger and frame	1.5μm
G_2	gap between stator finger and plate-mass	1.5μm
$W_{stator\text{-}finger}$	width of stator finger	4μm
$L_{sanchor}$	length (*y*-dimension) of the anchor to the spring	36μm
$W_{sanchor}$	width (*x*-dimension) of the anchor to the spring	36μm
$W_{framebeam}$	width of beams inside the peripheral frame	4.5μm
$G_{framebeam}$	gap between beams in the peripheral frame	11.2μm

The gap between the comb-fingers in the force comb-drives primarily affects the self-test force only. So, its optimal value is the minimum value ensuring that the force comb-fingers don't crash when the limit-stop is

hit. Therefore, it is assigned a dependent value of (G_{limit} + 0.3μm). All other style variable values listed in table 2 are DRC determined. Additionally, there are two important experimental constants: *Square_Size*, the size of a unit square in the plate-mass (=9μm), and *Hole_Size*, the size of etch hole in the unit square (=6μm).

II.5 CMOS Accelerometer Synthesis Constraints

There are two types of constraints in the synthesis formulation: behavioral and geometrical. The behavioral constraints are functional constraints to be met by the design, while the geometrical constraints are required in order to ensure the validity of the physical layout of the accelerometer.

The synthesized design is functionally constrained to meet the user specifications of sensitivity, noise, off-set, range, cross-axis sensitivity and bandwidth (defined in section II.3). Due to the presence of a parallel plate gap between comb-fingers, the displacement of the rotor finger causes an effective electrical force on the rotor finger. This leads to increased displacement of the plate-mass. This phenomenon is termed as electrical spring softening. In order to prevent excessive electrical spring softening, a constraint is placed to keep the electrical spring constant less than 80% of the mechanical spring constant. Also, for being able to self-test the accelerometer (using force comb-drives), the self-test force is constrained to be equivalent to at least 20% of the sensor range.

To prevent mode coupling, the lowest order resonant modes in x and θ (non-primary axes) are constrained to be away from the resonant mode in y (primary sensing axis) by at least a factor of two, *i.e.*, $\{\omega_x, \omega_\theta\} > 2\omega_y$. Another constraint ensures that the beam connecting the spring to the plate-mass does not buckle under the residual stress of the plate-mass.

The *distributed* electrical force acting on the rotor finger due to modulation voltage causes the rotor finger to bend. This can potentially cause errors in the measurement of plate-mass displacement. Therefore, the resonant frequency of each individual rotor finger is constrained to be less than the modulation voltage frequency by at least a factor of 1.5, i.e., $\omega_{finger} < \omega_{modulation\_voltage}/1.5$. This constraint ensures that the response of a

rotor finger at modulation voltage frequency is reduced by at least 3dB as compared to its low-frequency bending response.

A list of the behavioral constraints on a CMOS accelerometer is presented in Table 3. The functions *sens, noise, range, cross-axis sens, bandwidth,* K_{elec}, K_{mech}, ω_x, ω_θ, ω_y , *self_test_force, mass,* $L_{critical}$, ω_{finger}, *etc.* are all functions of the design and style variables, and the analytical expressions for these functions are listed in [6].

Table 3: Behavioral Constraints for CMOS Accelerometer Synthesis

Constraint	Expression
Sensitivity	*user_spec < sens < 1V/G*
Noise	*0 < noise < user_spec*
Range	*user_spec < range < 1000G*
Cross-axis sensitivity	*0 < cross-axis sens < user_spec*
Bandwidth	*user_spec < bandwidth*
Electrical spring softening	$0 < K_{elec} < 0.8*K_{mech}$
Frequency mode separation	$\{\omega_x, \omega_\theta\} > 2\omega_y$
Self test-force	*self_test_force > 0.2*mass*Range*
Buckling of connect beam	$L_{connect\text{-}beam} < L_{critical}$
Stiffness of connect beam	$K_{y(connect\ beam)} > 10*K_{y(spring)}$
Resonant frequency of individual rotor finger	$\omega_{modulation\_voltage} > 1.5*\omega_{finger}$

In addition, there are geometrical constraints to be met by the accelerometer layout. The design must meet the user specified area limitations (*x-size, y-size*). It must meet all process design rules for CMOS (*e.g.*, minimum Metal3 gap = 0.9μm) and it must meet the relative size constraints of the constituent elements (*e.g.*, all the comb-fingers must fit onto the plate-mass unit). Also, the MEMS-related DRC rules [17] have to be satisfied *e.g.*, special gap constraints are required to ensure proper release of the mechanical structure and constraints on beam dimensions ensure reasonable lateral/out-of-plane curl, *etc*.

A complete list of the geometric constraints is presented here:

Table 4: Geometric Constraints for CMOS Accelerometer Synthesis

Constraint	Expression	lower bound	upper bound
accelerometer *y-size*	L_{pmass}*Square_Size + 2*($Num_{trussbeams}$*L_{truss} + $L_{sanchor}$)	350µm	500µm
accelerometer *x-size*	W_{pmass}*Square_Size + 2*(L_{finger} + G_1 + $L_{sanchor}$)	200µm	270µm
spring *x*-size	2*(L_{beam} + W_{truss}) + G_{spring}	50µm	270µm
gap between fingers when the limit-stop is hit	GAP_{finger} - G_{limit}	0.3µm	10µm
gap between horizontal beams of the spring	L_{truss} - W_{beam}	4.5µm	10µm
spring width *vs.* mass+comb width	0.5*(W_{pmass}*Square_Size) + L_{finger} - L_{beam} - W_{truss}	G_{spring} /2	100µm
gap between spring and comb-drive (gap around the connect-beam)	[L_{pmass}*Square_Size - 2*$W_{strongbeam}$ - 2*$Num_{sfinger}$*($W_{rotor\text{-}finger}$ + 2*GAP_{finger} + $W_{stator\text{-}finger}$) - 2*$Num_{ffinger}$*($W_{rotor\text{-}finger}$ + 2*$G_{ffinger}$ + $W_{stator\text{-}finger}$)]/2	5µm	50µm

Using the above-stated formulation of layout dimensions (variables), constraints and optimization objectives, the optimization algorithm generates optimal values for the design variables. Thereafter, a hierarchical parametrized layout generator is used to generate the layouts for the constituting elements: spring, comb-finger, plate-mass, routing, and finally for the complete accelerometer.

III. Equation-Based Synthesis

In this approach, lumped parameter analytical equations are used to model the behavior of the accelerometer. This enables easy computation of the objective and constraint functions as well as their derivatives. Therefore, in this approach, the synthesis problem is formulated into a gradient-based non-linear constrained optimization problem. The search algorithm in this approach uses multiple grid point sampling for finding initial startpoints, sequential quadratic programming (SQP) for gradient-based optimization and brand and bound algorithm for handling integer valued variables. The optimization solution is finally used by a hierarchical parametric layout generator to produce the CMOS accelerometer layout.

III.1 Synthesis Algorithm

A non-linear constrained optimizer, NPSOL [13] is used for synthesis. This package uses Sequential Quadratic Programming (SQP) for optimization [18][19]. Each optimization starts out by finding a point that is feasible with respect to the bounds and linear constraints; and then involves a combination of major and minor iterations (in order to compute the steplength and the search direction respectively). The optimization problem is broken down into a quadratic subproblem using the gradient of the optimization objective function, and the Hessian of the Lagrangian function (see Eq. (III.1)). This quadratic subproblem is solved using standard quadratic programming (QP) techniques. This QP constitutes the minor iterations of SQP, and provides the next search direction as its result. The major iteration uses this search direction to compute a steplength that causes a decrease in the augmented Lagrangian merit function. The contour information obtained during this steplength computation is also used to update the Hessian of the Lagrangian optimization function. Thus, a number of major iterations on solving the sequentially modified quadratic subproblem results in the optimization solution.

$$\begin{bmatrix} \text{minimize } f(\bar{x}) \\ \text{subject to } C_i(\bar{x}) \end{bmatrix} \Rightarrow \begin{Bmatrix} \text{Lagrangian} = L(\bar{x}, \bar{\lambda}) = f(\bar{x}) - \sum_i \lambda_i \cdot C_i(\bar{x}) \end{Bmatrix} \quad \textbf{Eq. (III.1)}$$
$$\{\text{Hessian} = H(\bar{x}) = \nabla_x^2 L(\bar{x}, \bar{\lambda})\}$$

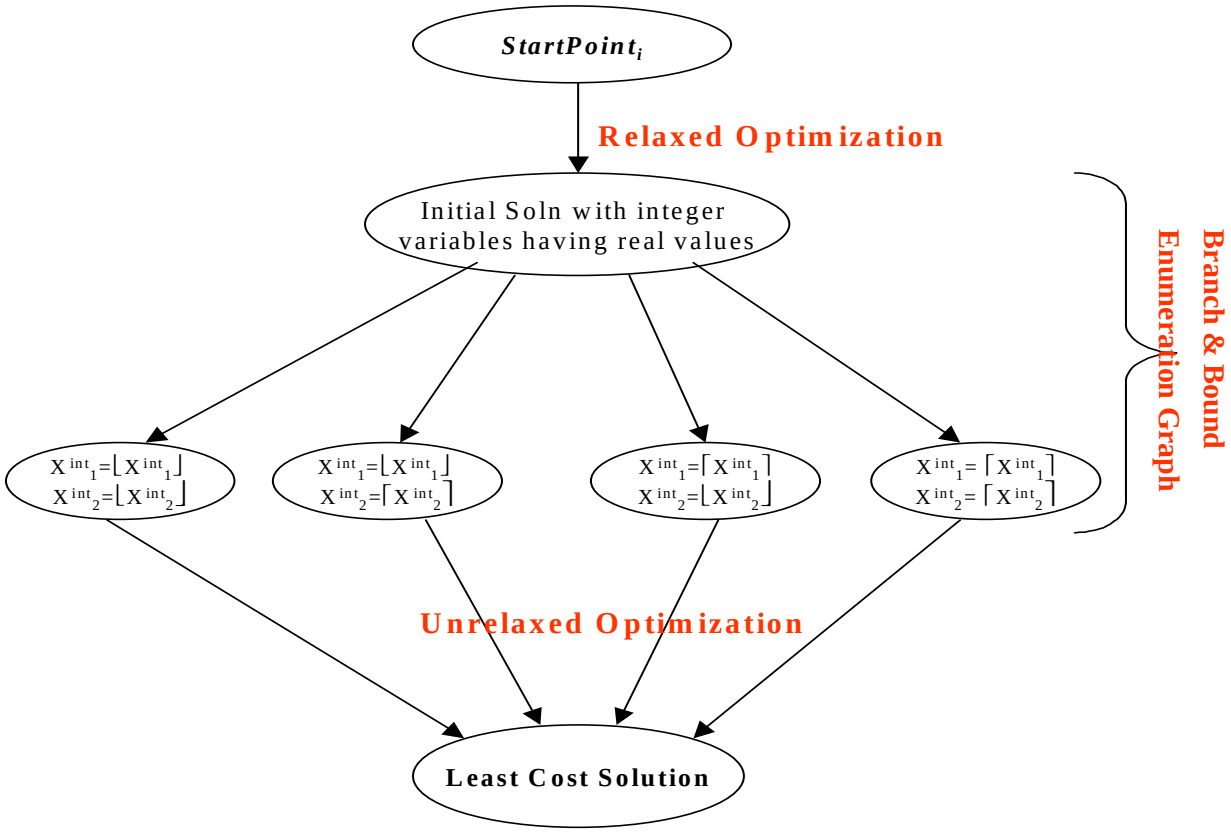


Fig. III.1 Flowchart for the Synthesis Algorithm

During the search for the optimal solution, non-linear constraints are added sequentially to the gradient-based optimization. Also, in order to accommodate integer valued variables in this formulation, branch and bound algorithm is used. Fortunately, the number of integer valued variables in our formulation is very small. Consequently, instead of using complicated search pruning techniques for branch and bound algorithm, we can just evaluate all the possible combinations, and pick the least cost solution. Therefore, the depth of the state-space/enumeration graph for branch and bound algorithm is just one, where we go from the root to all possible enumeration/evaluation of leaf nodes (see *Fig. III.1*).

With this simple implementation of branch and bound algorithm for integer valued variables, the optimization is carried out in two steps. First, the integer valued variables are relaxed to have real values and an approximate solution is obtained from the non-linear constrained optimization formulation. Then, an unrelaxed optimization problem is formulated with all the integer valued variables fixed to a particular value,

which being either the upper bound or the lower bound integer of the solution from the first optimization run. Finally, for all the possible combinations of integer valued variables taking the lower bound value ($\lfloor x \rfloor$) or the upper bound value ($\lceil x \rceil$), the unrelaxed optimization is performed; and the least cost solution is chosen.

However, gradient based optimization techniques are inherently quite sensitive to the starting solution for the optimization as they can get stuck in a minima local to the starting point. Thus, in order to obtain a global minimum rather than local minima, multiple sampling of the design variable space is done to obtain 100 starting points, and the best solution of the 100 minima is chosen as the global minimum. For all the optimization runs, the optimal solution was usually found to be within the first 50 start-points.

III.2 Lumped Parameter Accelerometer Models

As a system, the accelerometer is modeled as second order mass-spring-damper (mechanical) system followed by a capacitive transducer (see *Fig. II.2*). Analytical equations in three modes (in-plane *x*, *y*, θ) for a CMOS accelerometer system are similar to those derived for a polysilicon accelerometer system in [6].

The in-plane behavior of the serpentine springs is fully characterized using analytical models mentioned in [16], which are within ±5% of FEM simulations. The effect of spring mass on the system behavior is taken into account using effective mass of the spring in *x* and *y* directions (using the static mode shapes).

The electrical behavior of the accelerometer is modeled as a capacitive bridge with parasitic capacitance,

$$V_{sense+} = \frac{(C_1 - C_2)V_m}{C_1 + C_2 + C_{para}} \qquad \textbf{Eq. (III.2)}$$

where V_m is the high frequency modulation voltage applied across the bridge, and C_1, C_2 and C_{para} are as shown in *Fig. II.2*. The parasitic capacitance (C_{para}) consists of the sensor layout parasitic capacitance (computed during synthesis) and the input capacitance of the first gain-stage of the read-out circuit (specified by the user as a lumped quantity). The analytical expressions for the capacitance calculation for comb-drives have been taken from [20], which model the parallel plate as well as the fringe-field capacitances.

The noise of the system is modeled as Brownian noise of the mechanical system and the user-specified input referred noise of the first gain-stage of the read-out circuit ($\bar{v}_{n\text{-}ckt}$), as in Eq. (III.3(a)). The Brownian

mechanical noise of the accelerometer is dependent on the total mechanical damping coefficient ($B_{damping}$) and total system mass (m_{tot}), as is described in Eq. (III.3(b)).

$$Total\ Noise = \sqrt{\left(\frac{\overline{v_{n-ckt}}}{Sensitivity}\right)^2 + (\overline{Brownian\ Noise})^2} \qquad \textbf{Eq. (III.3(a))}$$

$$\overline{Brownian\ Noise} = \frac{\sqrt{4 \cdot k_{Boltzmann} \cdot T_{absolute} \cdot B_{damping}}}{m_{tot}} \qquad \textbf{Eq. (III.3(b))}$$

The viscous damping experienced by the accelerometer is modeled as Couette-flow damping below the plate-mass, Stokes-flow above the accelerometer, and squeeze-film damping between the comb-fingers [6]. For this design, squeeze-film damping usually dominates over all other forms of damping, requiring more accurate models [21]. Previously, it has been proposed to assume Hagen-Poiseuille flow between comb-fingers for modeling squeeze film damping [22][23]. This assumption leads to a damping coefficient given by .

$$B_{HP} \cong 7.2\mu_{air} w \left(\frac{l}{t}\right)^3 \qquad \textbf{Eq. (III.4)}$$

where, l, t and w are the length, width and thickness of the squeeze film, and μ_{air} is the viscosity of the air.

However, for typical dimensions of comb-finger gaps (1.5μm – 4μm) used in CMOS accelerometer design, this equation does not match well the model reported in [21]. The squeeze film model in [21] has been shown to match FEM simulations within ±5%, and the Hagen-Poiseuille damping model has more than 300% error with respect to this model (see *Fig. III.2*).

So, a curve fit has been performed based on the following equations:

$$B_{CurveFit} \cong B_{HP} \cdot (t)^{k_1} \cdot k_0 \qquad \textbf{Eq. (III.5)}$$

$$k_1 = 1.18 \qquad k_0 = 0.13 \cdot (10^6)^{k_1} = 1.563 \times 10^6$$

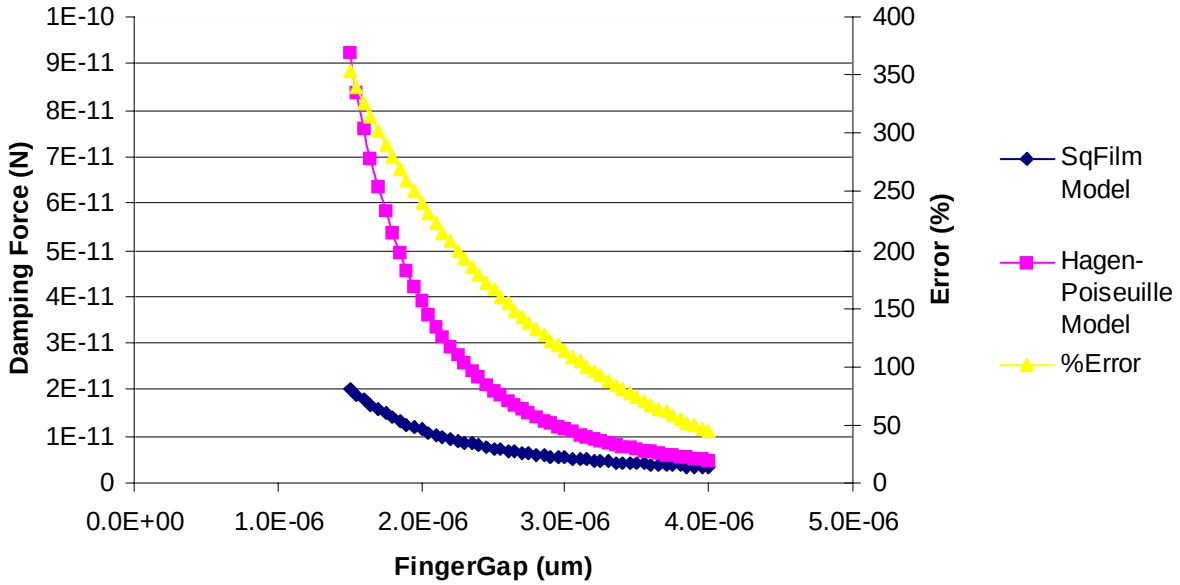


Fig. III.2 Squeeze Film Model [21] *vs.* Hagen-Poiseuille Damping Model

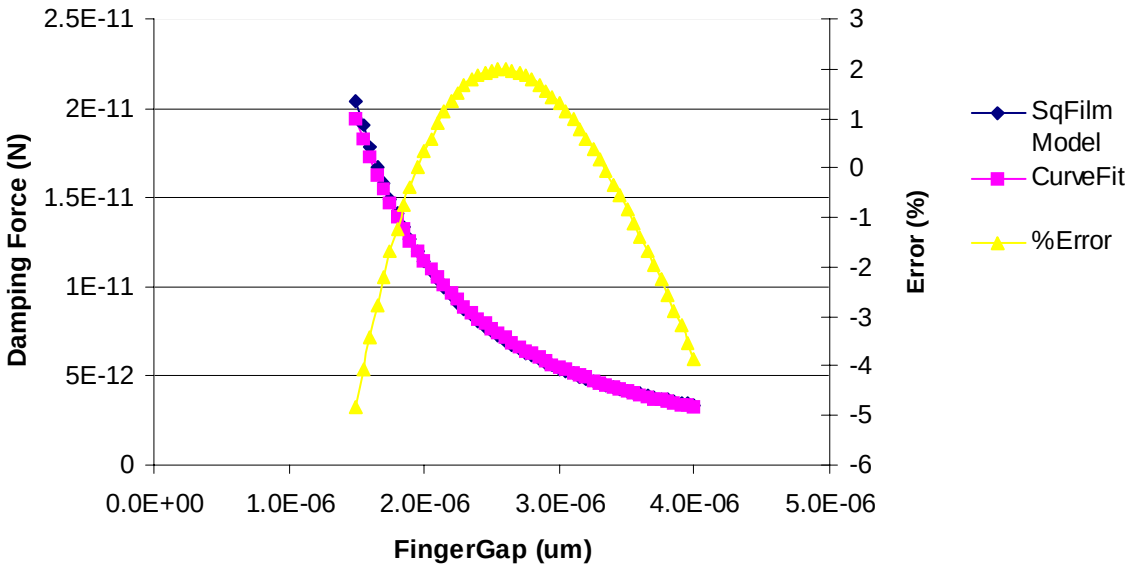


Fig. III.3 Squeeze Film Model [21] *vs.* Curve-Fitted Damping Model

This leads to a squeeze film model accurate to within ±5% as compared to the model reported in [21] (see *Fig. III.3*). Thus, this modified lumped parameter analytical model helps in better estimation of the Brownian mechanical noise of the sensor.

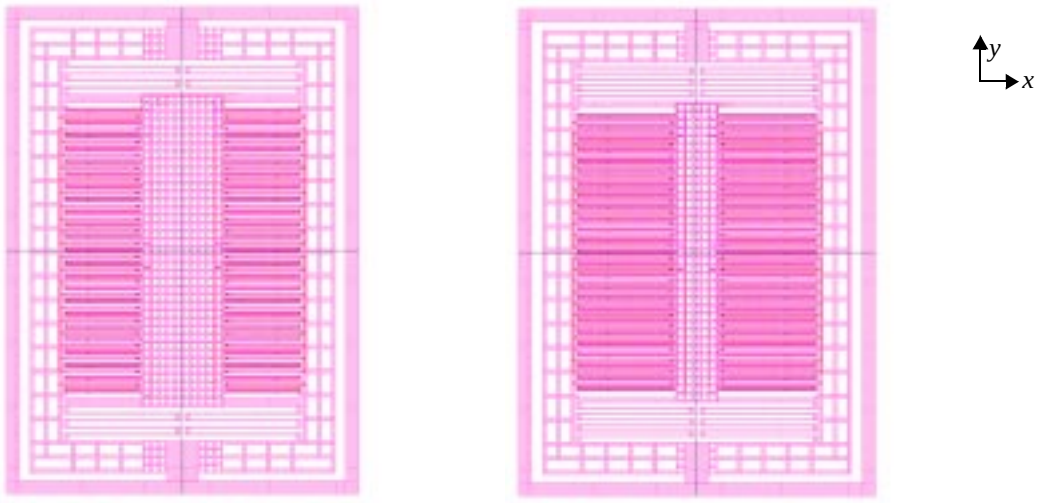
III.3 Synthesis Results

A prototype tool has been developed based on the synthesis methodology described in the previous sections. Layouts of CMOS accelerometers synthesized using this tool, for three different optimization objectives, are presented here. The details of these designs can be found in *Appendix II*. A short comparison of the area optimization case with a manual design is also presented. Finally, the sensitivity *vs.* noise trade-off for accelerometer design is discussed.

III.3.1 Noise and Sensitivity Optimization

The accelerometers presented in *Fig. III.4* have a common specification of range 25G, bandwidth 1kHz, cross-axis sensitivity 1% and maximum allowed area of 270 x 500 $(\mu m)^2$. The specified area limits are based on a manual design [11]. Additionally, the circuit noise is specified to be 10nV/$\sqrt{Hz}$, and the input capacitance of the circuit to be 149fF.

With a modulation voltage frequency of 2MHz, the constraint on the resonant frequency of an individual rotor finger (ω_{finger}) (see section II.5, table 3) forces ω_{finger} to be less than 1.33MHz. This constraint usually leads to long comb-fingers in the synthesized designs.



(a) Gap=2.31μm ω_y=6.83kHz K_y=0.93N/m (b) Gap=1.5μm ω_y=5.52kHz K_y=0.57N/m

Fig. III.4: Accelerometers optimized for (a) minimal noise & (b) maximal sensitivity

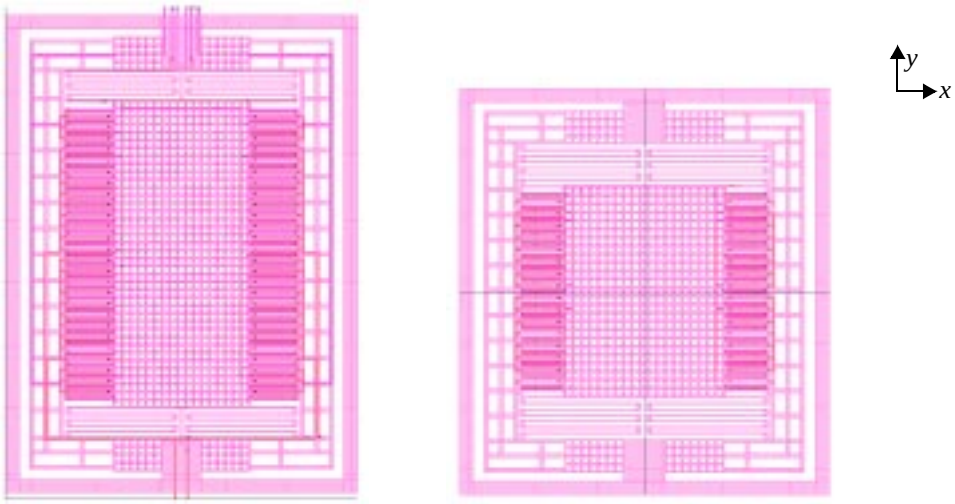
Synthesis with an objective to minimize noise results in a design shown in *Fig. III.4(a)*. This design has a sensitivity specification of 0.57mV/G. The system noise predicted by the synthesis module is 50μG/$\sqrt{Hz}$. This design uses maximum allowed area to capture maximal mass. The highlight of this design is a large gap (of 2.31μm) between the rotor and stator fingers. This shows that for the current specifications, squeeze film damping in the finger gaps dominates over other forms of damping, and that the Brownian mechanical noise dominates over electrical noise. Thus, the optimal solution to minimize noise is to enlarge finger gaps, and compensate for the reduced sensitivity with a low resonant frequency.

Fig. III.4(b) shows the accelerometer layout optimized for maximal sensitivity of *1.996mV/G* under the noise constraint of 100μG/$\sqrt{\text{Hz}}$. The highlights of this design are: *a narrow finger gap (of 1.5μm) and a flexible spring (K_y=0.57N/m)*. Both of these design choices lead to increased sensitivity (softer spring ⇒ reduced resonant frequency, narrow finger gap ⇒ increased sense capacitance). Due to limited area, an increase in number of spring beams leads to decrease in plate-mass size, and hence decrease in number of rotor fingers. Thus, the maximum allowed area places a limit on the maximum sensitivity obtainable within a particular amount of noise.

III.3.2 Area Optimization vs. Manual Design

Fig. III.5(a) is our reference, manually designed accelerometer [11] with sensitivity of 0.57mV/G and a noise floor of 57μG/$\sqrt{\text{Hz}}$. The remaining characteristics of this design are same as the specifications for the noise and sensitivity optimizations above. However, the resonant frequency of the rotor finger (ω_{finger}) in the manual design in 2.5MHz. Therefore, in order to have a fair comparison, the constraint on maximum value of ω_{finger} has been relaxed to 2.5MHz for area optimization.

Synthesis for minimizing area with specifications equal to that of the manual design results in the design shown in *Fig. III.5(b)*. This design uses a flexible spring (K_y=0.631N/m as compared to the manual design spring constant of K_y=1.6N/m), so that the sensitivity specification can be met with least possible mass and hence, area. However, the noise specification places a lower limit on the mass, to counter which the finger-gap



(a) K_y=1.6N/m
Mass=0.56μg Gap=1.5μ*m*

(b) K_y=0.631N/m
Mass=0.38μg Gap=1.6μ*m*

Fig. III.5: (a) Manually designed accelerometer (b) Accelerometer optimized for minimal area

is also slightly increased in this optimization (in order to reduce damping and hence, reduce noise). The synthesized accelerometer shows a 23% area improvement over the manual design, and has the maximum allowed noise floor of $57\mu G/\sqrt{Hz}$.

III.3.3 Design Space Exploration

Fig. III.6 shows the sensitivity *vs.* noise trade-off in accelerometer design optimization. The designs are optimized to minimize noise for different sensitivity values. The total system noise consists of mechanical Brownian noise and the electrical circuit noise. The minimum noise is obtained at the sensitivity of 0.4mV/G. The discrete jumps in the curves is due to presence of integer valued variables.

Naively, one would think that larger mass implies larger sensitivity and lower noise. But, as shown in *Fig. III.6*, for the given sensor configuration, *finger-gap dominates the noise figure* rather than the mass. In order to minimize mechanical noise, optimization tends to maximize the finger-gap (thus, reducing the squeeze-film damping between the comb-fingers). *To the right side of 0.4mV/G sensitivity, the mechanical noise dominates over electrical noise, while to its left, the electrical noise dominates over mechanical noise* (refer Eq. (III.3)). Consequently, towards the right, higher sensitivity implies lower finger-gap, which causes higher damping

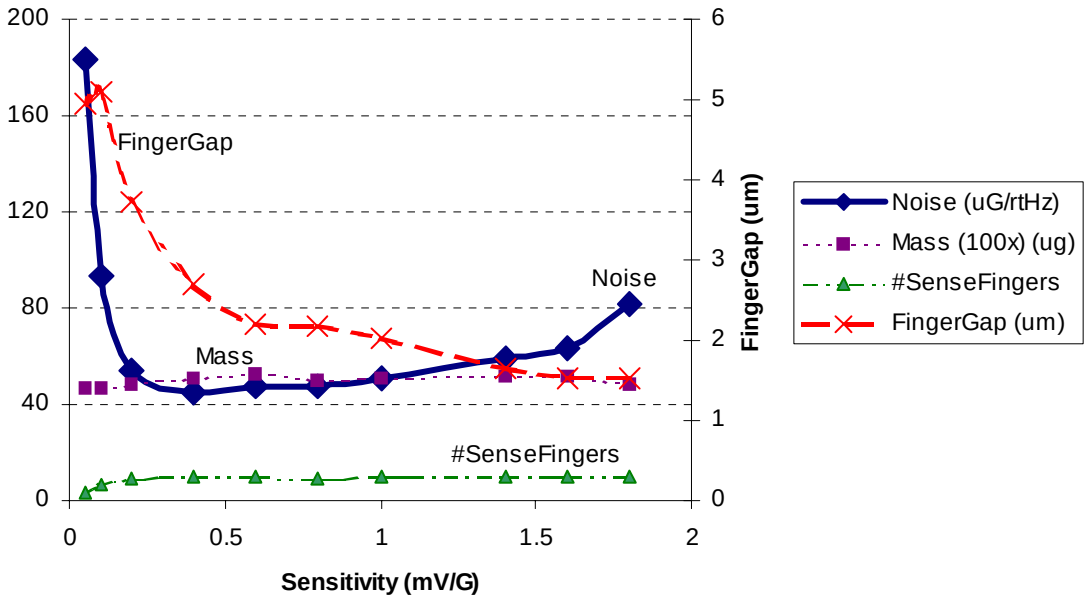


Fig. III.6: Sensitivity *vs.* Noise Trade trade-off analysis

and higher mechanical noise, leading to a higher system noise. On the other hand, to the left, decreasing sensitivity implies increasing contribution of electrical noise, and since, electrical noise is dominant over mechanical noise in this part of the graph, the system noise increases despite decreasing mechanical noise.

Coupling the above trade-off analysis with a parasitic capacitance-*vs.*-noise curve (instead of constant values) for the read-out circuit can help in exploring the sensor design as well as the circuit design together so that, we can reach an optimal *system* design.

III.4 Summary

The design and layout generation of CMOS accelerometers has been successfully automated using a synthesis methodology based on lumped parameter behavioral models. The optimizations performed using a prototype tool suggest significant improvements over manual designs. The trade-off analyses possible with synthesis help understand the design issues from a system level perspective, and lead to system design optimization in addition to device design optimization.

IV. Simulation-Based Synthesis

In this approach to synthesis, the evaluation of designs is performed using nodal simulation of the accelerometer in NODAS [9], while the search algorithm being used is simulated annealing [14]. With simulation-based evaluation, it is difficult to estimate the derivatives of the objective and constraint functions. Therefore, a non-gradient-based search algorithm, like simulated annealing has to be used in this approach. As compared to lumped parameter equation-based evaluation, simulation-based evaluation is much easier to use since, the designer need not re-derive analytical behavioral equations every time the topology changes. Simulation-based evaluation also provides more accuracy in evaluation, and thus, is the preferred method for industrial use.

IV.1 Algorithm Description

The framework for simulation-based synthesis is as shown in *Fig. IV.1*, where a layer of evaluation scripts acts as a simulator encapsulator for design evaluation. These evaluation scripts are just data processing commands written in the OCEAN environment in Cadence's IC Design tool-set. The simulation of the CMOS accelerometer system is performed in NODAS [9], which is also implemented in the IC Design framework. The netlists/schematics for accelerometer simulation are parametrized in terms of the design/optimization

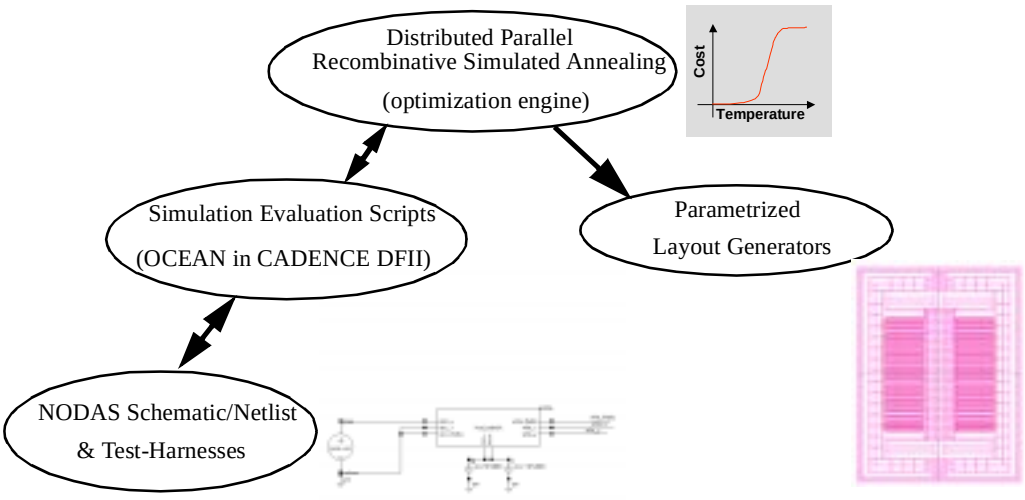


Figure IV.1: Nodal Simulation-based Synthesis Framework

variables. The topology of the accelerometer and the synthesis formulation are same as in chapter 2. The optimization solution again goes through parametrized layout generators to produce the accelerometer layout.

The search algorithm used in this framework is called parallel recombinative simulated annealing (PRSA) [8][24]. This stochastic search algorithm is favored over gradient-based search techniques primarily because of its empirical robustness even for highly non-linear and/or non-convex cost functions. PRSA uses a number of parallel annealing tasks instead of a single annealing task to search for the global optimum. These parallel annealing tasks can exchange information randomly, and, consequently, each of the parallel annealer runs a shorter schedule. The implementation of PRSA used in this framework is called Distributed PRSA, as reported in [8]. This implementation uses a genetic algorithm type cross-over move between parallel annealers in order to exchange information. It also allows the distribution of computationally expensive simulation-based evaluations across a network of computers instead of a single machine.

The basic synthesis problem (refer Eq. II.1) is mapped to an unconstrained optimization problem in this formulation. Here, we use scalar weights to weigh the objective functions ($O_i(\bar{x})$, like sensitivity, noise) as well as the constraint functions ($C_j(\bar{x})$, like total sensor area), and create a scalar cost function $COST(\bar{x})$. The

$$COST(\bar{x}) = \sum_{i=1}^{k} w_i \cdot O_i(\bar{x}) + \sum_{j=1}^{k} w_j \cdot C_j(\bar{x}) \qquad \textbf{Eq. (IV.1)}$$

minimum of this cost function is the optimal solution for the synthesis problem, and is obtained using the PRSA search algorithm. However, assigning the weights to different constraints and objective functions is not a trivial task. Without appropriate weights, the optimization might converge to an extremely bad solution. Instead of following an algorithmic approach [25], we have tried to obtain these weights just by hit and trial for different objective functions. Consequently, these weights are not yet the optimal weights, and the synthesis solutions are still under-optimized. As a future research direction, we need to work on algorithmic attempts to finding these weights for the unconstrained cost function formulation.

Distributed PRSA uses a concept of simulator encapsulation, where, a layer of evaluation scripts hides the details of simulation-based evaluation from the search engine. This technique also helps insulate the search

algorithm from pathological simulation cases. The data processing capabilities in Cadence's OCEAN environment have been used for encapsulation of NODAS. This is because, given the breadth of micro-electro-mechanical systems, we need the most generic data processing environment for encapsulation. OCEAN supports processing of all types of simulation data from NODAS through the use of evaluation scripts. However, OCEAN processes the simulation data only when it is stored on the disk. Since, a disk operation is required for each evaluation, the evaluation time for each candidate solution is *at least* of the order of 10s of seconds. A custom evaluation environment for post-processing the simulation data [8] can avoid such disk operations.

A stochastic algorithm like PRSA performs a few thousand evaluations of candidate solutions before converging to the optimal solution. This means that it takes overnight for a synthesis run even when the evaluation time is close to its lower bound (of ~10-20 seconds, placed due to involvement of disk operations). Therefore, it is absolutely essential to perform extremely time-efficient simulation of the accelerometer system. The next section describes the test-harnesses that would normally be used for accelerometer simulation, as well as their relatively much more time-efficient versions.

IV.2 Simulation of CMOS Accelerometer

The CMOS accelerometer topology described in chapter 2 uses a differential capacitive sensing mechanism. In this mechanism, two out-of-phase high frequency (~1MHz) modulation voltages are applied across the two capacitors C_1 and C_2, while the sense voltage is obtained at the common terminal, as shown in *Fig. IV.2(a)*.

This CMOS accelerometer topology is modeled as the NODAS schematic shown in *Fig. IV.2(b)*. The topology schematic in *Fig. IV.2(a)* shows four sensing comb-drives. In the NODAS schematic, however, the comb-drives corresponding to $C_{1\{a,\,b\}}$ and $C_{2\{a,\,b\}}$ have been combined into a single comb-drive on the left of plate-mass; and similarly the comb-drives for $C_{3\{a,\,b\}}$ and $C_{4\{a,\,b\}}$ are represented by the comb-drive on the right of plate-mass. The four force comb-drives shown in the NODAS schematic are not shown in the topology figure for the sake of clarity. The NODAS schematic has finally been abstracted into a subcircuit symbol

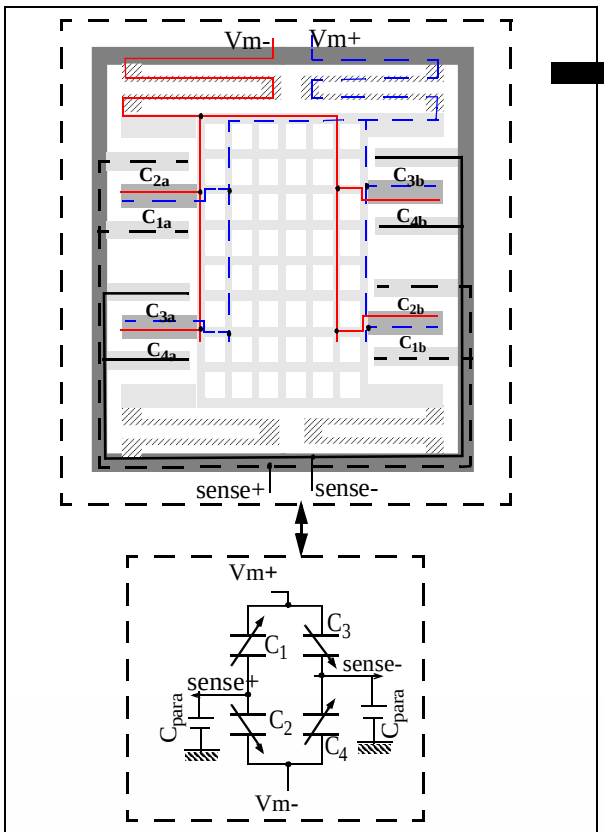


Fig. IV.2(a) Accelerometer Topology Schematic

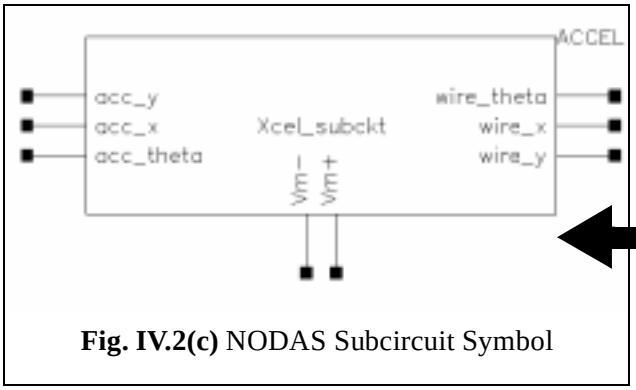


Fig. IV.2(c) NODAS Subcircuit Symbol

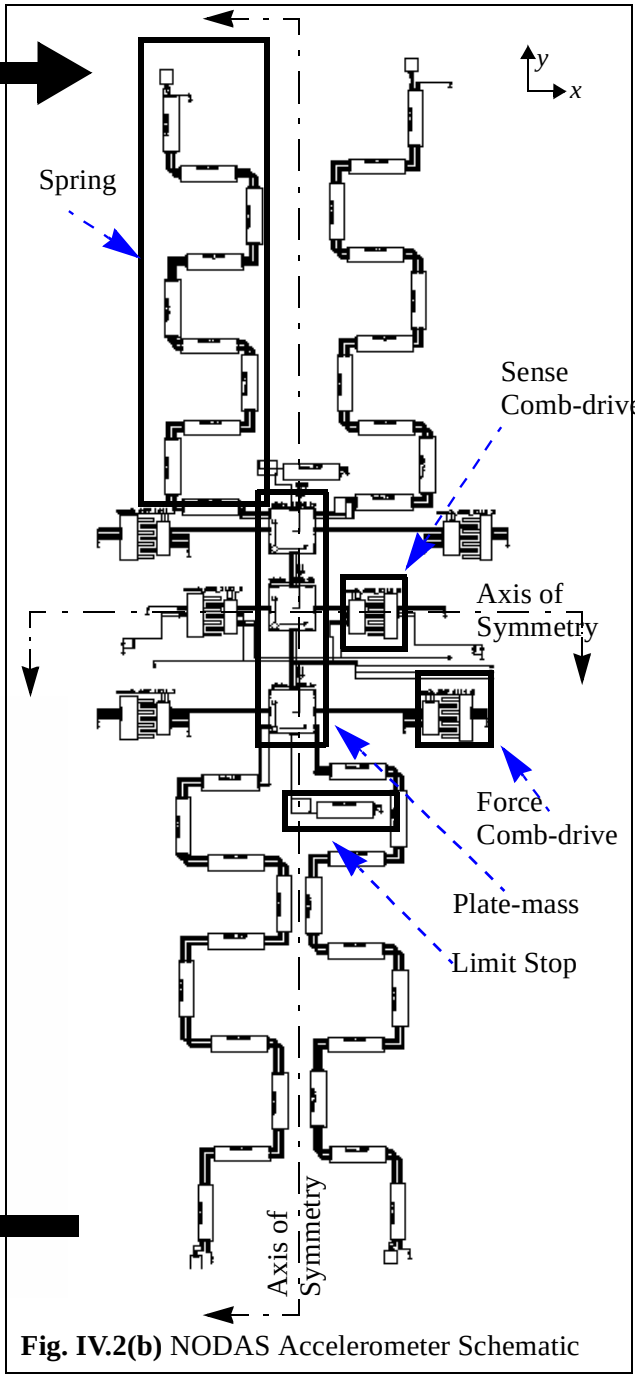


Fig. IV.2(b) NODAS Accelerometer Schematic

Fig. IV.2 CMOS Accelerometer Representations

shown in *Fig. II.2(c)*, for use in test-harnesses, where the terminals `acc_{x,y,θ}` represent the input terminals for acceleration, while `wire_{x,y,θ}` represent the input-output terminals for the displacement in *x*, *y* and θ directions respectively.

The distinguishing characteristics of a test-harness are its speed, its accuracy and the coverage of design-space for which the harness results are valid. In a simulation-based synthesis framework, the simulations need to be performed iteratively (or inside a loop) before the optimal solution can be reached. This makes the speed of simulation very critical, since, with higher simulation speed, one can either perform more evaluations in the same amount of time and thus, improve the quality of the optimal solution; or obtain the optimal solution in lesser amount of time.

Typically, a transient analysis on a system takes a long time to reach the steady state, primarily because it goes through all the non-linearities and varied time-constants of the system. On the other hand, it is much faster to perform an AC analysis, where the system behavior is linearized around the DC solution. However, since, linearization of the system behavior is only an approximation – valid for very small perturbations, AC analysis is usually inferior to transient analysis in terms of accuracy. But, if some accuracy can be traded-off for significant gain in simulation-speed, an AC analysis should be preferred, at least in the context of simulation-based synthesis.

In an accelerometer system, the modulation voltage has a frequency of the order of 1MHz, while the dominant eigenmode of mechanical system is of the order of few kHz. Thus, the mechanical part of system cannot respond to the high frequency modulation voltage. Consequently, the mechanical behavior of this system can be estimated/simulated with 99% accuracy even if the sinusoid modulation voltages are replaced by their equivalent root-mean-square (RMS) DC voltage sources. Since, this replacement is highly efficient in terms of simulation-time without adversely affecting the accuracy, it has been used in all the test-harnesses for the CMOS accelerometer synthesis.

Another key characteristic of a test-harness (from a synthesis perspective) is its coverage of the design space. This is a concern because in a synthesis environment, the search algorithm can go to any part of the design space. So, the lesser the design space for which a test-harness fails, the better the test-harness is from a synthesis perspective. For example, the eigenmode frequency of a (predominantly second order) mechanical system can be measured in two ways: as the resonant frequency where the displacement magnitude reaches its

peak; or as the frequency where the displacement phase crosses 90^{o}. Normally, a MEMS accelerometer system is under-damped so that it exhibits a resonant frequency. However, the synthesis algorithm can easily reach a candidate solution where the system is overdamped and does not exhibit resonance. Therefore, from a synthesis perspective, one should choose the latter test-harness. This choice might give slightly erroneous (< 5% error) measurements for systems that exhibit resonance but, provides a greater design-space coverage.

Similar principles regarding the simulation time, accuracy and design-space coverage ought to be applied when test-harnesses are being designed for synthesis of any other system as well.

In order to characterize the behavior of an accelerometer, we need to obtain the following measures using simulations: sensitivity, noise, range, bandwidth, offset and cross-axis sensitivity (defined in section II.3). The following sections discuss the details of the test-harnesses used for measuring these characteristics.

IV.2.1 Sensitivity Simulation

Sensitivity of an accelerometer is defined as its output voltage per unit acceleration. This definition leads to a simple harness, where one would apply a step input acceleration (of 1G), run a transient simulation and measure the steady state sense output voltage (as shown in *Fig. IV.3*). However, the transient simulation takes a long time (~1000 secs) to reach the steady state (because of reasons discussed in the previous sub-section), which makes it practically impossible to run simulations inside the optimization loop.

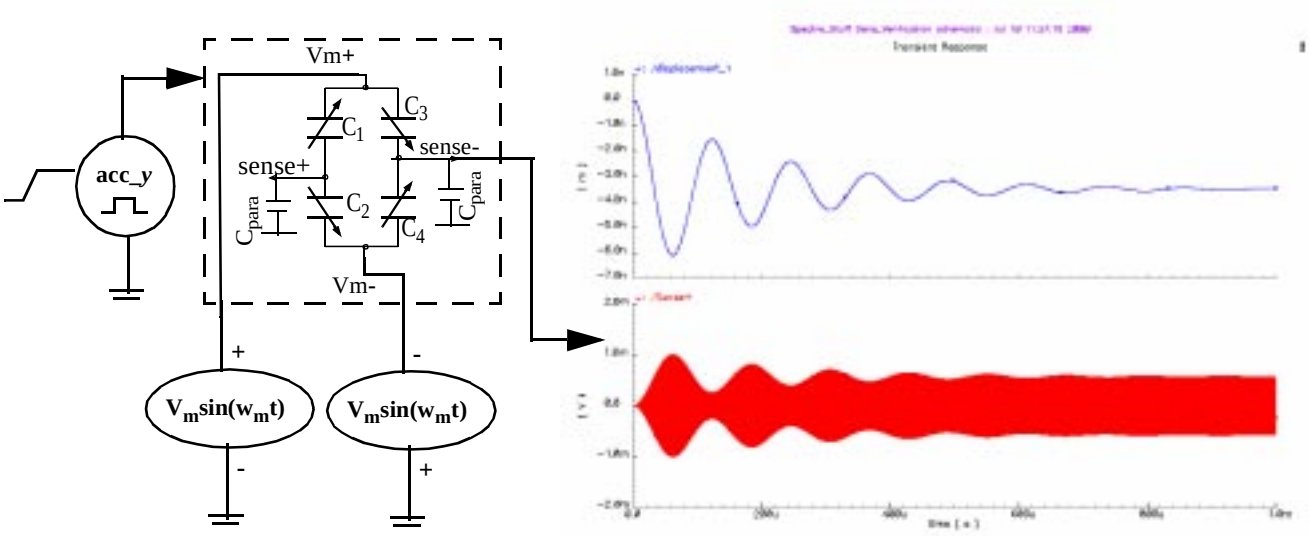


Fig. IV.3: Nominal Transient Simulation for Sensitivity Calculation

So, an efficient test-harness has been created for sensitivity evaluation. As discussed previously, the modulation voltages can be treated as constant voltages sources of their equivalent RMS values, as far as the mechanical part is concerned. This allows us to partition the simulation into two parts: the mechanically correct part and the electrically correct part.

In the first part, we apply DC voltage sources of value $|V_m|_{rms}$ ($=\frac{1}{\sqrt{2}}$ V) to the modulation voltage terminals, perform an AC sweep of the acceleration input in *y*-direction, and observe the mechanical displacement in *y*-direction at `wire_y` (as shown in *Fig. IV.4(a)*). The low frequency displacement is the steady state mechanical displacement of the plate-mass for 1G acceleration in *y*-direction. The pre-amplifier circuit is also reduced to just its equivalent input gate capacitance for this simulation.

This displacement is then applied as a DC position/displacement source to the comb-drive (which is equivalent to producing a constant change in the differential capacitances); the modulation voltages are now out-of-phase AC sources of magnitude $|V_m|$ (=1V); and an AC sweep is performed. The sample of AC response out-

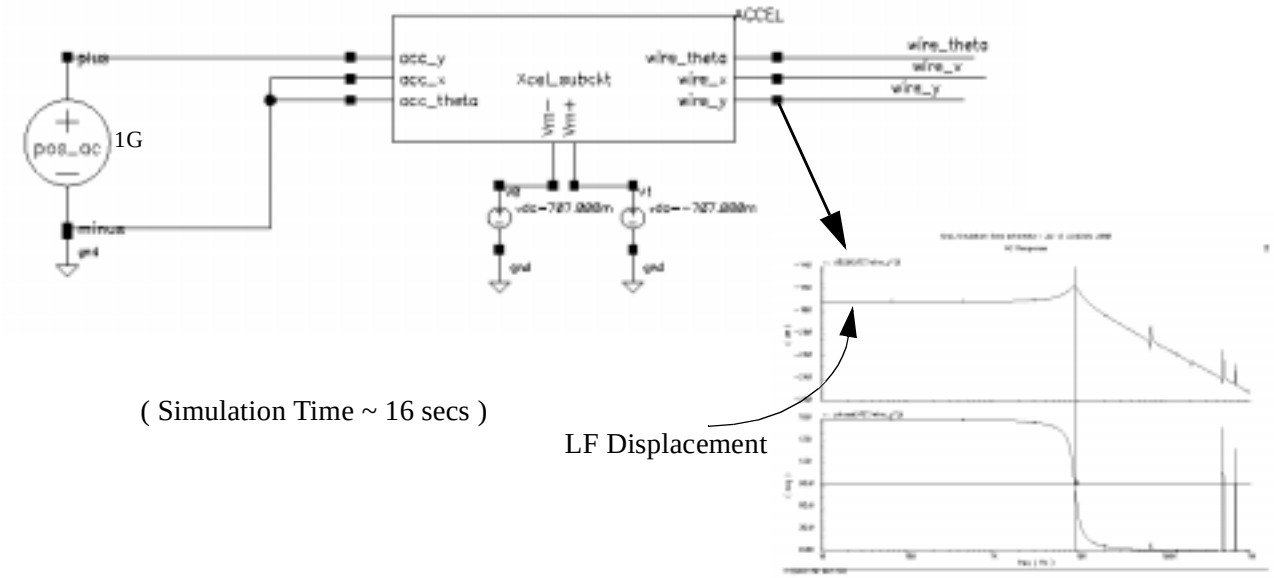


Fig. IV.4(a): Mechanically Correct part of the Sensitivity Calculation

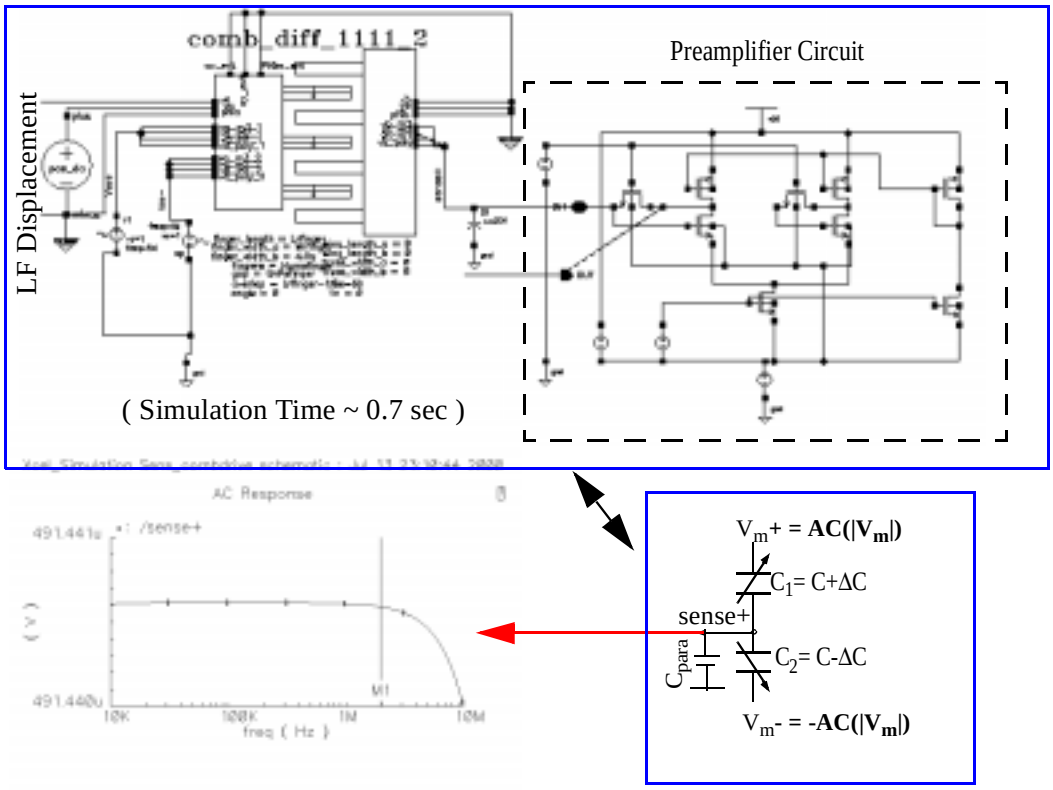


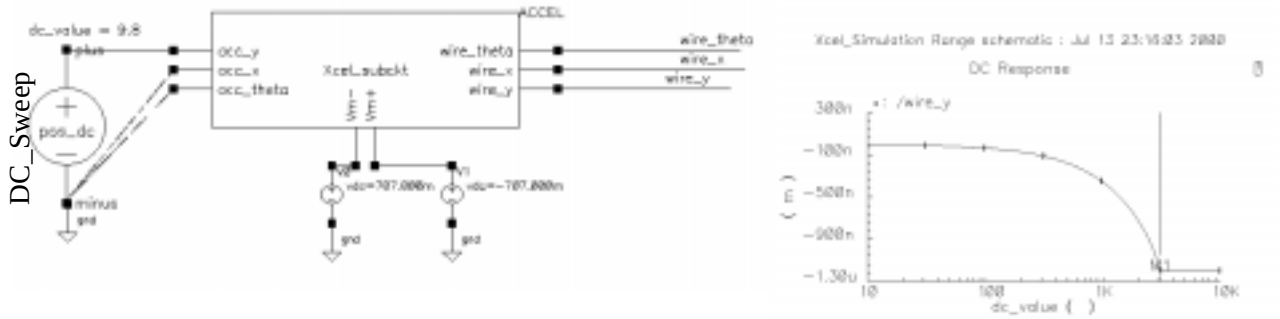
Fig. IV.4(b): Electrically Correct part of the Sensitivity Calculation

put at the *sense+* node at the modulation voltage frequency is the steady state output voltage observed for 1G acceleration, i.e., this sample is the sensitivity value (with < 2% error as compared to the transient analysis).

In addition to sensitivity, we can also measure the bandwidth, the resonant frequency in *y*-direction and the quality factor (*Q*) of the system from the same simulation as follows:

- bandwidth = frequency at which magnitude of *y*-displacement (in the mechanically correct part) crosses the -3dB point
- resonant frequency (w_y) = eigenmode frequency at which the phase of *y*-displacement (in the mechanically correct part) first crosses 90^o
- *Q* = (resonant frequency)/(bandwidth)

Thus, this harness performs only 2 AC simulations and is able to provide us four important characteristic measurement for the accelerometer behavior (with < 2% error). This harness is approximately 50x faster than the transient simulation harness.



(Simulation Time ~ 51 secs)

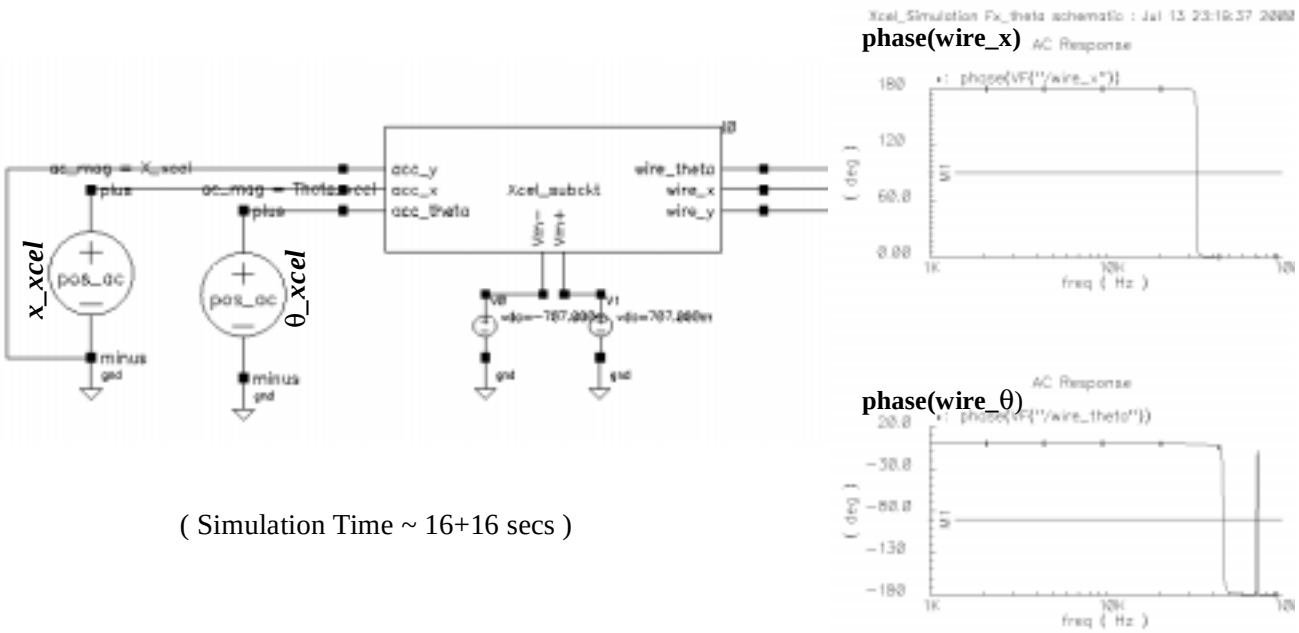
Fig. IV.5: Range Calculation

IV.2.2 Range Simulation

The range of an accelerometer is defined as the maximum acceleration detectable by the sensor. An accelerometer is functional till it hits the limit-stops whereby it can move no longer. So, an alternative definition of range is the minimum acceleration which makes the accelerometer hit the limit-stop. Therefore, ideally, one would like to perform a transient analysis with a step acceleration input (at `acc_y` terminal), and iteratively increase the magnitude of this step till the limit stop is hit by the overshoot peak of *y*-displacement.

A faster simulation harness uses RMS-valued DC sources at the modulation voltage terminals, which provides only the steady state displacement information; and perform a DC (logarithmic) sweep of the magnitude of the acceleration source at the `acc_y` terminal, as shown in *Fig. IV.5*. However, since, this harness reports the range with respect to the steady state displacement only, the limit-stop can actually be hit at a lower acceleration (as compared to the range reported by this harness) during the non-steady state part of sensor response.

This sweep is performed only from 1G to 1000G acceleration. If the limit stop is not hit with-in this band, the range of the accelerometer is assumed to be 1000G. This is necessary because during the course of optimization, some designs might actually have a range greater than 1000G, but, such cases are hardly of any practical interest, and are assumed to have a range of only 1000G.



(Simulation Time ~ 16+16 secs)

Fig. IV.6: Resonant Frequencies in *x* and θ Directions

IV.2.3 Resonant Frequencies of Minor Axes (*x*, θ)

The term resonant frequency is being used here to represent the eigenmode frequency, where the phase of displacement along a particular axis first crosses 90^o, as described in section IV.2. For measuring the resonant frequency in *x*-direction, an AC acceleration source is applied at the `acc_x` terminal (set *x_xcel*=1G, θ*_xcel*=0 in the schematic), and a frequency sweep is performed. The frequency at which the phase of *x*-displacement first cross 90^o is the eigenmode frequency of the system in *x*-direction. Similarly, for the eigenmode frequency in θ-direction, an AC acceleration source is applied at `acc_theta` terminal (set *x_xcel*=0, θ*_xcel*=1G in the schematic), and the frequency where phase of θ-displacement first crosses 90^o, is the resonant frequency in θ-direction.

IV.2.4 Mechanical/Electrical Spring Constants

The spring constant of a system in a particular direction is defined as the force per unit displacement. In order to measure the spring constant of the accelerometer system (in *y*-direction), a AC force source (with magnitude F_y) is applied at the `wire_y` terminal, as shown in *Fig. IV.7*. The low frequency displacement in *y*-

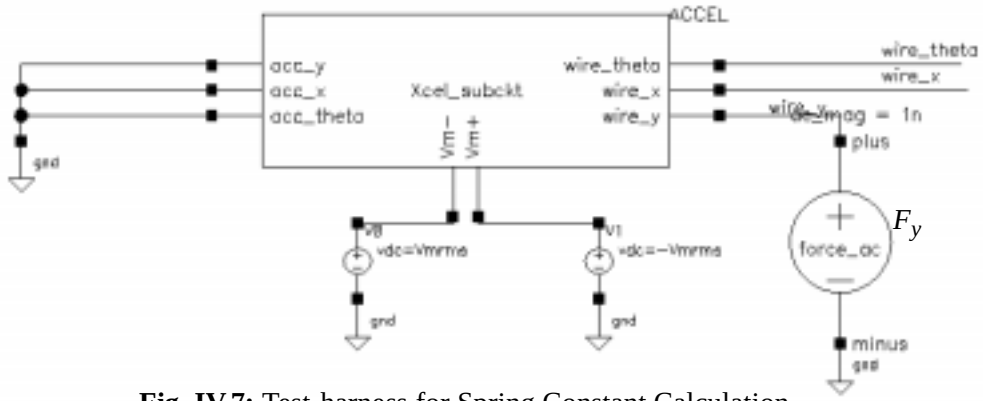


Fig. IV.7: Test-harness for Spring Constant Calculation

direction is measured at `wire_y,` and the ratio of the applied force magnitude (F_y) and this displacement is the spring constant (K_y). If RMS voltages (DC) are applied to the modulation voltage terminals (schematic design variable $V_m^{rms} = 1/\sqrt{2}$), there is spring softening due to these voltages, and the spring constant obtained is the effective spring constant ($K_y^{eff} = K_y^{mech} - K_y^{elec}$). We can ground the modulation voltages ($V_m^{rms} = 0$) to remove the electrical spring softening, and obtain the value of mechanical spring constant (K_y^{mech}).

IV.2.5 Cross-axis Sensitivity

The cross-axis sensitivity of an accelerometer is defined as the output voltage produced per unit acceleration applied along a minor axis (in this case, *y* is the primary axis for sensing, and *x* is the minor axis). Alternatively, it can be defined as the ratio of displacements along the major and minor axes for an acceleration applied along the minor axis, since, an accelerometer actually produces a voltage proportional to these displacements only.

Cross-axis sensitivity can be systematic as well as random in nature. Due to complete in-plane symmetry for this accelerometer topology, there is no systematic cross-axis sensitivity. So, random manufacturing variations along the die are the primary cause of cross-axis sensitivity. Therefore, in order to simulate cross-axis sensitivity of an accelerometer, one can introduce some manufacturing variations (like change in beam width

across the die) in the accelerometer schematic, apply an AC acceleration source in *x* direction, and measure the low frequency displacements in *x* and *y* directions for the modified schematic. The ratio of displacement in *y* direction to the displacement in *x* direction is the cross-axis sensitivity of the accelerometer.

However, for the given topology of symmetric serpentine spring being used in this accelerometer design, the cross-axis sensitivity is usually very low ($< 10^{-3}$) even for reasonable process variations along the die [16]. Therefore, for the sake of speeding up evaluation time, we have omitted this simulation. Nevertheless, we do obtain the nominal cross-axis sensitivity value (without introducing any process variations) using the displacement values from the simulations described in sections IV.2.3, which is usually $< 10^{-6}$.

IV.2.6 Noise

Noise is defined as the minimum acceleration detectable by an accelerometer. The mechanical as well as the electrical parts contribute to the system noise. At present, NODAS does not have a simple way of simulating mechanical noise. One can insert equivalent noise sources for each mass element in the netlist, and then try to simulate the minimum detectable acceleration of the system. However, in this implementation, this has been avoided. Instead, the information from the above simulations is used to obtain the mechanical damping coefficient value ($B_{damping}$) and the effective mass of the system in primary axis (M_y^{eff}) as follows:

$$M_y^{eff} = \frac{K_y^{eff}}{(w_y)^2} \qquad B_{damping} = \frac{w_y \cdot M_y^{eff}}{Q} \qquad \overline{Brownian\ Noise} = \frac{\sqrt{4 \cdot k_{Boltzmann} \cdot T_{absolute} \cdot B_{damping}}}{M_y^{eff}} \qquad \textbf{Eq. (IV.2)}$$

The sensitivity simulation in section IV.2.1 provides the resonant frequency (ω_y) and quality factor (Q) values, while the spring constant (K_y^{eff}) is obtained from the simulation in section IV.2.4. After obtaining the Brownian mechanical noise value, the system noise is calculated using Eq. III.3(a), where the noise floor of the read-out electrical circuit has already been measured to be 10nV/$\sqrt{Hz}$ at 2MHz frequency.

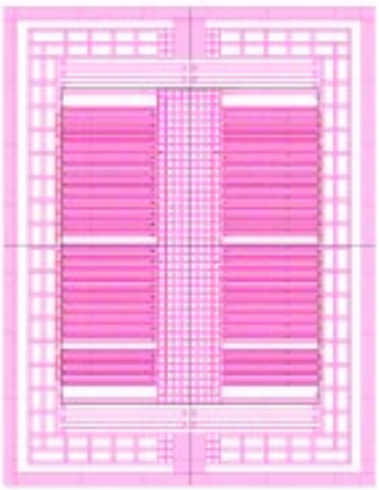
All the above-mentioned test-harnesses together constitute the evaluation suite for a candidate solution for simulation-based synthesis of a CMOS accelerometer.

IV.3 Synthesis Results

The measurements on the simulations described in the previous section are performed using scripts written in OCEAN. Sample scripts for these evaluations are listed in *Appendix III*. Using Distributed PRSA, all these simulations are distributed across a network of computers for faster evaluation. In this section, we will discuss some preliminary synthesis results obtained using a prototype tool implementing the simulation-based synthesis strategy for a CMOS accelerometer. The details of these synthesized designs are presented in *Appendix IV.*

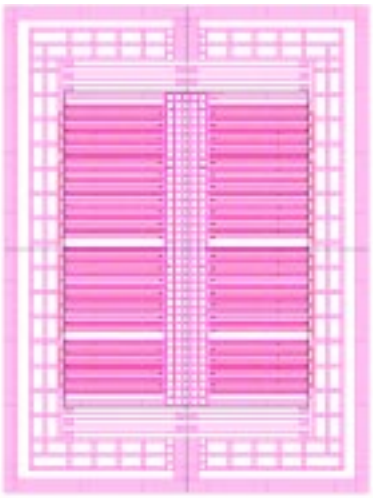
Similar to the equation-based synthesis case, the optimizations in this approach were performed for a common specification of range 25G, bandwidth 1kHz, cross-axis sensitivity 1% and maximum allowed area of 270 X 500 $(\mu m)^2$. Additionally, the preamplifier circuit is as shown in *Fig. IV.2(b)*, and has an equivalent input capacitance of 149fF, and an approximate noise floor of 10nV/$\sqrt{Hz}$.

The synthesis run with an objective to maximize the sensitivity of the accelerometer results in the design shown in *Fig. IV.8*. This design has a sensitivity of 1.91mV/G with a noise floor of 100μG/$\sqrt{Hz}$, while the initial sensitivity specification was 0.5mV/G. Also, the area of this design is 297 X 499 $(\mu m)^2$, while the initial constraint on area was 270 X 500 $(\mu m)^2$. This highlights another key aspect of unconstrained optimization formulation, that no constraint is a *hard* constraint. This means that in an unconstrained optimization formula-



Gap=1.5μm K_y=0.66N/m ω_y=5.55kHz

Fig. IV.8: Sensitivity Optimization with Simulation-based Synthesis



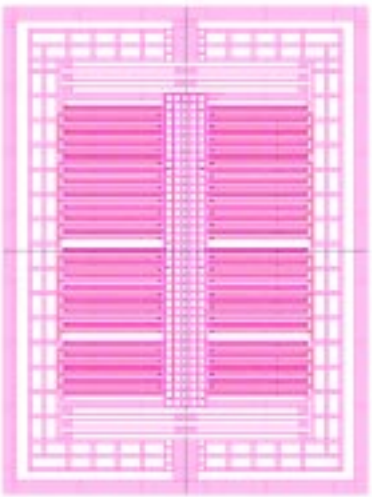
Gap=1.92μm K_y=1.29N/m ω_y=8.15kHz

Fig. IV.9: Noise Optimization with Simulation-based Synthesis

tion, we can reach a solution that does not meet a constraint within the specified limits but, is still the minimum of the weighted cost function. Therefore, in this run, the *x*-dimension constraint of 270μm has been exceeded by the optimization in order to maximize the sensitivity. Otherwise, the conclusions to be drawn from this optimization are similar to those from the equation-based synthesis: that sensitivity maximization gains by minimizing the finger gaps, increasing the finger length, and reducing the resonant frequency (in *y*-direction).

The optimization with an objective to minimize accelerometer noise results in the design shown in *Fig. IV.9*. This design, similar to the design optimized using equation-based synthesis, uses larger finger gaps in order to reduce the squeeze film damping, and consequently to reduce the Brownian mechanical noise. This design however, has a system noise floor of 73μG/$\sqrt{Hz}$, while equation-based synthesis produced a design with noise floor of 50μG/$\sqrt{Hz}$ (and the manual design has a noise floor of 57μG/$\sqrt{Hz}$ with a relaxed constraint for ω_{finger}). This suggests the fact that the unconstrained weighted cost function used in simulation-based synthesis is still not optimal, and causes the search algorithm to converge on a sub-optimal solution.

The objective to minimize the accelerometer area results in the design shown in *Fig. IV.10*. This design has an area of 264 X 432 $(\mu m)^2$, which is approximately 15.5% improvement over the initial area specifications of 270 X 500 $(\mu m)^2$. This optimization, similar to the equation-based synthesis, tries to meet the sensitivity constraint by using a small finger-gap (hence, using a large sense capacitance) and by using a flexible spring. However, in simulation-based synthesis, we cannot vary the number of truss beams, as this is equivalent to changing the spring topology, and requires a change in the simulation netlists (see section II.3.2 on spring design). Therefore, simulation-based synthesis is not able to increase the flexibility of the spring (as is done by equation-based synthesis by using 7 instead of 5 truss



Gap=1.5μm K_y=1.19N/m ω_y=8.8kHz

Fig. IV.10: Area Optimization with Simulation-based Synthesis

beams in the spring). The spring constant using simulation-based synthesis is 1.19N/m, while equation-based synthesis can achieve a spring constant of 0.63N/m by using more truss beams in the serpentine spring. Consequently, the area optimization using simulation-based synthesis is inferior as compared to equation-based synthesis. However, this shortcoming of simulation-based synthesis can be fixed if we use a parametric macromodel for the serpentine spring [26], where the number of truss beams will just be a parameter to the spring macromodel, and changing the number of truss beams would not introduce/remove any element in the simulation netlist.

IV.4 Summary

A prototype tool implementing the simulation-based synthesis approach for CMOS accelerometer synthesis has been created. The encapsulation of NODAS has been done in the OCEAN environment in Cadence Design System's IC Design framework. OCEAN requires the simulation data to be disk-written for processing, which causes the simulation-based evaluation time to be very long. It takes at least overnight to finish a synthesis runs (with ~ 3000 evaluations). This long computation time also makes it hard to perform a trade-off analysis (as was presented for equation-based synthesis in section III.3.3).

Preliminary synthesis results from this prototype tool again suggest significant optimization/improvement possibilities over manual design. However, the results are certainly inferior (sub-optimal) to those obtained using equation-based synthesis. This suggests that the unconstrained cost function created for simulation-based synthesis is not yet optimal, and needs to be tuned.

Another drawback of using a stochastic search algorithm is its random nature. The inherent randomness involved with stochastic search algorithms results in different solutions for different runs of the same optimization problem. PRSA tries to mitigate this problem by using a number of parallel runs and by frequently exchanging information between these runs but, it cannot get rid of the randomness completely.

Nevertheless, the simulation-based synthesis approach does provide much more flexibility and accuracy to the designer as compared to the equation-based synthesis. In this approach, the designer doesn't need to derive complex lumped parameter equations for the design that he/she wants to optimize, which makes simulation-

based synthesis much more easier to use. The evaluation scripts (in OCEAN) are generic for a particular system design, and can be used even if the designer make small changes in the system topology.

In conclusion, we believe that simulation-based synthesis will be the preferred method of design optimization for MEMS design, primarily due to its generic nature, and ease of use. However, as of now, a lot of work remains to be done to make NODAS-based evaluation faster, and to devise a systematic strategy to convert a constrained optimization problem in to an unconstrained one [25].

V. Verification of Synthesis

This chapter reports our attempts on verifying the results predicted by the prototype tools (implementing equation-based synthesis and simulation-based synthesis) for synthesis of a CMOS accelerometer. The verification of synthesis results has been attempted in two phases:

In the first phase, we have tried to validate the lumped parameter equations for a CMOS accelerometer by comparing the equation-based evaluation results to the evaluation results using NODAS simulation, while NODAS simulation results have previously been verified [27].

And, in the second phase, the designs synthesized using equation-based synthesis were fabricated, and the experimentally measured results from these designs are reported.

V.1 Lumped Parameter Model Validation for Equation-based Synthesis

In order to validate the lumped parameter analytical equations, the behavior of four accelerometer designs as predicted by these equations is compared with that predicted by nodal simulation in NODAS (Table 5). The designs being used as benchmarks in this comparison are: `Manual`, the manual design in [11]; `Area`, the design optimized for minimal area; `Noise`, the design optimized for minimal noise; `Range`, the design optimized for maximal range, and `Sens`, the design optimized for maximal sensitivity using equation-based synthesis. As these designs were synthesized using an intermediate version of the equation-based synthesis prototype tool, they differ from the optimal designs discussed in section III.3. The details of these designs are listed in *Appendix V.* As the table shows, most of the behavior characteristics predicted by equations match the simulated behavior to within ±10%.

The discrepancy in the calculation of resonant frequency in θ-mode, F_{theta}, is primarily because of a bug in the moment of inertia calculation for plate-mass (with holes) in NODAS. The plate-mass model currently being used in NODAS doesn't account for the presence of holes inside the plate-mass for moment of inertia calculation. Consequently, NODAS calculates larger than actual moment of inertia, resulting in lower reso-

nant frequency. This effect is predominantly visible in designs: *Manual* and *Range*, which have larger plate-masses.

Table 5: NODAS Simulation *vs.* Lumped Parameter Equations Comparison

design	F_y (KHz)	*BW* (KHz)	*Q*	*Sensitivity** (mV/G)	*Range* (G)	K_y (N/m)	K_x (N/m)	F_x (KHz)	F_θ (KHz)	*Crosen* ($x10^{-9}$)	*Noise* ($\mu G/\sqrt{Hz}$)	M_y (µg)
Manual-Sim	8.783	13.55	5.62	0.535	343	1.761	33.7	33.633	47.47	5	57.4	0.578
Manual-Eqn	8.507	13.15	5.75	0.569	333	1.606	37.684	39.84	87.41	0	56.2	0.562
*Area*Sim	9.69	14.83	3.79	0.325	533	1.07	19.08	33.19	43.7	2.6	103	0.29
AreaEqn	9.27	14.22	3.78	0.351	414	0.949	18.538	38.58	52.18	0	101	0.28
*Noise*Sim	7.63	11.8	6.6	0.38	387	1.09	22.29	29.7	45.79	0.74	57.8	0.473
*Noise*Eqn	7.354	11.37	5.95	0.408	327	0.965	21.96	33.8	62.74	0	59.7	0.452
*Range*Sim	8.62	13.29	5.44	0.63	298	1.756	35.08	33.36	47.97	0.12	56	0.598
*Range*Eqn	8.12	12.54	5.34	0.706	262	1.514	34.915	37.67	87.22	0	55.1	0.582
*Sense*Sim	6.99	10.24	1.84	1.18	182	1.04	26.4	29.98	43.02	0.2	87.8	0.541
*Sense*Eqn	6.64	9.706	1.73	1.281	158	0.918	25.598	33.99	58.19	0	89	0.53

*. Sensitivity is measured with a modulation voltage of amplitude 1V and frequency 2MHz

The discrepancy in calculation of resonant frequency in *x*-direction (cross-axis direction), F_x, is primarily due to difference in the effective mass models (in *x*-direction) being used in NODAS and lumped parameter equations. The spring constants from equations and NODAS (K_x) are reasonably close but, the effective mass calculation causes the resonant frequencies to differ. The effective mass of the spring in *x*-direction, as reported by NODAS is higher than the total mass of the spring. This suggests that the serpentine spring does not resemble its static mode in *x*-direction, and improved lumped models need to be developed for estimating its resonant frequency in *x*-direction.

The range estimation using lumped parameter equations assumes a linear behavior for the sensor for its complete range. But, the accelerometer no longer behaves linearly at high accelerations close to 100G. This causes a mismatch between the range predicted by the equations and non-linear simulation.

These designs were fabricated at the AMS foundry, and the experimental measurements from the fabricated devices are listed in the next section.

V.2 Experimental Verification

The fabricated accelerometers can be tested in two possible ways:

— optical test: in this test, the accelerometers are driven using the self test force (generated by applying voltages to the force comb-drives), and the mechanical displacement of the mass is optically measured under the MicroVision system [28].

— electrical test: in this test, the accelerometers are driven either by mounting them on a vibration-table or by using the self test force; the electrical signal of capacitive sensor is then amplified using an on-chip pre-amplifier, and the output signal is observed/measured using a spectrum analyzer (HP 4395A).

The optical setup can therefore, verify just the mechanical response of the spring-plate-mass system in the accelerometer. The electrical test setup, on the other hand, verifies the mechanical as well as the electrical behavior of the accelerometer. For the electrical setup, some problems were encountered with mounting the accelerometer on the vibration-table. The response of the mounting assembly was interfering with the observed accelerometer behavior at certain frequencies. Consequently, the self test force was used to obtain the frequency response of the accelerometers in the optical as well as the electrical test setup.

A few points are worth mentioning here. First, this was the first attempt at fabricating a CMOS MEMS device at the AMS 0.6μm foundry, while all the estimates of the mechanical properties used in the synthesis formulation were based on the MEMS devices fabricated at HP 0.5μm foundry. This can potentially induce some difference in the expected and the observed behavior of the accelerometers. Also, residual stress gradients in this process led to unexpected curling (lateral as well as out-of-plane) for the AMS devices causing failure and/or loss of sense capacitance. Thus, further research efforts are required to obtain better estimates of mechanical properties of devices fabricated with AMS.

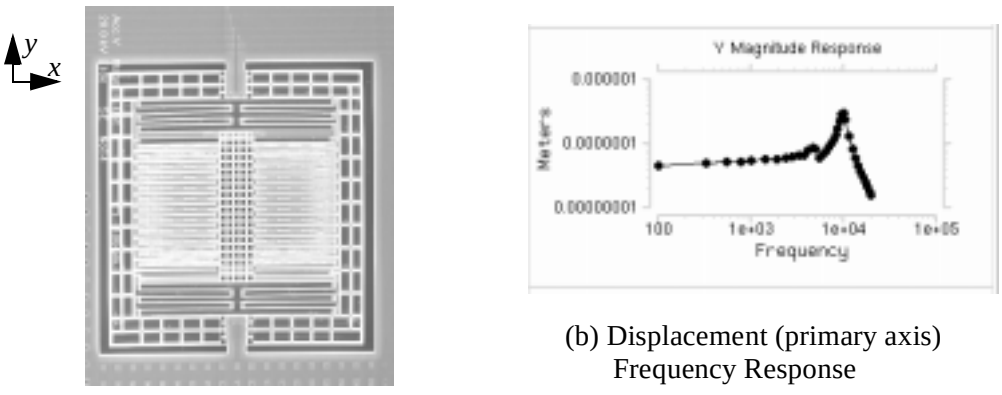
Secondly, an on-chip pre-amplifier is required to amplify the differential capacitive sense signal from the accelerometer. However, it is impossible to estimate the gain of the pre-amplifier unless it is fabricated and

tested separately. Simulations can provide an estimate of the pre-amplifier gain. However, due to process variations, the gain of the pre-amplifier varies significantly. Consequently, the measured output voltages cannot be compared with the expected/simulated sensitivity values. In future, one should either put the pre-amplifier in a feedback loop to have a good control over pre-amplifier gain, or a separate pre-amplifier must be fabricated and tested in order to estimate its gain.

A frequency of 2MHz was used for modulation voltages in the electrical test setup. For sensitivity measurements, the amplitude of modulation voltage was set to 1V; while for noise measurements, an amplitude of 2.5V (5V peak-to-peak) was used. A 1G sinusoidal acceleration (at 200Hz) was applied using the vibration table for both these measurements. Around 2MHz, the spectrum analyzer has an inherent noise floor of ~100nV/rtHz (~ –20dBμV/rtHz). The noise floor observed on most of the devices (~ –12dBμV/rtHz) was very close to this inherent noise floor, which may cause significant errors in the measurement of noise floor of the accelerometers. Further experiments need to be performed where the sense signal is demodulated and amplified before it is measured using the spectrum analyzer. Due to this lack of precision in noise measurement, it is not possible to compare the expected noise values to the measured ones.

***Area* Accelerometer**

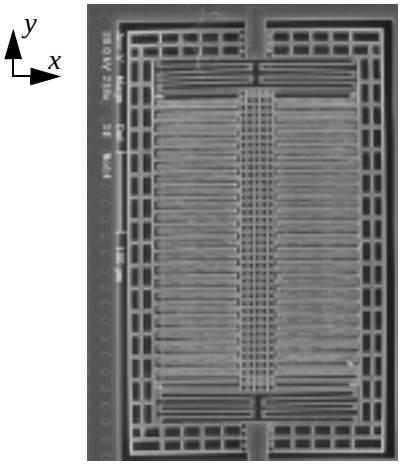
A SEM photograph of the device *Area* is shown in *Fig. V.2*. The resonant frequency of this accelerometer in *y*-direction (F_y) was measured to be 9.15kHz using the optical test setup, which is well within 10% of the



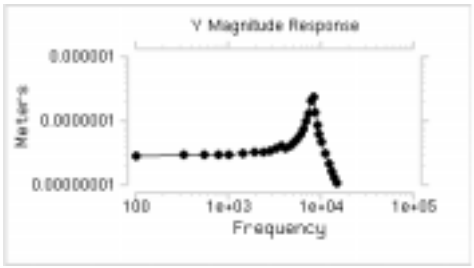
(a) SEM

(b) Displacement (primary axis) Frequency Response

Fig. V.2: *Area* Accelerometer



(a) SEM



(b) Displacement (primary axis) Frequency Response

Fig. V.3: `Noise` Accelerometer

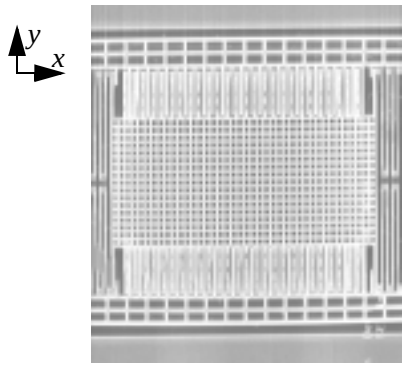
expected value (see Table 5). Unfortunately, the only chip with the working `Area` device got damaged so that the two modulation voltage terminals were shorted out. Consequently, the electrical behavior of the `Area` device could not be tested.

`Noise` *Accelerometer*

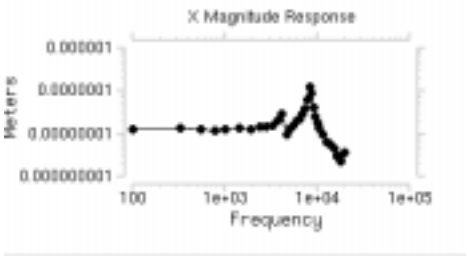
Fig. V.3 shows a SEM photograph of the device `Noise`, as fabricated at the AMS foundry. Optically, as well as electrically, the resonant frequency in *y*-direction (F_y) was verified to be 7.8kHz, while the expected resonant frequency was ~7.5kHz. The output voltage (*after pre-amplification*) was observed to be 243μV/G. The device has a range greater than 13G, which is the maximum allowed acceleration on the vibration table. The accelerometer had a maximal signal to noise ratio (SNR) of ~43.7dBμV/rtHz for a 1G acceleration input, as observed using the spectrum analyzer. For a 2G acceleration input, the output signal increases by 5.8dBμV. Therefore, the output voltage approximately varies as 5.8dBμV for every 2X change in the acceleration input. This translates to an input referred noise floor of ~5.4mG/rtHz for the device. However, as suggested earlier,

$$Noise \approx 2^{-\left\langle \frac{43.7}{5.8} \right\rangle} = 5.4 \frac{mG}{\sqrt{Hz}}$$

this measurement could be erroneous due to the inherent noise floor of the spectrum analyzer. The ambient noise of the test setup also contributes to the observed noise floor.



(a) SEM



(b) Displacement (primary axis) Frequency Response

Fig. V.4: *Range* Accelerometer

Range Accelerometer

The resonant frequency (F_y) of this accelerometer was measured to be 8kHz optically as well as electrically. The *pre-amplified* output voltage was observed to be 85.2μV/G. The range of this accelerometer is greater than the maximum allowed acceleration on the vibration table (13G). *Fig. V.5* shows a plot of the dynamic linearity of this device. However, (possible) non-linearities of the pre-amplifier circuit are also coupled in this plot of the *amplified* output voltage *vs.* the input acceleration. The observed maximal SNR for 1G acceleration was ~36.4dBμV/rtHz. The output voltage varied by 6dBμV for a 2X change in the input acceleration. Thus, the SNR of 36.4dBμV is equivalent to an input referred noise floor of 14.9mG/rtHz. Again, the

$$Noise \approx 2^{-\left\langle \frac{36.4}{6.0} \right\rangle} = 14.9 \frac{mG}{\sqrt{Hz}}$$

noise floor of the spectrum analyzer and ambient noise could be the reasons behind this discrepancy.

Sense Accelerometer

This design had very thin spring beams, minimum comb-finger gaps, and very long comb-fingers for maximal sensitivity (refer *Appendix V*). These features coupled with the unexpected curling in the AMS process led to failure of all the *Sense* accelerometers that were released. Consequently, the behavior of the *Sense* accelerometer design could not be tested.

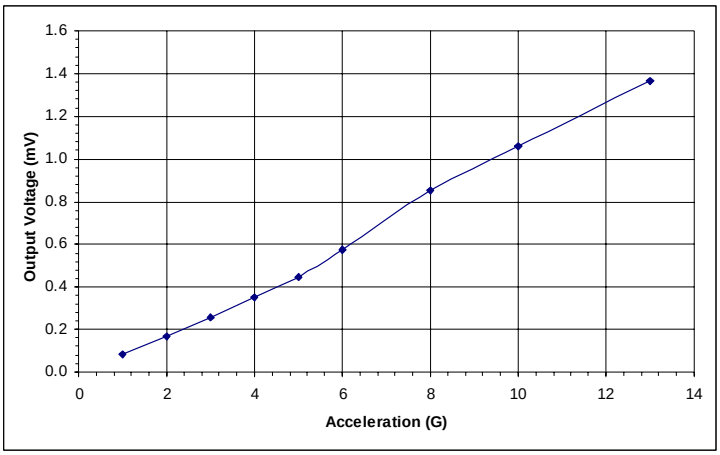


Fig. V.5: Dynamic Linearity of *Range* Accelerometer at 200Hz

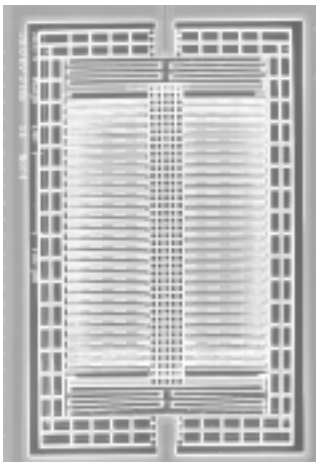


Fig. V.6: *Sense* Accelerometer

V.3 Summary

The comparison between evaluation of designs using lumped parameter equations with that using nodal simulation in NODAS proves the validity of the equations used in equation-based synthesis of the CMOS accelerometer. The fact that at least the mechanical part of the experimental results matches the synthesis predicted behavior within ±10% proves that the use of lumped parametric equations/nodal simulations is a reasonable approximation to finite element methods for analyzing accelerometer behavior. However, as is clear from the electrical verification part of the experimental results, curling of CMOS MEMS devices alters the transducer behavior significantly, and needs to be understood/quantified better in order to improve synthesis. The synthesis results should be used with caution in the absence of appropriate curling information.

Also, no conclusion could be drawn regarding the accuracy of synthesis results. Due to lack of information about pre-amplifier gain, the sensitivity predictions could not be verified; while due to lack of precision/ shielding in the experimental setup, the noise predictions could not be compared with the measured results. Any experiments in future must be planned very carefully to take care of the above-mentioned problems.

Thus, the synthesis methodology has been shown to work for a CMOS accelerometer. It can actually automate the translation of high level behavioral specifications into working silicon, but, nothing can be concluded about the accuracy of synthesis results as of now.

VI. Conclusions & Future Work

This work demonstrates that the synthesis paradigm actually works for a CMOS accelerometer. It can successfully translate high level behavioral specifications into a working piece of silicon with considerably less human effort as compared to a manual design. The synthesized devices when fabricated using a well characterized process match their predicted behavior reasonably well. The synthesis results also suggest significant optimization possibilities in the design of a CMOS accelerometer over manual design. The trade-off analyses possible using a synthesis tool provide useful insights into system behavior, and lead to system design optimization in addition to the cell being synthesized. The simulation-based synthesis framework makes use of a library of MEMS elements to synthesize a CMOS accelerometer. Thus, in this framework, there is no need to generate complicated lumped parameter equations for a new micro-electro-mechanical system, providing new opportunities to synthesize other frequently used MEMS cells as well.

Future Work

The experimental verification of synthesis results requires further efforts, especially, for the electrical behavior of the accelerometer. Simulation-based synthesis, though flexible, more accurate and easy to use, is not yet in its optimal state. The encapsulation of NODAS using OCEAN scripts is unnecessarily and excessively time consuming (because of disk-operations involved), and needs to be improved upon. Also, we need to implement a systematic strategy to find the optimal weights for the constraints and objective functions, in order to formulate an un-constrained optimization problem for simulation-based synthesis.

This synthesis framework can easily be extended to synthesis for other frequently used MEMS cells like, resonators, gyroscopes, etc. For this purpose, the process of identifying the design variables and geometric constraints for a system can also be automated. Further work also needs to be done to optimize device/system design in the face of manufacturing variations. Another possible direction of research is mixed-domain synthesis, *i.e.*, to optimize the micro-electro-mechanical sensor as well as its circuits together, as a system.

References

[1] I.E. Grossmann and M.M. Daichendt, "*New Trends in Optimization-based Approaches to Process Synthesis*," Computers & Chemical Engineering, vol. 20, no. 6-7, pp. 665-683.

[2] S.J. Fenves, "*Computer-aided Design in Civil Engineering*," Proceedings of the IEEE, vol. 69, no. 10, pp.1240-1248.

[3] R. Harjani, R.A. Rutenbar and L.R. Carley, "*OASYS: A Framework for Analog Circuit Synthesis*," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 8, no. 12, pp. 1247-66.

[4] L.R. Carley, G.G.E. Gielen, R.A. Rutenbar and W.M.C. Sansen, "*Synthesis Tools for Mixed-Signal ICs: Progress on Frontend and Backend Strategies*," Proceedings of Design Automation Conference, 1996.

[5] T. Mukherjee, S. Iyer and G.K. Fedder, "*Optimization-based Synthesis Of Microresonators*," Sensors and Actuators A, October 1998, pp. 118-127.

[6] T. Mukherjee, Y. Zhou and G.K. Fedder, "*Automated Optimal Synthesis of Microaccelerometers*," MEMS 99, Orlando, Florida, pp. 326-331.

[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] M. Krasnicki, R. Phelps, R.A. Rutenbar and L.R. Carley, "*MAELSTORM: Efficient Simulation-Based Synthesis for Custom Analog Cells*," Proceedings DAC, 1999.

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

[10] G. Zhang, H. Xie, L.E. de Rosset and G.K. Fedder, "*A Lateral Capacitive CMOS Accelerometer with Structural Curl Compensation*," MEMS 99, Orlando, Florida, pp. 606-611.

[11] H. Luo, G.K. Fedder and L.R. Carley, "*A 1 mG Lateral CMOS-MEMS Accelerometer*," MEMS 2000, pp. 502-507.

[12] Analog Devices, "*ADXL150 — ±50G Accelerometer with 10mG Resolution*," Datasheet, 1997.

[13] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright, "*User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming*," Technical Report SOL 86-2, Stanford University, January 1986.

[14] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, "*Optimization by Simulated Annealing*," Science, vol. 220, no. 4598, 1983.

[15] M.S.C. Lu, Xu Zhu and G.K. Fedder, "*Mechanical Property Measurement of 0.5-μm CMOS Microstructures*" MRS 1998 Spring Meeting, San Francisco, CA, April 1998.

[16] S. Iyer, Y. Zhou and T. Mukherjee, "*Analytical Modeling of Cross-axis Coupling in Micromechanical Springs*," Modeling and Simulation of Microsystems, Puerto Rico, 1999, pp. 632-635.

[17] K. He *et al., "CMOS-MEMS Design Rules,"* MEMS Lab Technical Document, Carnegie Mellon University.

[18] P.E. Gill, W. Murray and M.H. Wright, "*Practical Optimization*," Academic Press, 1981.

[19] R. Fletcher, "*Practical Methods of Optimization*," Wiley, New York, 1987.

[20] W.H. Chang, "*Analytic IC Metal-line Capacitance Formulas*," IEEE Trans. Microwave Theory Techniques, pp. 608-611, September 1979.

[21] S. Vemuri, G.K. Fedder and T. Mukherjee, "*Low-Order Squeeze Film Model for Simulation of MEMS Devices*," Proceedings of MSM 2000, San Diego.

[22] M.A. Lemkin, "*Micro Accelerometer Design with Digital Feedback Control*," Ph.D. Thesis, EECS Department, University of California, Berkeley, 1997.

[23] W. Kuehnel, "*Modeling of Mechanical Behavior of a Differential Capacitor Acceleration Sensor*," Sensors and Actuators A, vol. A48, pp. 101-8, May 1995.

[24] S.W. Mahfoud and D.E. Goldberg, "*Parallel Recombinative Simulated Annealing: A Genetic Algorithm*," Parallel Computing, vol.21, 1995.

[25] E.S. Ochotta, "*Synthesis of High-Performance Analog Cells in ASTRX/OBLX*," Ph.D. Thesis, ECE Department, Carnegie Mellon University, 1994.

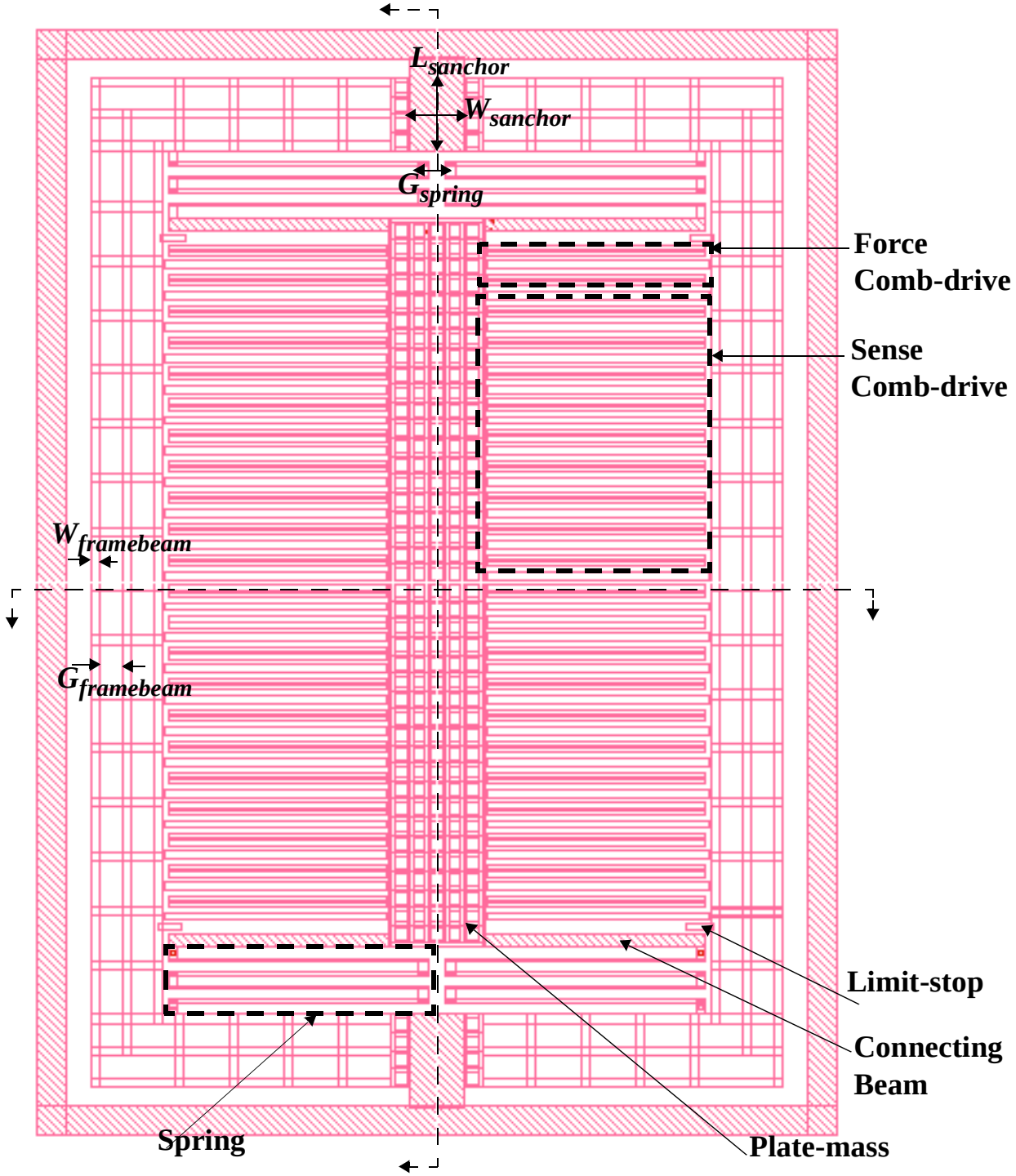
[26] G.K.Fedder, "*Simulation of Microelectromechanical Systems*," Ph.D. Thesis, EECS Department, University of California, Berkeley, 1994.

[27] Q. Jing, H. Luo, T. Mukherjee, L.R. Carley and G.K. Fedder, "*CMOS Micromechanical Bandpass Filter Design Using a Hierarchical MEMS Circuit Library*," MEMS 2000, Japan, pp. 187-192.

[28] D.M.Freeman, A.J.Aranyosi, M.J.Gordon and S.S. Hong, "*Multidimensional Motion Analysis of MEMS using Computer Microvision*," Technical Digest. Solid-State Sensor and Actuator Workshop, pp. xiv+382, 1998.

Appendices

I. CMOS Accelerometer Layout



This figure shows a typical CMOS accelerometer layout, the constituent elements of this layout, and some style variables (listed but not shown in chapter 2, table 2).

II. Equation-based Synthesis Results

Table 6: Design Variables for Designs Synthesized using Equation-based Synthesis

DesignVar	Noise Optimized Design	Sensitivity Optimized Design	Area Optimized Design	Manual Design
L_{beam} (μm)	126.1	126.1	125.26	122.4
W_{beam}(μm)	1.8	1.8	1.8	2.1
L_{truss}(μm)	7.7	6.3	6.3	6.6
W_{truss}(μm)	5	5	5	5.1
$Num_{trussbeams}$	5	7	7	5
L_{finger}(μm)	87.6	110.1	46.26	54.9
$W_{rotor\text{-}finger}$(μm)	5.7	5.7	5.65	5.7
G_{limit} (μm)	2	0.9	1.31	1.2
GAP_{finger}(μm)	2.311	1.5	1.61	1.5
$W_{strongbeam}$(μm)	10	5.15	1.83	3
L_{pmass}	35+4*	33+4*	21+4*	35+4*
W_{pmass}	6+4*	1+4*	15+4*	13+4*
$Num_{sfinger}$	10	11	7	10
$Num_{ffinger}$	1	1	1	3

*. 4 squares are used for routing in the plate-mass in *x* as well as *y* direction

NOTE: All the dimensions listed in the context of equation-based synthesis are center-to-center dimensions. In the evaluation of designs listed below, primary axis of the designs is termed *x* instead of *y*.

II.1 Evaluation of Noise-Optimized-Design using Lumped Parameter Models

mass(mxeff): 0.505923 (ug) mass(myeff): 0.542122 (ug) mass(Ithetaeff): 5.90283e-18
spring -- Kxeff: 0.931882 (N/m), Kxmech: 0.943855 (N/m), Kxelec: 0.0119724 (N/m)
spring(Ky): 16.9554 (N/m), spring(Ktheta): 6.8134e-07 (uN*um)
resonant frequency -- fx(mechanical): 6874.41 (Hz) fx(effective): 6830.67 (Hz)
resonant frequency(fy): 28146.8 (Hz) resonant frequency(ftheta): 54072.5 (Hz)
quality factor: 6.62464damping factor: 3.27763e-06 (N*S/m)
Couette Bx: 1.25661e-07 (N*S/m) Squeeze Bx: 3.15197e-06 (N*S/m)
Brownian noise floor: 46.9894 (ug/rtHz)
Parasitic cap due to circuit: 1.86e-13 comb sensing cap: 3.19917e-14
sensitivity: 0.57 (mv/g)
bandwidth: 10570.5 (Hz)

noise: 0.501577 (mg)
range: 343.254 (g)
cross-axis sensitivity: 1e-07 %

0 < acc_xsize = 423 < 500
0 < acc_ysize = 270 < 270
0 < uSpring_ysize = 268.2 < 270
0.3 < gsfinger - gxlimit = 0.3 < 20
4.5 < ltruss - wbeam = 5.9 < 20
-16.5 < wpmass/2 + lrfinger -lbeam1-wtruss-gylimit-wmstop/2 =-16.5 < 250
5 < gap between uspring and finger = 5.09889 < 250
0 < kxelec/kxmech = 0.0126845 < 0.8
0.57 (mv/g) < sensitivity = 0.57 (mv/g) < 0.627 (mv/g)
0 (mg) < noise = 0.501577 (mg) < 1 (mg)
25 (g) < range = 343.254 (g) < 1000 (g)
0 < cross-axis sensitivity = 1e-09 < 0.05
100 (Hz) < bandwidth = 10570.5 (Hz) < 100000 (Hz)
0.2 < self_test_force/(mxeff*Range*9.8) = 0.312656
0 < 2.0*lbeam2/lcrit = 0.182323 < 1
0.666667 < (fy - fx)/fy = 0.75732 < 1
0.666667 < (ftheta - fx)/ftheta = 0.873676 < 1
0 < 1.5*resonant_frequency_X_comb-finger/(Modulation Voltage Frequency) = 0.724236 < 1
0 < (Kx_Stiff_beam - 10*Kxeff_spring)/Kx_stiff_beam = 0.9809 < 1
0 < x-dir-size/MaxSize = 1 < 1

II.2 Evaluation of Sensitivity-Optimized-Design using Lumped Parameter Models

mass(mxeff): 0.474415 (ug) mass(myeff): 0.526123 (ug) mass(Ithetaeff): 6.11685e-18
spring -- Kxeff: 0.570032 (N/m), Kxmech: 0.630812 (N/m), Kxelec: 0.0607797 (N/m)
spring(Ky): 7.25202 (N/m), spring(Ktheta): 2.83395e-07 (uN*um)
resonant frequency -- fx(mechanical): 5803.57 (Hz) fx(effective): 5516.9 (Hz)
resonant frequency(fy): 18685.8 (Hz) resonant frequency(ftheta): 34257.6 (Hz)
quality factor: 1.5805damping factor: 1.04048e-05 (N*S/m)
Couette Bx: 1.20803e-07 (N*S/m) Squeeze Bx: 1.0284e-05 (N*S/m)
Brownian noise floor: 89.2817 (ug/rtHz)
Parasitic cap due to circuit: 1.86e-13 comb sensing cap: 6.83979e-14
sensitivity: 1.99615 (mv/g)
bandwidth: 7957.28 (Hz)
noise: 0.894221 (mg)
range: 107.749 (g)
cross-axis sensitivity: 1e-07 %

0 < acc_xsize = 405 < 500
0 < acc_ysize = 270 < 270
0 < uSpring_ysize = 268.2 < 270
0.3 < gsfinger - gxlimit = 0.6 < 20
4.5 < ltruss - wbeam = 4.5 < 20
-16.5 < wpmass/2 + lrfinger -lbeam1-wtruss-gylimit-wmstop/2 =-16.5 < 250
5 < gap between uspring and finger = 6.7 < 250
0 < kxelec/kxmech = 0.0963516 < 0.8
0.5 (mv/g) < sensitivity = 1.99615 (mv/g) < 1000 (mv/g)
0 (mg) < noise = 0.894221 (mg) < 1 (mg)
10 (g) < range = 107.749 (g) < 1000 (g)

0 < cross-axis sensitivity = 1e-09 < 0.05
100 (Hz) < bandwidth = 7957.28 (Hz) < 100000 (Hz)
0.2 < self_test_force/(mxeff*Range*9.8) = 3.90078
0 < 2.0*lbeam2/lcrit = 0.280648 < 1
0.666667 < (fy - fx)/fy = 0.704754 < 1
0.666667 < (ftheta - fx)/ftheta = 0.838958 < 1
0 < 1.5*resonant_frequency_X_comb-finger/(Modulation Voltage Frequency) = 0.458473 < 1
0 < (Kx_Stiff_beam - 10*Kxeff_spring)/Kx_stiff_beam = 0.830179 < 1
0 < x-dir-size/MaxSize = 0.9864 < 1

II.3 Evaluation of Area-Optimized-Design using Lumped Parameter Models

mass(mxeff): 0.376654 (ug) mass(myeff): 0.428072 (ug) mass(Ithetaeff): 2.61118e-18
spring -- Kxeff: 0.63057 (N/m), Kxmech: 0.643606 (N/m), Kxelec: 0.0130359 (N/m)
spring(Ky): 7.30067 (N/m), spring(Ktheta): 1.58146e-07 (uN*um)
resonant frequency -- fx(mechanical): 6579.05 (Hz) fx(effective): 6512.08 (Hz)
resonant frequency(fy): 20784.9 (Hz) resonant frequency(ftheta): 39168.4 (Hz)
quality factor: 6.36846damping factor: 2.41994e-06 (N*S/m)
Couette Bx: 1.03005e-07 (N*S/m) Squeeze Bx: 2.31693e-06 (N*S/m)
Brownian noise floor: 54.233 (ug/rtHz)
Parasitic cap due to circuit: 1.86e-13 comb sensing cap: 1.67748e-14
sensitivity: 0.57 (mv/g)
bandwidth: 10074.2 (Hz)
noise: 0.57 (mg)
range: 207.343 (g)
cross-axis sensitivity: 1e-07 %

0 < acc_xsize = 297 < 500
0 < acc_ysize = 268.313 < 270
0 < uSpring_ysize = 266.513 < 270
0.3 < gsfinger - gxlimit = 0.3 < 20
4.5 < ltruss - wbeam = 4.5 < 20
-16.5 < wpmass/2 + lrfinger -lbeam1-wtruss-gylimit-wmstop/2 =-16.5 < 250
5 < gap between uspring and finger = 5 < 250
0 < kxelec/kxmech = 0.0202545 < 0.8
0.57 (mv/g) < sensitivity = 0.57 (mv/g) < 0.627 (mv/g)
0 (mg) < noise = 0.57 (mg) < 0.57 (mg)
25 (g) < range = 207.343 (g) < 1000 (g)
0 < cross-axis sensitivity = 1e-09 < 0.05
100 (Hz) < bandwidth = 10074.2 (Hz) < 100000 (Hz)
0.2 < self_test_force/(mxeff*Range*9.8) = 0.453317
0 < 2.0*lbeam2/lcrit = 1 < 1
0.666667 < (fy - fx)/fy = 0.686692 < 1
0.666667 < (ftheta - fx)/ftheta = 0.833741 < 1
0 < 1.5*resonant_frequency_X_comb-finger/(Modulation Voltage Frequency) = 0.857936 < 1
0 < (Kx_Stiff_beam - 10*Kxeff_spring)/Kx_stiff_beam = 0.687575 < 1
0 < x-dir-size/MaxSize = 0.7704 < 1

II.4 Evaluation of the Manual-Design using Lumped Parameter Models

mass(mxeff): 0.562122 (ug) mass(myeff): 0.601229 (ug) mass(Ithetaeff): 4.82039e-18
spring -- Kxeff: 1.60616 (N/m), Kxmech: 1.63333 (N/m), Kxelec: 0.0271692 (N/m)

spring(Ky): 37.6836 (N/m), spring(Ktheta): 1.45406e-06 (uN*um)
resonant frequency -- fx(mechanical): 8579.19 (Hz) fx(effective): 8507.54 (Hz)
resonant frequency(fy): 39845.6 (Hz) resonant frequency(ftheta): 87412.8 (Hz)
quality factor: 5.74671damping factor: 5.22866e-06 (N*S/m)
Couette Bx: 1.53583e-07 (N*S/m) Squeeze Bx: 5.07508e-06 (N*S/m)
Brownian noise floor: 53.4156 (ug/rtHz)
Parasitic cap due to circuit: 1.86e-13 comb sensing cap: 3.05746e-14
sensitivity: 0.569312 (mv/g)
bandwidth: 13148 (Hz)
noise: 0.562295 (mg)
range: 332.961 (g)
cross-axis sensitivity: 1e-07 %

0 < acc_xsize = 423 < 500
0 < acc_ysize = 267.6 < 270
0 < uSpring_ysize = 261 < 270
0.3 < gsfinger - gxlimit = 0.3 < 20
4.5 < ltruss - wbeam = 4.5 < 20
-16.5 < wpmass/2 + lrfinger -lbeam1-wtruss-gylimit-wmstop/2 =-14.1 < 250
5 < gap between uspring and finger = 4.55 < 250
0 < kxelec/kxmech = 0.0166343 < 0.8
10 (mv/g) < sensitivity = 0.569312 (mv/g) < 11 (mv/g)
0 (mg) < noise = 0.562295 (mg) < 1 (mg)
25 (g) < range = 332.961 (g) < 1000 (g)
0 < cross-axis sensitivity = 1e-09 < 0.05
100 (Hz) < bandwidth = 13148 (Hz) < 100000 (Hz)
0.2 < self_test_force/(mxeff*Range*9.8) = 1.24323
0 < 2.0*lbeam2/lcrit = 0.627306 < 1
0.666667 < (fy - fx)/fy = 0.786488 < 1
0.666667 < (ftheta - fx)/ftheta = 0.902674 < 1
0 < 1.5*resonant_frequency_X_comb-finger/(Modulation Voltage Frequency) = 1.84393 < 1
0 < (Kx_Stiff_beam - 10*Kxeff_spring)/Kx_stiff_beam = 0.69988 < 1
0 < x-dir-size/MaxSize = 0.978 < 1

III. OCEAN Scripts for Simulation-based Evaluation

III.1.a Sensitivity Simulation: mechanical part

```
simulator('spectre)
design( "/tmp/vishal/sim1/netlist/netlist")
resultsDir( "/tmp/vishal/sim1" )
delete('save )
analysis('ac ?start "100"  ?stop "1M" ?save "selected"  ?oppoint "no"  ?annotate ""  )
save('v "wire_y" )
desVar("Wrfinger" 5.7u)
run()
selectResult( 'ac )
disp = value(vm("wire_y") 'freq 100)
printf("RESULT= %L \n" disp)
wn = cross(vp("wire_y") 90 1 "falling")
printf("RESULT= %L \n" wn)
BW = bandwidth(vm("wire_y") 3 "low")
printf("RESULT= %L \n" BW)
Q = value(vm("wire_y") 'freq wn)/disp
printf("RESULT= %L \n" Q)
```

III.1.b Sensitivity Simulation: comb-drive/electrical part

```
simulator('spectre)
design("/tmp/vishal/sim1b/netlist/netlist")
resultsDir( "/tmp/vishal/sim1b" )
delete('save )
modelFile( '("/afs/ece.cmu.edu/usr/vishalg/mems/models/spice.n91x.model" ""))
analysis('ac ?start "500K" ?stop "5M" ?save "selected"  ?oppoint "no"  ?annotate ""  )
save('v "out1")
desVar("Wrfinger" 5.7u)
run()
selectResult('ac)
sensitivity = value(vm("out1") 'freq 1M)
printf("RESULT= %L \n" sensitivity)
```

III.2 Range Simulation

```
simulator('spectre)
design( "/tmp/vishal/sim2/netlist/netlist")
resultsDir( "/tmp/vishal/sim2" )
delete('save )
analysis('dc ?dev "I83"  ?param "dc_value"  ?start "10" ?stop "10k"  ?save "selected"  ?write ""
?homotopy "source" ?annotate ""  )
save('v "wire_y")
desVar("Wrfinger" 5.7u)
run()
selectResult( 'dc )
if(((range = cross(getData("wire_y") -LimitStop 1 "falling"))==nil) range=10k t)
printf("RESULT= %L \n" range)
```

III.3 $F_{\{x,\theta\}}$ Simulation

```
simulator('spectre)
design( "/tmp/vishal/sim4/netlist/netlist")
resultsDir( "/tmp/vishal/sim4" )
delete('save )
analysis('ac ?start "1k"  ?stop "5M" ?save "selected"  ?oppoint "no"  ?annotate "" )
save('v "wire_x" "wire_theta" "wire_y")
desVar("Wrfinger" 5.7u)
desVar("X_xcel" 10)
desVar("Theta_xcel" 0)
run()
selectResult( 'ac )
fx = cross(vp("wire_x") 90 1 "falling")
deltaX = value(vm("wire_x") 'freq 10)
deltaY = value(vm("wire_y") 'freq 10)
crosens = deltaY/deltaX
desVar("X_xcel" 0)
desVar("Theta_xcel" 0.005)
run()
selectResult( 'ac )
ftheta = cross(abs(vp("wire_theta")) 90 1 "rising")
printf("RESULT= %L\n" fx)
printf("RESULT= %L\n" ftheta)
printf("RESULT= %L\n" crosens)
```

III.4 Spring Constants Simulation

```
simulator('spectre)
design( "/tmp/vishal/sim3/netlist/netlist")
resultsDir( "/tmp/vishal/sim3" )
delete('save )
analysis('ac ?start "1"  ?stop "10" ?save "selected"  ?oppoint "no"  ?annotate "" )
save('v "wire_y" )
desVar("Wrfinger" 5.7u)
desVar("Vmrms" 0)
run()
selectResult( 'ac )
disp = value(vm("wire_y") 'freq 10)
kxmech = 1e-9/disp
desVar("Vmrms" 1)
run()
selectResult( 'ac )
disp = value(vm("wire_y") 'freq 10)
kxelec = 1e-9/disp
kxelec = kxmech - kxelec
printf("RESULT= %L \n" kxmech)
printf("RESULT= %L \n" kxelec)
```

IV. Simulation-based Synthesis Results

Table 7: Design Variables for Designs Synthesized using Simulation-based Synthesis

DesignVar	Noise Optimized Design	Sensitivity Optimized Design	Area Optimized Design
L_{beam} (μm)	112.3	135	114.4
W_{beam}(μm)	1.8	1.8	1.8
L_{truss}(μm)	4.98	4.85	7.21
W_{truss}(μm)	10	5	8.76
$Num_{trussbeams}$	5	5	5
L_{finger}(μm)	110	110	93.6
$W_{rotor\text{-}finger}$(μm)	5.7	5.7	5.7
G_{limit} (μm)	1.54	1	0.95
GAP_{finger}(μm)	1.92	1.5	1.5
$W_{strongbeam}$(μm)	7.91	7.7	10
L_{pmass}	38	40	30
W_{pmass}	5	8	8
$Num_{sfinger}$	15*	19*	15*
$Num_{ffinger}$	4	3	1

*. $Num_{sfinger}$ sense fingers constitute one comb-drive on each side of the plate-mass, as shown in Fig. IV.2(b)

NOTE: All dimensions listed in context of Simulation-based Synthesis are edge-to-edge dimensions (because NODAS assumes its element parameters to be edge-to-edge dimensions).

V. Accelerometers Fabricated with AMS

Table 8: Design Variables for Designs Fabricated & Tested

DesignVar	Noise Optimized Design	Sensitivity Optimized Design	Area Optimized Design	Range* OptimizedDesign
L_{beam} (μm)	124.9	124.9	124.9	124.9
W_{beam}(μm)	1.8	1.8	1.8	2.1
L_{truss}(μm)	6.8	6.3	7.4	6.8
W_{truss}(μm)	5	5	5	6.3
$Num_{trussbeams}$	5	5	5	5
L_{finger}(μm)	110.1	110.1	110.1	56.1
$W_{rotor\text{-}finger}$(μm)	5.7	5.7	8.7	5.7
G_{limit} (μm)	1.5	0.9	1.3	1.0
GAP_{finger}(μm)	2.9	1.5	1.6	1.5
$W_{strongbeam}$(μm)	6.4	10	1.83	4.8
L_{pmass}	36+4**	36+4**	16+4**	36+4**
W_{pmass}	1+4**	1+4**	1+4**	13+4**
$Num_{sfinger}$	10	10	5	12
$Num_{ffinger}$	1	3	1	1

*. This design was optimized with an objective to maximize range of the accelerometer. However, usually CMOS accelerometers have a good range (> 100G) so that this objective is not of much importance, and hence, has not been discussed in chapter 3.

**.4 squares are used for routing in the plate-mass in *x* as well as *y* direction