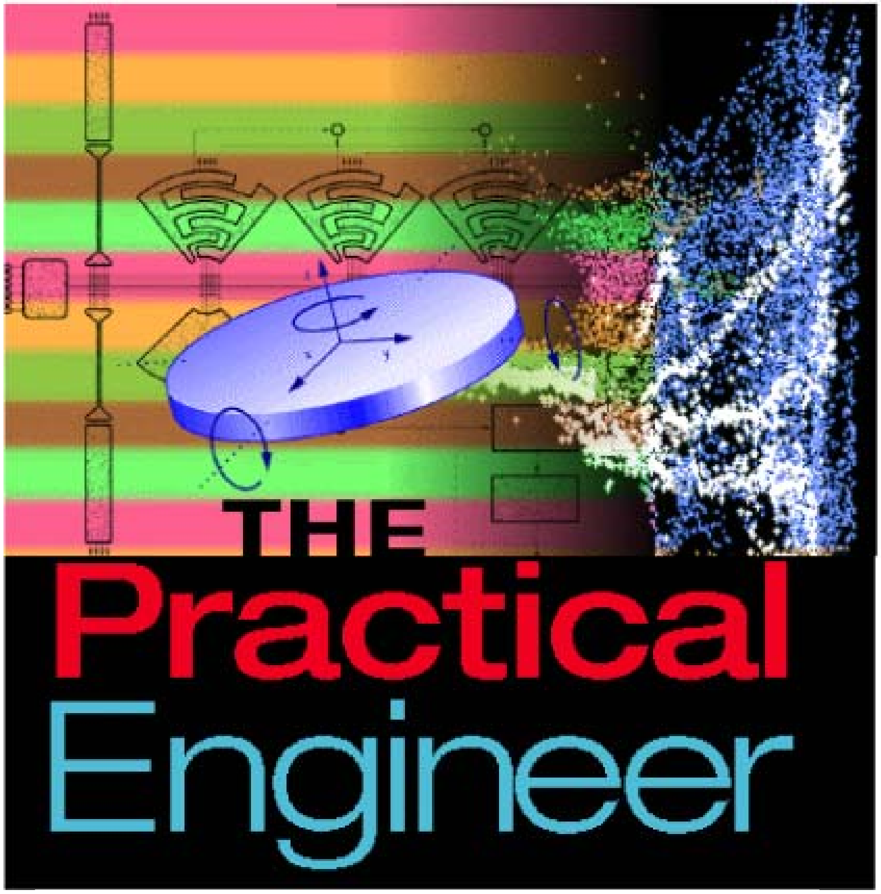


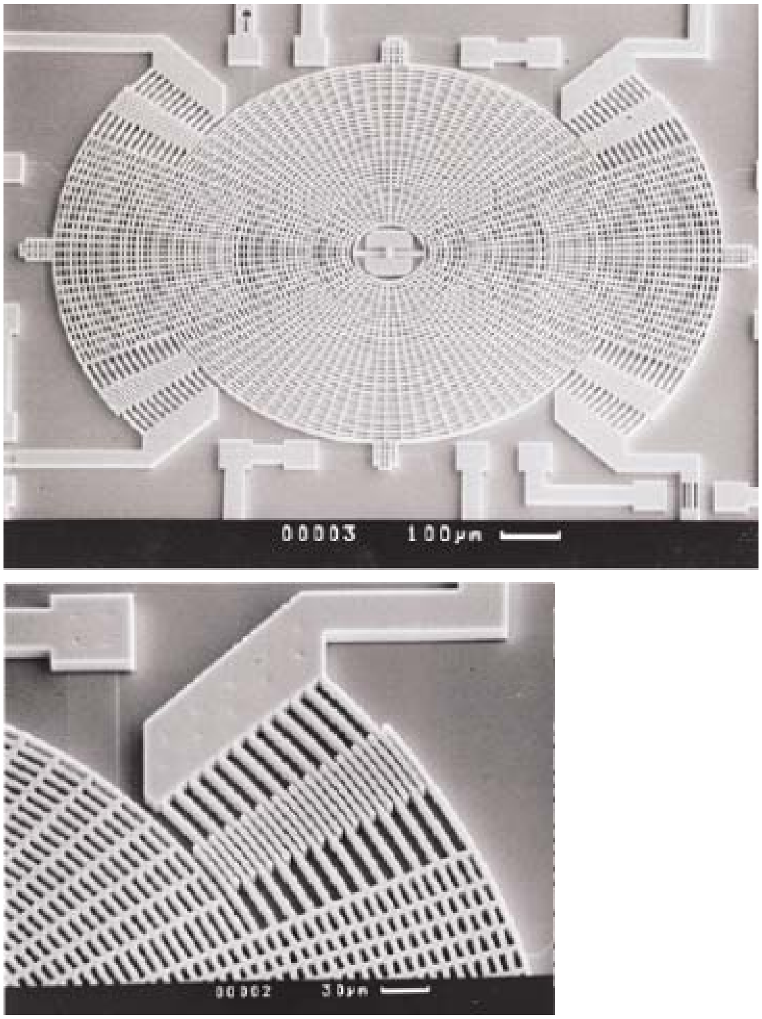
How to model and simulate microgyroscope systems

Darrell Teegarden **Analogy Inc.** &
Gunar Lorenz & Reinhard Neul **Robert Bosch GmbH**

The shrinking of entire microsystems onto silicon chips has already bred products with virtuoso performance at moderate prices. And while the earlier systems-on-a-chip mixed analog and digital circuitry, the newer ones are likely to toss in mechanical components--perforce microscopic in size--as well.

The technology is variously referred to as microelectromechanical systems (MEMS) or microsystems technology. The technique used is called micromachining, although in fact semiconductor processing is involved. Material deposition, sacrificial oxides, and etching help to sculpt moving parts that may be no larger than a micrometer.

Leading the way are micromachined accelerometers, common today in systems for deploying airbags. Looming on the horizon is a new class of microfabricated inertial sensors--angular rate gyroscopes [Fig. 1].



[1] The micromechanical angular rate sensor has a butterfly-shaped polysilicon rotor suspended above the substrate, free to oscillate about the center tether [top]. The rotor's perforations are a necessary evil, needed to allow etching beneath the rotor during manufacturing. Four interdigitated combs on the outer edge of the rotor drive it into resonant oscillation [bottom]. Electrical leads carry the driving signal to the combs and the measurement signals from the detection electrodes below the rotor.

These sensors measure the rate at which an object rotates. When they can be mass-produced on fabrication lines, they will engender reliable yet economical systems for monitoring how and where vehicles move, preventing them from flipping over, and controlling skids. Robert Bosch GmbH, Stuttgart, Germany, plans to begin producing micromachined angular rate sensors this year.

Defining Terms

Obviously, the process of manufacturing this kind of structure bears heavily on the future of the gyro devices, and will be covered in upcoming *IEEE Spectrum* articles. But equally vital, and a prior step, is the tool set needed to model and simulate MEMS and their interaction with electronic circuits.

Designing microsensor systems today is a complicated business. The skills of an analog and digital IC designer, a materials scientist, and a structural engineer are all needed. In addition, as the sensor element evolves, designers have to cope with complex interactions between micromechanical and electrical system components. Then there is the unknown: mechanical-thermal noise, variations of device parameters due to the manufacturing process, and other effects perturb signals in ways that have yet to be investigated and understood. These issues present serious technical challenges to very large-scale IC (VLSI) designers who are now being asked to build microsensor systems on silicon.

Much of the knowledge is at hand. Methods for designing digital systems are advanced and well known. Techniques for designing analog electronic systems are also well understood but less automated, even counting the use of analog circuit simulation tools such as Spice or Analogy's Saber simulator. Structural designers are at home with tools that apply finite-element methods to the analysis of the spatial distribution of forces and structural displacements. Physicists investigate the spatial relationships among electric fields, charge distributions, and electrostatic forces. But that still leaves it up to the microsensor system designer to integrate everything into his design.

Engineers at Robert Bosch have done battle with the problem successfully. They were designing a micromachined angular-rate gyroscope chip, and faced with simulating and analyzing it, they assembled a set of existing simulators and field solvers for the job.

Designer needs

As MEMS design is practiced today, it commonly calls for a large number of foundry runs to build prototypes and for time-consuming experiments to obtain a detailed overview of how the devices behave. This need not be the case. If an accurate physical model of the MEMS device were used in running simulations of the system, the characterization process could be speeded up and improved.

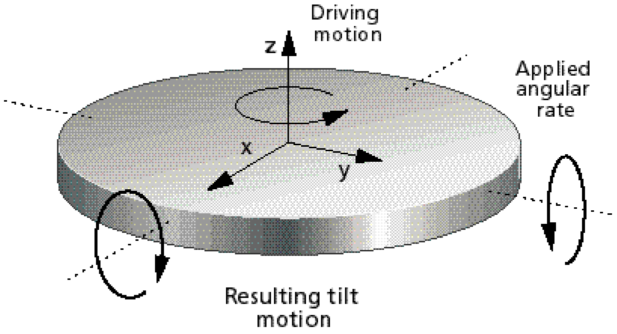
Analysis tools can be categorized as either field-solvers or network simulators. Each type is strong where the other is weak. Field-solvers--such as finite-element method and boundary-element method simulators--can solve complex partial differential equations derived from a detailed description of the physical design; but those equations are far from simple and take a lot of time to solve, disqualifying field-solvers for system-level modeling. Network simulators, conversely, do well at system description and simulation with a building-block orientation, but do not inherently model the physical design. Also, few system simulators for ICs--whether digital types or based on Spice--can tackle both mixed analog-digital and mixed mechanical-electrical system simulation.

What MEMS device and system designers urgently need is physical-level and system-level tools customized for MEMS analysis and designed so that the two levels can exchange information. This exchange is best realized by taking the results of the physical-level tools and encapsulating them into models usable by the system-level tools. In effect, the accuracy of the long-drawn-out physical simulations is passed on to the system designer at faster system-level simulation speeds.

The key to such model encapsulation is the use of an analog hardware description language, such as the MAST modeling language. Models written in MAST work with the Saber simulator. (Both MAST and Saber are commercial tools obtainable from Analogy Inc.) There is no need to tie separate special-purpose simulators together, because Saber's single simulation engine can handle analog-digital and electronic-mechanical simulations at one and the same time. Field-solver tools intended for MEMS applications are now being sold by such companies as IntelliSense Corp., Wilmington, Mass.; Integrated System Engineering AG, Zurich; and Microcosm Technologies Inc., Raleigh, N.C. The AutoMM tool from Microcosm goes one step further: it automates the job of generating MAST models from MEMS physical design descriptions.

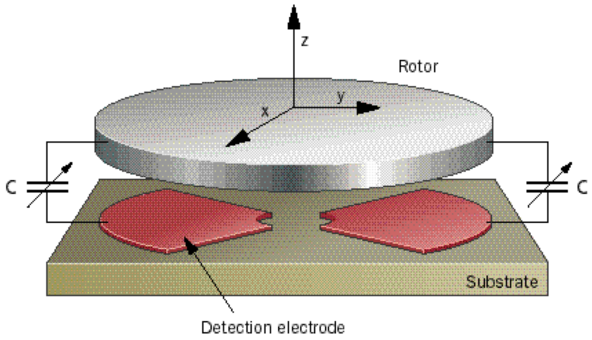
For modeling the angular-rate sensor, Bosch engineers chose a physical-level field-solver called Mafia to simulate the entire sensor design one piece at a time. Symmetry and simplifying assumptions were used to reduce the complexity of the problem wherever possible. (Mafia stands for the solution of MAxwell's equations by the Finite Integration Algorithm. A commercial, general-purpose tool from Computer Simulation Technology GmbH in Darmstadt, Germany, it has been in use for over 20 years.) From the simulations of each piece, the engineers extracted a system-level model into MAST manually, but systematically. The models were then collected into a library of building blocks that could be reused and recombined to investigate new designs and approaches. The building blocks were also used in simulations to detect and explore critical interactions between electronic and mechanical blocks without the need for costly fabrication runs and delays. The whole sensor system, including readout electronics, was modeled with the Saber network simulation tool.

Angular-rate gyro system model



[2] The rotor is driven into a resonant oscillation about the z-axis. When the device package is rotated about the y-axis, the substrate rotates about the same axis--the motion is transferred to the rotor through elastic connections at the center. Obedient to the law of conservation of angular momentum, the rotor tilts along the x-axis. This motion is sensed by the electrodes below the rotor and converted into an electrical signal.

The mechanical sensor consists mainly of the inertial rotor [Fig. 2]. This flat butterfly-shaped polysilicon structure is suspended by two beams in the center of the structure, around which it can rotate by about 3 degrees. The rotor is caused to oscillate about the vertical (*z*) axis by four comb drives on the outer edge of the structure. The rotor's oscillation frequency of about 1.5 kHz is close to its resonant frequency (that is, the frequency at which the rotor naturally oscillates about its suspension points). When the system is rotated about the *y* axis (as might happen when an airplane banked, for example), the rotor tilts and wobbles about the *x* axis, upholding the physical principle of conservation of angular momentum. The tilt oscillation's frequency is the same as that of the resonant oscillation and its amplitude is proportional to the rate at which the system is rotating about the *y* axis. The variable tilt is detected by measuring the change in the capacitance between the rotor and a pair of underlying electrodes [Fig. 3].



[3] The oscillation of the tilting rotor is quantified by measuring the differential capacitance between it and the electrodes beneath. Capacitance measurements corresponding to rotor displacements of about the diameter of an atom can be attained.

For modeling purposes, Bosch engineers lumped the complex mechanical behavior into a handful of reusable electromechanical components. These were the inertia of the rotor, its physical connections to the substrate, and the damping effects of the air around it. From the mechanical point of view, the inertial rotor is simply a mass with six degrees of freedom of motion. The link to its surroundings is described by six coordinates in space (its location along the *x, y,* and *z* axes and its rotation about the same three axes) and the forces and torques that act on it. Also for the purpose of modeling, Bosch engineers shrank the butterfly-shaped rotor to a dimensionless point characterized only by its mass and moment of inertia about each axis.

The rotor's suspension from the substrate was modeled with beams and levers. The levers represent the geometric link between the rotor mass and the connecting points to the beams. The beams model the elasticity (force/position relationship) of the members connecting the rotor mass to the substrate anchors. Thanks to Saber's modeling language, MAST, these elements can be described directly by algebraic and ordinary differential equations.

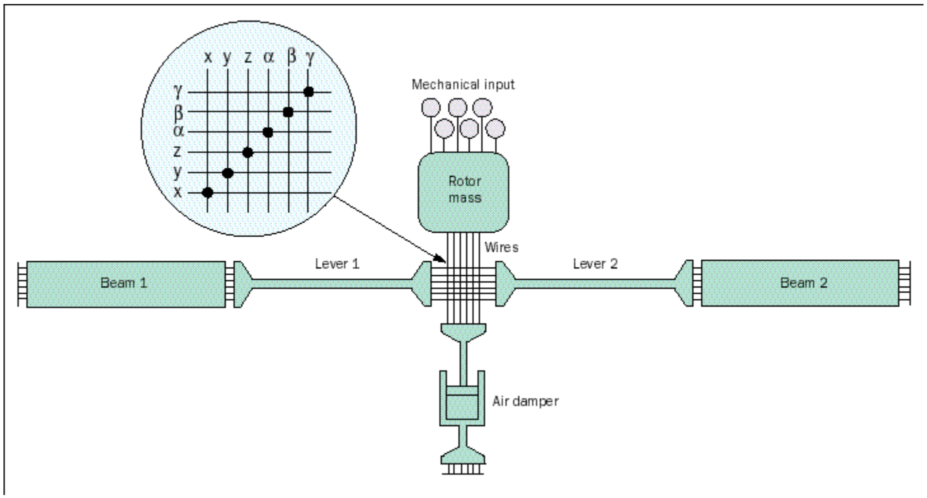
The beam and lever elements described thus far suffice to model the key static characteristics of the gyro system (for example, the rotor's angular displacement as a function of applied torque). But because the rotor is always oscillating, the system performance cannot be analyzed without including the rotor's inertia and the dynamic damping effects of the surrounding air. The pneumatic damping is also the chief source of noise in the system. The random, temperature-dependent motion of the air molecules impinges on the movement of the minuscule mechanical structures, producing noise in the electrical signal. Analyzing such dynamic details using a field-solver simulator is prohibitively slow. But using a system simulator like Saber, with models derived from field-solver simulations, makes such simulations feasible.

Thermal noise is well known in the world of electronics but has seldom to be considered in large mechanical systems. But the effect cannot be ignored in micro-scale MEMS structures, where displacements on the order of the diameter of an atom have to be measured.

Mechanical-thermal noise effects depend on the absolute temperature and the damping of the mechanical system. Among the effects are noise induced by temperature-dependent oscillations of the polysilicon crystal lattice of the MEMS structure and the impact of surrounding gas molecules. The noise shows up as random motion of the structure, independent of frequency, and in the case of the gyro system, as random voltage fluctuations of the detection electrodes. Just as the white noise (whose power is independent of frequency) in an electrical component depends on the value of its resistance, so the white noise in a mechanical system depends on the system's mechanical damping. Any mechanical system in thermal equilibrium, no matter how complex, can be analyzed for mechanical-thermal noise by adding a noise force generator alongside each damper.

Because of its correspondence to damping, this noise can be regarded as an upper limit of the allowable air pressure inside the sensor housing. The fewer the molecules inside the chamber, the fewer the molecular impacts. Even so, damping coefficients are difficult to determine analytically and are the subject of future development work.

The mass, beam, lever and damper components complete the rotor model [Fig. 4].



[4] A Saber schematic of a damped oscillating mass suspended by two beams represents the resonating rotor minus the comb drives or sensor electrodes. The three spatial and three rotational coordinates are represented by "wires" between the two levers and between the mass and the damper. The forces and torques at each connection point (where x intersects x, for example) are summed to zero, yielding differential equations that are solved to determine the motion of the system. Simulations showing the positions as a function of time or frequency were performed from this schematic.

All the components are connected on the schematic by six "wires," stand-ins for the six degrees of freedom of the rotor mass in Saber. These wires are called the independent across-variables, while the torques and forces that depend on the rotor's position and rotation are called the dependent through-variables. [See "A variable account," at the end of this document]

The ground symbols at the terminals of the beam and damper symbols refer to the substrate coordinate system. These connections are analogous to the ground reference in an electrical network. Finally, the sources provide mechanical inputs that simulate the torques and forces exerted by the sensor's external environment (namely, the stimulus that the sensor is designed to measure). All that is missing from the sensor model now are the comb drives and the sensing electrodes.

Modeling comb drives

The combs on the MEMS sensor [Fig. 5] are curved polysilicon fingers that serve both as actuators (converting electric energy into forces and torques) and as sensing devices, to detect displacements between the substrate and the moving rotor. The combs come in interdigitated pairs: one part, the stator, is anchored to the substrate physically but isolated from it electrically, and the other is attached to the rotor. The displacement between the combs is determined by measuring the electrical capacitance between them. An electronic circuit translates the change in capacitance into a meaningful sensor signal.

The forces between the combs depend on their relative position and the applied voltage. Control of this voltage gives precise control of the movement of these microstructures; and for the angular-rate sensor an independent and stable amplitude of the resonant in-plane oscillation can be guaranteed. Thus, the system-level model of a comb structure must accurately model both the position-dependent capacitance and the forces between the stator and rotor comb fingers.

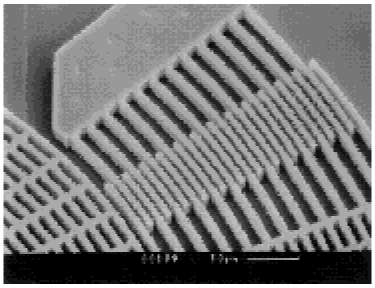
The diagram of model development shows how Bosch engineers derived the system-level model of the comb from the analysis of the physical-level description [Fig. 5]. For the purpose of modeling, the curved comb structures were approximated by a simple linear description in which all the fingers are of the same length, equal to the average length of the curved fingers. Further, the whole structure was assumed to be composed out of the basic element shown, while symmetry allowed us to further simplify the structure geometry [Fig. 5, second and third from top, respectively].

The simplified geometry of the structure, composed of two rectilinear blocks, was entered into the electric field solver tool, Mafia. From the potentials applied to the rotor, stator, and substrate, Mafia calculated the electric field distribution within the structure. From an electrostatic standpoint, the physical-level model was viewed as three charge distributions in space, analogous to the three conductors (rotor, stator, and substrate). The job of the field-solver tool was to determine the charge distribution and then to calculate the lumped capacitances between the three conductors.

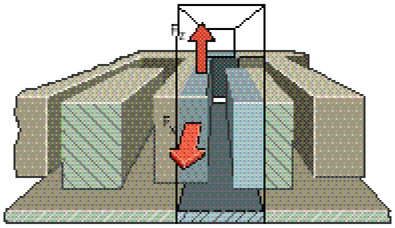
The capacitances had to be determined for all three combinations: rotor-substrate, stator-substrate, and rotor-stator. These capacitances are a convenient way to ignore the distributed nature of the rotor, stator, and substrate conductors. But they are a strong function of the relative geometric position of the conductors. As the conductors move relative to each other, the terminal capacitances change nonlinearly.

The three-dimensional surface shown in Fig. 5, fourth from top, is a plot of the output capacitance data of the Mafia field solver. The data shown here is for the stator-substrate capacitance as a function of in-plane (*x*) and out-of-plane (*y*) motion of the rotor in relation to the stator. Similar plots were drawn for the rotor-substrate and the rotor-stator capacitances. Although these surface plots contained the information required to develop accurate high-level models, their form was not conducive to encapsulation into a MAST model of the comb drives.

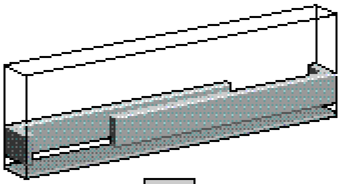
Micromechanical comb structure



Cutaway model of the comb

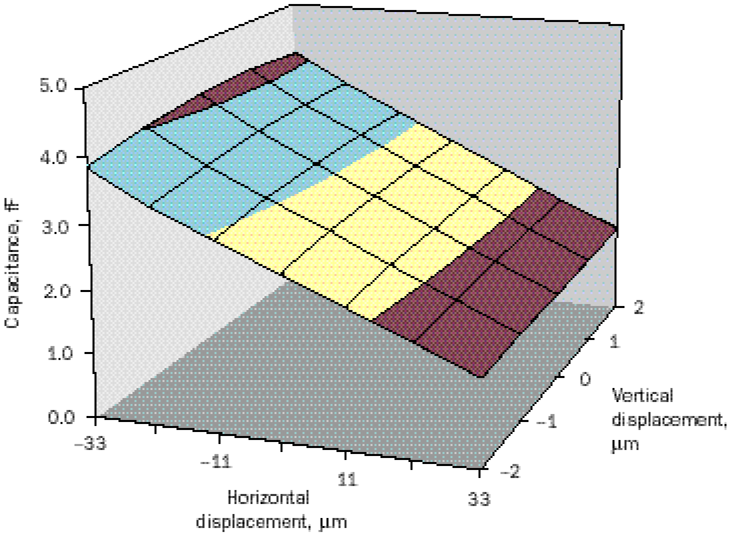


Simplified physical model



Results of repeated calculations

Capacitance vs. displacement



Fitted capacitance functions

C = f(x,z)

[5] The sequence of steps to generate a system-level model of a comb drive include [from top down]: a micrograph of the comb drive; a three-dimensional approximation of the structure; simplification using symmetry; field-solving calculations as a function of relative position; curve-fitting equations and force calculations from the capacitance matrices C and the node voltages V. The capacitance and force equations and help create a system-level model by implementing the equations in an analog hardware description language. The result: a fast, reusable model parameterized only by the calculated capacitance matrices.

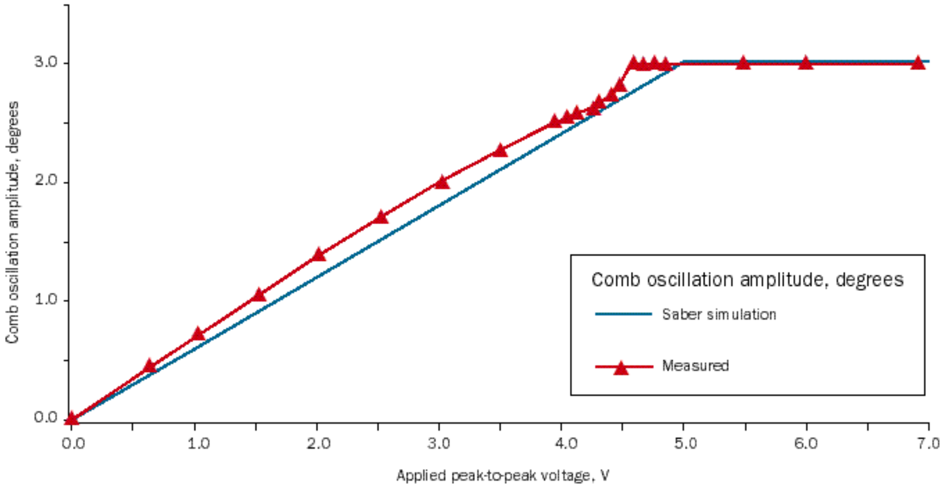
To remedy matters, each of the three surfaces was reduced to an equation that captured the capacitance as a function of *x* and *y*. This was done by using least-square fitting routines available in the Matlab tool from The Mathworks Inc., Natick, Mass. The result, represented by the bottom element of Fig. 5, is three cubic equations with coefficients that have been fit to the data in the part just above.

The algebraic functions yield capacitance as a function of the rotor's in- and out-of-plane position in relation to the stator. They can be differentiated to yield the partial derivatives as a function of *x* and *y*, respectively, which in turn are used to calculate the electrostatic forces.

Enter the physical principle of virtual work--the basis for the calculation of the electrostatic force on the rotor, stator, or substrate. This principle starts with the electrostatic force on a body in equilibrium and relates it to the change in stored energy (work) associated with its motion in the direction of interest. (Incidentally, the principle of virtual work is related to the definition of work as the scalar product of a given force vector displacing an object with a given displacement vector.) The electrostatic-force equations were implemented directly in the MAST modeling language. The curve-fitting coefficients determined in Fig. 5 became input parameters to the MAST model. Note that the model could be reused for a geometrically different comb drive by simply repeating the steps in the figure to generate a new set of coefficients for the model.

Thus was born a reusable, compact, accurate, computationally efficient model of the comb drive. None of the tedious field-solver computations and curve-fitting operations need be done more than once. The force calculations, carried out during the Saber system-level simulation, were a simple function evaluation that depended on the constants for the cubic equation used to calculate the capacitance and the comb drive voltages.

The ultimate test of the accuracy of Saber simulations is to compare them with reality. In Fig. 6, the blue curve is measured on a real sensor and the yellow one derives from the corresponding Saber simulations. Both set the oscillation amplitude of the resonant in-plane motion of the rotor against the voltage amplitude applied across the comb structures. As the driving amplitude builds up, so, too, does the mechanical amplitude of the oscillating rotor. At an amplitude of three degrees, however, both curves stop increasing; the rotor has reached the mechanical limits of its in-plane oscillation. Another measure of simulation accuracy is to compare Saber simulations with the results of the Mafia three-dimensional simulations.



[6] The comb drive model was verified by physical measurement on a gyro test structure. The test comb drive device was stimulated with a voltage of varying amplitude, and its consequent oscillation was measured, in degrees. The corresponding simulations matched well.

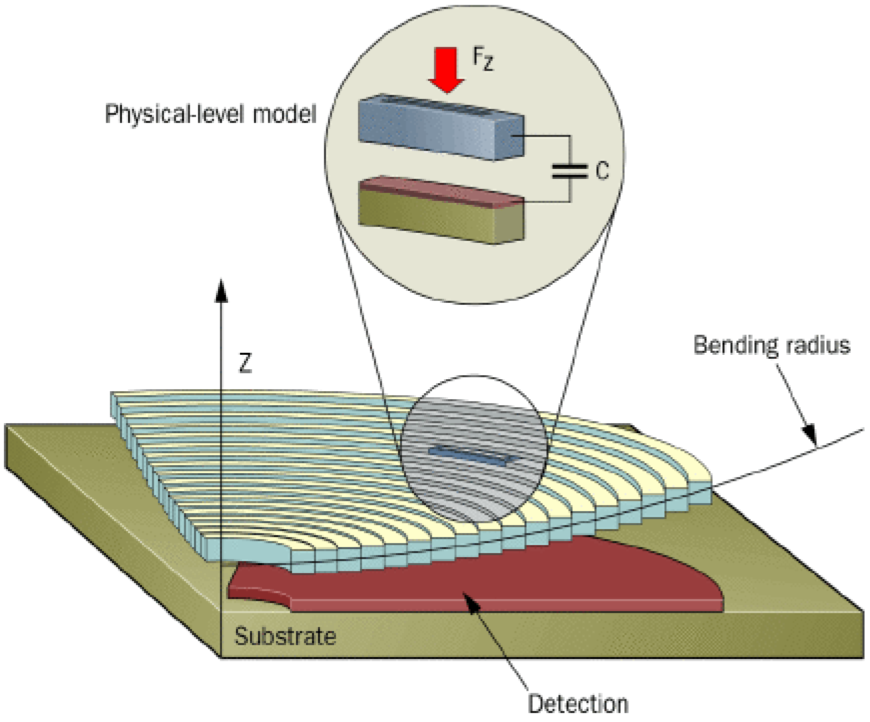
Modeling detection electrodes

The other parts needed to complete the system-level sensor model are the pie-shaped electrodes that lie beneath the rotor and measure its tilt in relation to the substrate. Whenever the rotor tilts, the capacitance between it and the underlying electrodes changes. Their size and arrangement, therefore, strongly influence the sensitivity of the gyro. Once again, field-solver calculations supplied the accuracy needed to model this effect.

The detection scheme can be likened to a parallel-plate capacitor. One of the plates is the detection electrode itself, a thin conductive plate almost ideally shaped for a capacitor. The rotor directly above the detection electrode forms the other plate; but this plate is not ideal because manufacturing steps require it to be perforated like a lattice in order to provide access for the etching that allows it to move.

In addition, micromechanical structures are typically bent by surface stress caused by the manufacturing process. This gives the inertial rotor the look of a dish with a large bending radius.

In order to describe such a structure with reusable building blocks, several more basic elements have to be defined. As with the comb structures, the rotor structure is treated as being composed of elementary cells [Fig. 7].

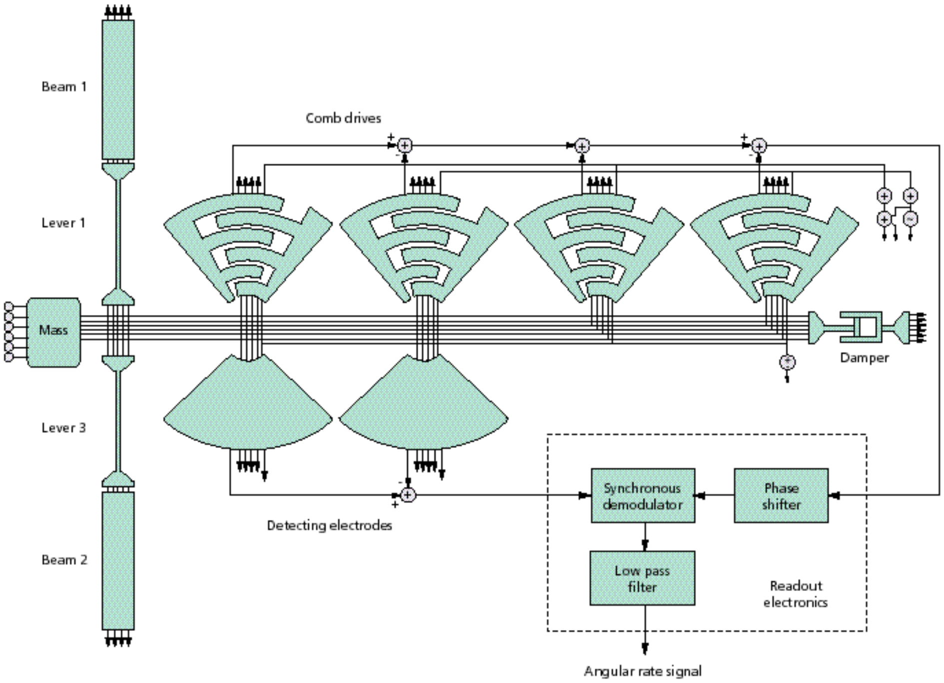


[7] The capacitance between the rotor and the substrate (stator) and the resulting force exerted on the rotor were modeled with an approach similar to the comb drive. The rotor's curvature, which is a manufacturing side-effect, complicates the model's development. Careful creation of the elementary cells [enlarged portion] makes possible a fast, accurate model of the detection electrodes to be constructed.

The elementary cells are blocks of different lengths with a slot etched into them. Together with the underlying fragment of the detection electrode, they form an elementary capacitor defined by the block length and the air gap between block and electrode. A physical-level structure of such simplicity can easily be analyzed with a field-solver. Rotor curvature is handled by varying the air gap in different cells. The calculations are repeated for several block lengths and several distances between block and electrode, producing two-dimensional functions for the force and the capacitance of such a cell.

Once such functions exist, the capacitances and the forces between the rotor and the detecting electrodes can be calculated by adding up all cell forces and capacitances--under the assumption that the effects of neighboring cells are additive. During the Saber system-level simulations, the capacitances and forces between the rotor and stator can be quickly calculated from the position of the rotor with respect to the stator.

A component that models the detection capacitance provides the final part needed for the sensor model. Combining the basic elements culminates in a network description for the system's behavior [Fig. 8].



[8] A schematic of the entire angular rate sensor network was constructed from the derived building blocks using the SaberSketch schematic tool that comes with the Saber simulator. The effects of the comb drives, the detection electrodes, the damping of the air, the inertia of the rotor, and the elasticity of the suspension can be observed for all six degrees of freedom of motion. The effectiveness of the signal-processing electronics can be determined. In fact, system-level design tradeoffs can be investigated in a very short time--a matter of minutes.

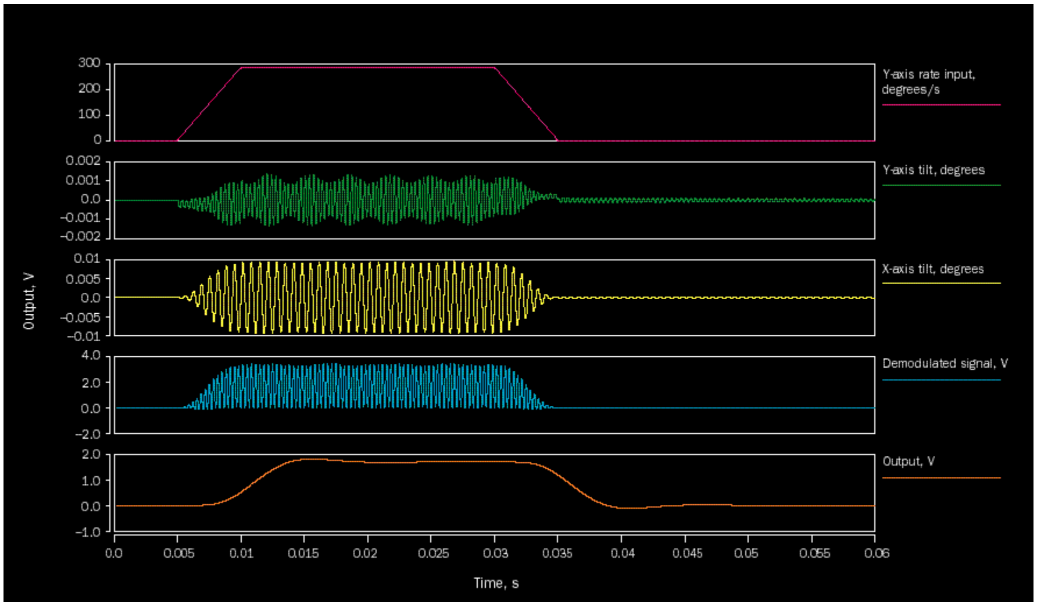
The gyro system schematic

The model of the rotor and its suspension appear on the left side of the schematic. The seven horizontal lines allude to one electrical plus six mechanical "connections" between the rotor mass and the beams and levers that represent its suspension from the substrate. Each mechanical connection represents a mechanical degree of freedom. The electrical connection is the voltage between the rotor and the stator.

The symbols describing the interdigitated radial comb drives can be seen above these connections. They are connected to the voltage sources [top right corner]. During simulation, they produce enough driving voltage to excite the resonant in-plane motion.

The detection electrodes are the fan-like structures below the rotor connections. All comb and detection electrode symbols have a "capacitance" output. These serve as an input to a behavioral model (one that describes the "behavior" of the electronic readout circuit with high-level equations rather than with transistors). What determine the appropriate techniques for circuit modeling are the tradeoffs desired between the accuracy and the run time of the simulation. The detection circuitry could just as easily be modeled as a transistor-level circuit or a mathematical function that describes how a circuit's input is related to its output. Different levels of model abstraction--for example, a behavioral model, or one composed of functional blocks or transistors--can be freely mixed in the same Saber simulation. A mixture of analog and digital models may even be included, if appropriate.

Once such a system model exists, the designer can investigate its response to applied angular rates or accelerations in short order--just a few minutes of simulation with the Saber simulator on a workstation or PC. Generally, a simulation of transient behavior shows the mechanical input stimulus, intermediate mechanical and electrical responses, and the output signal [Fig. 9].



[9] The gyro system simulation shows [from top to bottom] the input stimulus, the motion of the rotor in the same plane as the stimulus, and its motion in the x-axis (conserving angular momentum). The remaining signals show the conversion of this x-axis tilt information into a usable electronic signal by demodulation and filtering. The final signal shows the sensor's output in response to the original input.

The time-varying quantities of the corresponding connections inside the model are presented in five plots. The first graph [top] shows the angular rate stimulus applied to the sensor (in the *y*-axis) by applying a mechanical angular rate "source" to the sensor substrate mechanical connection. The second graph shows the motion of the rotor, in the same plane as the input stimulus, as it reacts to its inertia around the input axis. The principle of conservation of angular momentum is reflected in the appearance of the signal in the third plot. It is a mechanically amplitude-modulated signal at the frequency of the resonant (z-axis) oscillation with an amplitude that corresponds to the applied angular rate. This signal is the fundamental "information carrying" signal.

The model of the electrical readout circuit is the source of the two bottom plots, which show the electrically demodulated and filtered signals of the tilting rotor. Note how similar the output voltage and the y-axis rate Input are. There is a slight delay and rounding of the signal, compared to the original input, due to the dynamics of both the mechanical and electrical system. Most of the delay stems from the low-pass filter used to remove the resonant oscillations from the demodulated signal. This filter's characteristics can be designed to trade off response time and the feed-through of the resonant oscillations.

Practical with today's tools

The virtues of system-level modeling of MEMS structures are obvious. System performance, efficiently and systematically traced to the physical design, can be quickly investigated with computer simulations. Whereas development of these techniques and the actual models took about one person-year of effort, models for new gyro devices can now be completed in a few days and simulated in minutes. This translates into far fewer costly fabrication runs and lengthy delays. Design variations can be explored very quickly.

For example, a designer can investigate the potential performance gains of adding two extra comb drives by executing a simple copy-and-paste operation on a schematic. Once the basic building blocks of the MEMS structures are defined and characterized, portions of designs can be reused and new designs can be constructed, all without delay. Best of all, these analyses can be undertaken with commercial tools available today.

To probe further

To learn more about Robert Bosch GmbH and its products, inquire by fax, (49+71) 1811 7622. Or visit their World Wide Web page at http://www.Bosch.de.

For more information about the Saber simulator and MAST modeling language from Synopsys, see the Web page (http://www.synopsys.com)

The Mafia field-solving tool is available from Computer Simulation Technology in Darmstadt, Germany (http://www.cst.de). Mafia is a general-purpose field-solving product based on the theory of discrete Maxwell grid equations.

Commercial tools focused on analysis of MEMS physical designs are available from IntelliSense Corp. at http://www.intellis.com/, Integrated System Engineering AG at http://www.ise.ch/, and Microcosm Technologies Inc. at http://www.memcad.com.

An early text *Electromechanical Dynamics* by H. H Woodson and J. R Melcher (John Wiley & Sons, New York, 1968) provides an excellent foundation for analysis of electromechanical systems.

A considerable effort has been funded by the U. S. Defense Department (Darpa) to promote the development of tools for design and analysis of MEMS systems. See the Composite CAD Web page (http://web-ext2.darpa.mil/ETO/CompCAD). Several of the projects in this program are developing models compatible with the Saber simulator. These include Carnegie Mellon University,

(http://www.ece.cmu.edu/afs/ece/usr/mems/www/mems-projects.html), Microcosm Technologies (http://www.memcad.com), and the Massachusetts Institute of Technology.

The German consortium MIMOSYS develops microsystems technology models. See http://www.c-lab.de/mimosys/welcome.htm (in German).

About the authors

Darrell Teegarden is a principal engineer at Analogy Inc., in Beaverton, Ore. His 10 years at Analogy have been focused on modeling and simulation tool development; he holds a patent related to hardware description languages.

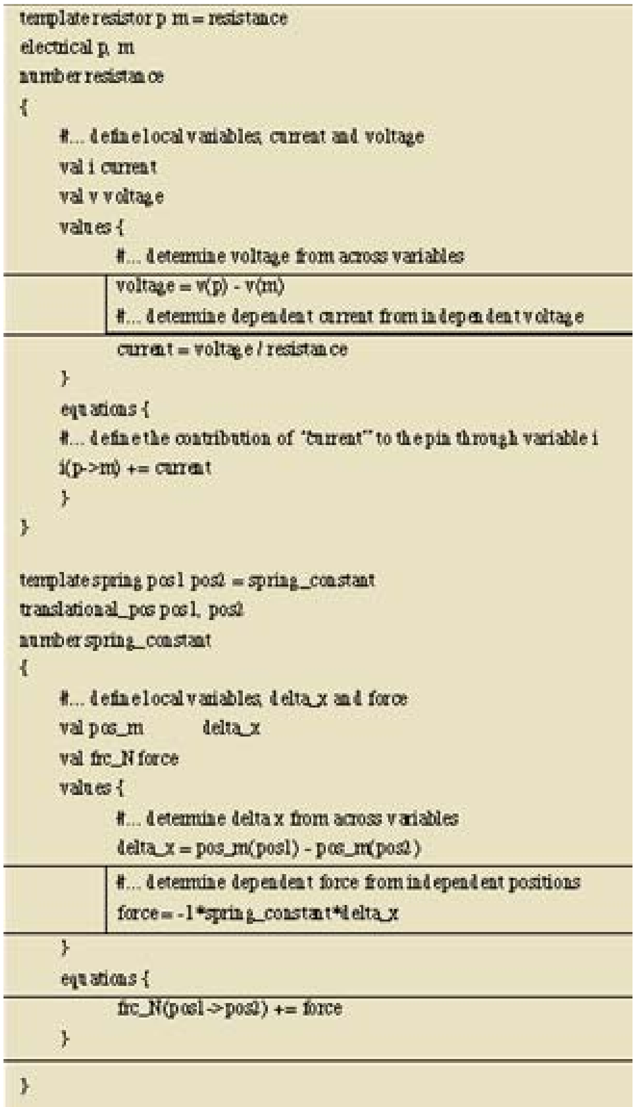
Gunar Lorenz is a doctoral candidate working at the corporate research centre of Robert Bosch GmbH in Stuttgart, Germany. Together with his supervisor Reinhard Neul, he is working on the topic of modeling microsensor systems. His e-mail address is lorenz@fli.sh.bosch.de.

Reinhard Neul is coordinator, since 1992, in the department of information and systems technology at the corporate R&D centre of Robert Bosch GmbH. He is involved in developing methods for designing mechatronic systems and especially microelectromechanical systems using modeling and simulation techniques. His e-mail address is neul@fli.sh.bosch.de.

Spectrum editor: Linda Geppert

IEEE Spectrum July 1998 Volume 35 Number 7

A variable account



```
template resistor p m = resistance
electrical p, m
number resistance
{
     #... define local variables, current and voltage
     val i current
     val v voltage
     values {
            #... determine voltage from across variables
            voltage = v(p) - v(m)
            #... determine dependent current from independent voltage
            current = voltage / resistance
     }
     equations {
     #... define the contribution of "current" to the pin through variable i
     i(p->m) += current
     }
}

template spring pos1 pos2 = spring_constant
translational_pos pos1, pos2
number spring_constant
{
     #... define local variables, delta_x and force
     val pos_m       delta_x
     val frc_N force
     values {
            #... determine delta x from across variables
            delta_x = pos_m(pos1) - pos_m(pos2)
            #... determine dependent force from independent positions
            force = -1*spring_constant*delta_x
     }
     equations {
            frc_N(pos1->pos2) += force
     }
}
```

Models from different energy domains (for instance, electrical, thermal, or mechanical energy) can be described and combined with the aid of across- and through-variables. Use of this powerful, general-purpose mechanism allows the model developer to focus on the relationship of the dependent (through) variables to the independent (across) variables and to create versatile, reusable building blocks.

For example, in a simple resistor model [see figure], the independent across-variable is the voltage across the resistor, while the dependent through-variable is the current through it.

A network of these resistor elements can be created by simply juxtaposing them at connection points and relying on the simulator for the next step: assembling and solving the system of equations for the across-variables (voltages), including the constraint that the currents must sum to zero at the connection points. Thus the model developer can create a reusable model without knowing ahead of time how it is going to be assembled into a network.

The MAST modeling language provides an easy, platform-independent way of creating these model elements. The example shown of the MAST language code models a mechanical spring with one degree of freedom. This code shows how new intrinsic building blocks can be created. The dependent through-variables are calculated as a function of the independent across-variables, which MAST allows the model writer to assume are determined. All he or she then does is calculate the resulting through-variable and "add" its contribution to the connection point. When models of this kind are combined in a network, the simulator forces all contributions of through-variables at any connection point to sum to zero--in obeyance of Kirchoff's current law for electrical networks and Newton's law for mechanical systems, for example.

Many energy domains lend themselves to being modeled by across- and through-variables. For example, in the thermal domain, temperature and power are the across- and through-variables; in the hydraulic domain, they are pressure and flow. This technique provides a convenient mechanism for building modular, highly reusable models.