A Modeling Approach to Include Mechanical Microsystem Components into the System Simulation

R. Neul, U. Becker, G. Lorenz — Robert Bosch GmbH, Stuttgart, Germany

P. Schwarz, J. Haase — FhG-IIS/EAS Dresden, Dresden, Germany

S. Wünsche — SIEMENS Components Inc., Burlington, USA

Abstract

For MEMS devices modern technologies are used to integrate very complex components and subsystems closely together. Due to mixed-domain problems as well as the occuring interactions between the closely coupled system components the design is a sophisticated process. The interactions between the MEMS components have to be analysed by system simulation already in an early design stage. In this paper a modeling approach is introduced that enables the incorporation of mechanical microsystem components into the system simulation using network and system simulators like SABER. The approach is based on multi-terminal models of basic mechanical elements and their composition to more complex microsystems. First results for a micromechanical resonator are presented.

1 Introduction

Microsystems are characterized by the interaction of components operating on different physical domains. The analysis of such complex systems requires the modeling and simulation of single components as well as the overall system simulation. Two general methods exist for system simulation:

- Coupling of special simulators for the different physical domains or the different abstraction levels.
- Modeling of the different components with a common modeling approach suitable for a powerful system simulator.

Both approaches are under investigation for some years. This paper is focusing especially on the second method.

A similar situation exists in electronics: analog and digital systems of very different complexity have to be considered in their interaction, especially in mixed-mode designs. Modeling and design of such systems is supported by powerful circuit and system simulators, libraries of basic components, and hardware description languages (HDL's). Therefore, it is useful to look for similarities in microsystems and microelectronics design, modeling, and simulation.

Using analogies, network-based modeling approaches were developed in non-electrical domains long time ago [9], [10], [17]. The main idea is to decompose a complex system into components (or subsystems) coupled together by signals. These signals may be divided into flow quantities and differences quantities. Conservation laws exist for these two kinds of quantities (generalized KIRCHHOFF's laws). Many components may be modelled as two-poles (resistors, capacitors, springs, masses, ...), but the approach is not restricted to such two-poles or two-ports (transformers, gyrators) also.

A more general approach decomposes a complicated microsystem into much more simpler components. These components may be modelled as **n-poles** (multi-terminals). These n-poles may be decomposed further into a combination of basic components (structural modeling) or may be described by equations or differential equations (behavioural modeling - as it is done in this paper). In general, we propose a **combined structural-behavioural modeling** approach.

The usage of behavioural modeled n-poles is the main difference to the approaches used in the 60's to the late 80's. Circuit simulators in this area like the well-known SPICE simulator were not able to handle behavioural descriptions. Therefore, all of the subsystems had to be structurally modelled by networks composed of very simple basic elements. Especially, the modeling of nonlinear effects and of multi-dimensional problems was too cumbersome for a widely accepted design methodology and the development of building-block libraries based on such a network approach. By using the proposed way it is possible to model microsystems in three spatial dimensions and to model the interactions between motions in different directions and between different physical domains. This is a step towards generating **nonlinear macromodels** - one of the most important possibilities for simulating the behaviour of complete micromachines [1], [18]. Already in the **linear** case, such macromodels proved to be very useful in the simulation of the interaction between the mechanical and the electronic subsystems for control and signal processing [5], [11], [14], [19], [21].

The simplest way to model complex systems by mixed structural-behavioural descriptions is the application of

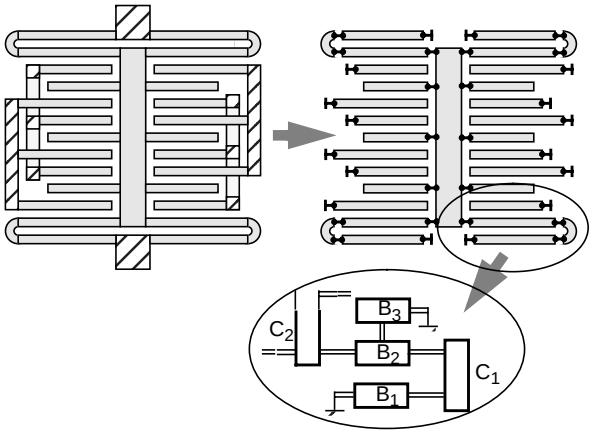


Fig. 1: Principle of an acceleration sensor and its decomposition into beam elements

Hardware Description Languages (HDL's). Powerful simulators like SABER, ELDO, or SPECTRE have their own model description languages (MAST, HDL-A, and SPECTRE-HDL, respectively). The advent of a new HDL (VHDL-AMS, under standardization by IEEE) for mixed discrete and continuous systems as well as electrical and non-electrical systems is a large step to a unified treatment of heterogeneous microsystems [3]. The ability of the forthcoming VHDL-AMS-simulators to handle large continuous systems (non-electrical as well as electrical) and digital systems (e.g. for integrated signal processing) is a very natural way to model and to simulate microsystems. For basic structures like beams or membranes, the construction of behavioural models may be supported by model generators [15]. In this context FEM simulations can be used to verify VHDL-AMS-models.

The modeling approach is sketched out in Fig. 1: A complicated micromechanical acceleration sensor is decomposed into simpler components, e.g. beam elements and coupling elements. Each component is modelled as a n-pole with its own behavioural description. Flow and difference quantities are the forces and torques in all 3 directions (x, y, z) and the displacements or velocities, respectively. The combination of the components is modeled by the interconnection of the n-poles.

In this paper, we describe the development of behavioural models of micromechanical components with different degrees of accuracy. The models were coded in MAST for the SABER simulator and their accuracy was compared with FEM simulation results using ANSYS. In this way, a model library may be developed for the design of micromechanical structures and - in principal - other kinds of microsystems.

2 Fundamentals

The analysis of mechanical components in the design of MEMS devices is based on finite element analyses. The mechanical structure is therefore split into finite elements and the behaviour of the whole scheme is modeled by the complex ensemble of these elements [2], [8]. The accuracy of the simulation is determined by the choice of the particular approach and the number of elements used. Either volume elements or spatial beams can be used as finite elements to simulate micromechanical beams (fig. 2). A volume element considers the spatial dimensions of the beam. It has 8 nodes with forces acting on and displacements occuring.

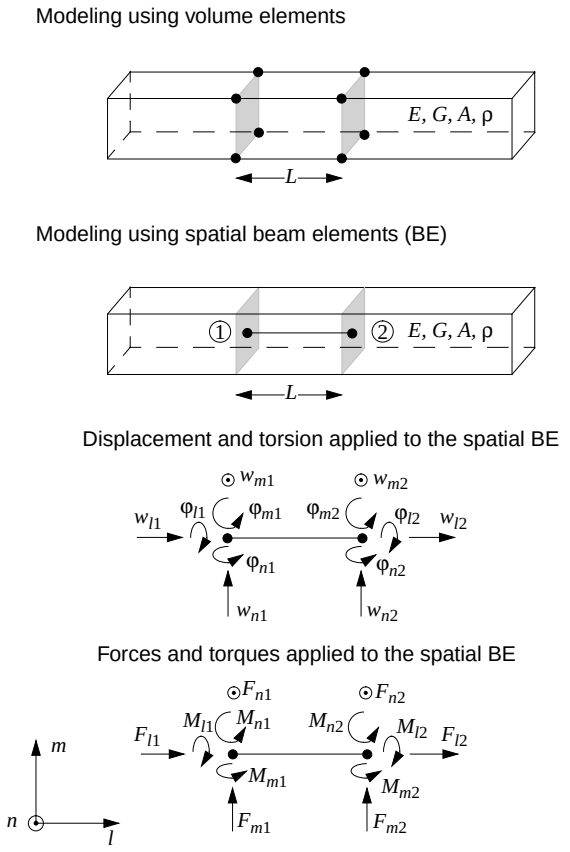


Fig. 2: FE-Modeling of the beam

The spatial beam approach is based on an abstraction, where only both end points of the beam and the forces and displacements occuring on them are taken into account. Effects like e.g. the transverse contraction cannot be taken into account with this type of model.

Static and dynamic behaviour of each finite element is modeled by equation (1).

$$\boldsymbol{M} \cdot \ddot{\boldsymbol{x}} + \boldsymbol{D} \cdot \dot{\boldsymbol{x}} + \boldsymbol{K} \cdot \boldsymbol{x} = \boldsymbol{F} \qquad (1)$$

$\boldsymbol{M}$, $\boldsymbol{D}$ and $\boldsymbol{K}$ represent the mass, damping and stiffness

matrices of the element [13]. Displacements and torsions of the nodes are comprised in vector ***x*** and the forces and torques acting on the element are represented by vector ***F***. An entire micromechanical structure is composed of these finite elements. The behaviour of the entire system is described by an equation of the same type as equation (1), but using stiffness, mass and damping matrices ***M'***, ***D'*** and ***K',*** which are composed of the matrices of the elements. Subsequently vectors ***x'*** and ***F'*** are used, containing all nodes of the entire system. Fortunately the work of setting up ***M'***, ***D'***, ***K'***, ***x'*** and ***F'*** is done by the simulator.

As the spatial beam shall be the basis of the modeling for the system simulation with SABER, it will be considered in more detail. If a spatial beam element is considered (fig. 2), only both ends (node 1 and 2) are important for the external behaviour. Linear and rotational displacements in the 3 spatial directions can occur at these points. They are caused by forces and torques affecting the nodes in the 3 spatial directions.

The behaviour of a beam element is determined by the properties of its material (density ρ, matrix of modulus of elasticity E, matrix of modulus of shearing elasticity G) and the geometrical dimensions (length L, cross-sectional area A=b*h, volume V). Furthermore the torsional planar moment of inertia I_t, the planar moments of inertia I_m and I_n and also the polar moment of inertia I_p are required for the calculations. The displacements contained in **x** are denoted by w and the torsions by φ. The first index of these quantities indicates the direction of effects and the second index is the number of the node under consideration. Forces are denoted by *F* and torques by *M* comprising vector ***F***. The same index declarations as for displacements and torsions apply to both of them.

Subsequently there is a distinction between the coordinate system of the element (*l*, *m*, *n*) and the coordinate system of the entire system (*x*, *y*, *z*). Within one element the displacements and torsions are related to the coordinate system of the element, while after connecting the elements to form a more complex system, the position of the elements is considered with respect to the whole system's coordinate system.

On the assumption of the linear theory the stiffness matrix ***K*** is given by a 12*12 matrix. For nonlinear configurations FE codes use an additional stiffness matrix ***K****, which however shall not be taken into account in the further considerations.

In the static case the stiffness matrix ***K*** represents the relation between the vector of displacements ***x*** and vector of forces ***F***. The coupling between a bending around the *m*-axis and a displacement in *n*-direction (and vice versa) is taken into account in the ***K***-matrix.

The mass matrix ***M*** contains the rotational and translational inertia of the beam element. FE codes often use lumped inertia. This results in a sparse 12*12 matrix with 8 non zero elements, subsequently this shall be used as a basis for modeling in SABER.

In FE codes the damping matrix ***D*** can be defined arbitrarily using Rayleigh constants, a material dependent damping, a constant damping ratio, a modal damping or an element damping. In the following a damping matrix is

$$\boldsymbol{K} = \begin{bmatrix}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_n}{L^3} & 0 & 0 & 0 & \frac{6EI_n}{L^2} & 0 & -\frac{12EI_n}{L^3} & 0 & 0 & 0 & \frac{6EI_n}{L^2} \\
0 & 0 & \frac{12EI_m}{L^3} & 0 & -\frac{6EI_m}{L^2} & 0 & 0 & 0 & -\frac{12EI_m}{L^3} & 0 & -\frac{6EI_m}{L^2} & 0 \\
0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_m}{L^2} & 0 & \frac{4EI_m}{L} & 0 & 0 & 0 & \frac{6EI_m}{L^2} & 0 & \frac{2EI_m}{L} & 0 \\
0 & \frac{6EI_n}{L^2} & 0 & 0 & 0 & \frac{4EI_n}{L} & 0 & -\frac{6EI_n}{L^2} & 0 & 0 & 0 & \frac{2EI_n}{L} \\
-\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12EI_n}{L^3} & 0 & 0 & 0 & -\frac{6EI_n}{L^2} & 0 & \frac{12EI_n}{L^3} & 0 & 0 & 0 & -\frac{6EI_n}{L^2} \\
0 & 0 & -\frac{12EI_m}{L^3} & 0 & \frac{6EI_m}{L^2} & 0 & 0 & 0 & \frac{12EI_m}{L^3} & 0 & \frac{6EI_m}{L^2} & 0 \\
0 & 0 & 0 & -\frac{GI_t}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GI_t}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_m}{L^2} & 0 & \frac{2EI_m}{L} & 0 & 0 & 0 & \frac{6EI_m}{L^2} & 0 & \frac{4EI_m}{L} & 0 \\
0 & \frac{6EI_n}{L^2} & 0 & 0 & 0 & \frac{2EI_n}{L} & 0 & -\frac{6EI_n}{L^2} & 0 & 0 & 0 & \frac{4EI_n}{L}
\end{bmatrix}$$

$$\boldsymbol{x} = \begin{bmatrix} w_{l1} \\ w_{m1} \\ w_{n1} \\ \varphi_{l1} \\ \varphi_{m1} \\ \varphi_{n1} \\ w_{l2} \\ w_{m2} \\ w_{n2} \\ \varphi_{l2} \\ \varphi_{m2} \\ \varphi_{n2} \end{bmatrix} \qquad \boldsymbol{F} = \begin{bmatrix} F_{l1} \\ F_{m1} \\ F_{n1} \\ M_{l1} \\ M_{m1} \\ M_{n1} \\ F_{l2} \\ F_{m2} \\ F_{n2} \\ M_{l2} \\ M_{m2} \\ M_{n2} \end{bmatrix}$$

$$\boldsymbol{M} = \begin{bmatrix} \rho V & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & \rho V & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & \rho V & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & \frac{\rho L I_p}{2} & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & \rho V & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & \rho V & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & \rho V & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & \frac{\rho L I_p}{2} & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} d_1 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & d_2 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & d_3 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & d_4 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & d_1 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & d_2 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & d_3 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & d_4 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \\ 0 & 0 & 0 & 0 & 0\,0 & 0 & 0 & 0 & 0 & 0\,0 \end{bmatrix}$$

used having the same locations of nonzero elements as the mass matrix.

Once the matrices for each element are provided, the stiffness, damping and mass matrices of the entire system can be set up. The stiffness matrix of the entire system is formed of the stiffness matrices of the elements and so on. To provide the system's matrices, a coordinate transformation from the element's coordinate system (l,m,n) to the global coordinate system (x,y,z) is necessary in most cases. Then the solution of the system of equations is carried out using the appropriate boundary conditions like node forces and torques, displacements and torsions.

3 Modeling for SABER

The objective of the modeling for SABER is to provide basic mechanical elements, using the analog behavioural description language MAST. A more complex system like e. g. an acceleration sensor will be composed of this basic elements. In this section the derivation of a model is explained, using the spatial beam as an example. A variety of mechanical systems can be composed in the same way.

As shown in fig. 2 the spatial beam has 2 mechanical pins, each having 6 degrees of freedom. Consequently the model of the beam for the network analysis has 12 terminals, 6 for each mechanical pin. These can be subdivided into 3 translational (labeled t) and 3 rotational terminals (labeled r). Each terminal has corresponding through and across variables. The translational terminals use the force F as through variable and the displacement w as across variable. The rotational terminals use the torque M as through and the angle of rotation φ as across variable. Terminals, across and through variables are provided with 2 indices. The first index indicates the direction of the effect and the second index the node under consideration. Fig. 3 shows the terminal definition of the spatial beam element.

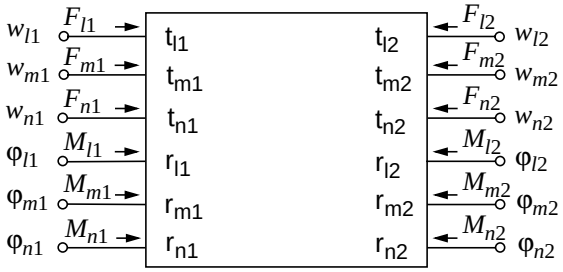


Fig. 3: Terminal definition of the spatial beam element

The behaviour of the beam element at the terminals can be described by equation (1), if ***M***, ***K***, and ***D*** represent the element matrices. As a result a beam can be composed of an arbitrary number of beam elements. The restraint conditions at the ends of the beam can be modeled by external wiring. A fixed restraint, leading to zero displacements and torsions at a distinct node, is achieved if all terminals of this node are clamped to ground. An open restraint, with vanishing forces and torques, is simply modeled by open terminals on that end. Furthermore models of fragments of beams have been provided, consisting of 2, 4, 6 and 8 basic elements.

4 Connection Elements

The terminal behaviour of a beam element is described using terminal quantities in a local coordinate system. This local coordinate system is fixed relative to the beam. In order to connect beam elements not arranged along a straight line, but joining another at a certain angle, a connection element is necessary to provide the coordinate transformation from one element coordinate system to the other. This is necessary to enable the simulator to satisfy the equilibrium conditions for forces and torques at the connecting node.

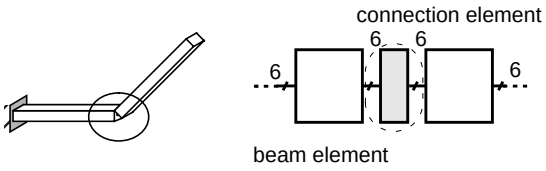


Fig. 4: Modeling of the connection of beam elements with different directions

The terminal behaviour of the connection element is described by the transformation matrix **T**. This matrix is given by the angles between the axes of the local coordinate systems of the two connected beam elements:

$$\boldsymbol{T} = \begin{bmatrix} \cos(l2, l1) & \cos(m2, l1) & \cos(n2, l1) \\ \cos(l2, m1) & \cos(m2, m1) & \cos(n2, m1) \\ \cos(l2, n1) & \cos(m2, n1) & \cos(n2, n1) \end{bmatrix}$$

cos(*l*2, *l*1) is the cosine value of the angle between the l-axes of the coordinate systems at terminal 2 and 1 of the connection element. The other elements of **T** have to be determined accordingly. Using the same variable definitions as in fig. 3 the terminal quantities of the connection element have to meet the following relations:

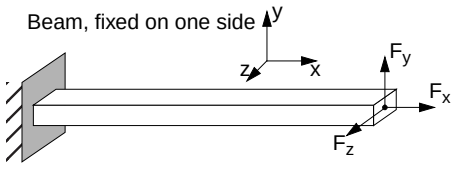
$$\begin{bmatrix} w_{l1} \\ w_{m1} \\ w_{n1} \end{bmatrix} = \boldsymbol{T} \cdot \begin{bmatrix} w_{l2} \\ w_{m2} \\ w_{n2} \end{bmatrix} \qquad \begin{bmatrix} \varphi_{l1} \\ \varphi_{m1} \\ \varphi_{n1} \end{bmatrix} = \boldsymbol{T} \cdot \begin{bmatrix} \varphi_{l2} \\ \varphi_{m2} \\ \varphi_{n2} \end{bmatrix}$$

$$\begin{bmatrix} F_{l1} \\ F_{m1} \\ F_{n1} \end{bmatrix} = \boldsymbol{T} \cdot \begin{bmatrix} F_{l2} \\ F_{m2} \\ F_{n2} \end{bmatrix} \qquad \begin{bmatrix} M_{l1} \\ M_{m1} \\ M_{n1} \end{bmatrix} = \boldsymbol{T} \cdot \begin{bmatrix} M_{l2} \\ M_{m2} \\ M_{n2} \end{bmatrix}$$

Basing on these relations a SABER-model of connection elements was developed.

5 Model Validation

A first validation of the modeling approach was carried out, using a simple cantilever beam described in [16]. This test beam is shown in fig. 5. It consists of p+ doped Silicon and is manufactured using anisotropic etching techniques.



Height of beam	$h = 0.2 \cdot 10^{-6}$ m
Width of beam	$b = 5.0 \cdot 10^{-6}$ m
Length of beam	$L = 160.0 \cdot 10^{-6}$ m
Density	$\rho = 2326$ kg/m^3
Modulus of elasticity	$E = 1.302 \cdot 10^{11}$ N/m^2
Modulus of shear	$G = 79.62 \cdot 10^{9}$ N/m^2
Mass	$m = 3.7216 \cdot 10^{-13}$ kg

Fig. 5: Cantilever beam for test purposes

This simple structure was chosen, because it is possible to obtain an analytical solution as a reference. ANSYS-simulations of the beam were also done for comparison, using spatial beam elements from ANSYS respectively. The beam was subdivided into 100 elements along its length.

To test the static behaviour a force of $7.3*10^{-4}$ µN was applied to the beam in all 3 directions. Table 1 compares the results of the analytical calculations and the static analysis of the ANSYS and SABER models. The maximal displacement at the free end of the beam is considered. These maximal displacements can be calculated analytically ([4] or [7]) by the equations:

$$\Delta x = \frac{F_x L}{AE} \qquad \Delta y = \frac{F_y L^3}{3EI_y} \qquad \Delta z = \frac{F_z L^3}{3EI_z}$$

The results of the SABER simulation match quite good to the results of the finite element analysis and the analytical calculations. Because we are dealing with linear sys-

Table 1: Static analysis of the beam, fixed on one side.

Static Load [µN]	Analytical Solution [µm]	ANSYS (100 Elements) [µm]	SABER (2 Elements) [µm]	SABER (4 Elements) [µm]	SABER (8 Elements) [µm]
$F_y=7.3*10^{-4}$	2.295	2.298	2.296	2.296	2.296
$F_z=7.3*10^{-4}$	$3.672*10^{-3}$	$3.667*10^{-3}$	$3.674*10^{-3}$	$3.674*10^{-3}$	$3.674*10^{-3}$
$F_x=7.3*10^{-4}$	$8.964*10^{-7}$	$8.971*10^{-7}$	$8.971*10^{-7}$	$8.971*10^{-7}$	$8.971*10^{-7}$

tems, displacements of single elements are added and thus there is no difference between models with a different number of elements.

The example from fig. 5 is also used to verify the dynamic properties of the models generated with our modeling approach. For this purpose the natural frequencies of a beam can be calculated analytically ([4], [7] or [20]). The longitudinal vibration is described by the partial differential equation:

$$\frac{\partial^2}{\partial t^2} w_x(x,t) = \frac{E}{\rho} \frac{\partial^2}{\partial x^2} w_x(x,t)$$

$w_x(x,t)$ is the longitudinal displacement. Considering the special boundary conditions in this case, one can obtain the well known natural frequencies:

$$\omega_k = \left(k - \frac{1}{2}\right) \frac{\pi}{L} \sqrt{\frac{E}{\rho}} \qquad k = 1, 2, \ldots$$

Bending vibrations are described by:

$$\frac{\partial^2}{\partial t^2} w_y(x,t) = - \frac{EI_y}{\rho A} \frac{\partial^4}{\partial x^4} w_y(x,t)$$

$w_y(x,t)$ is the lateral displacement in y-direction. Using I_z instead of I_y yields the lateral displacement $w_z(x,t)$ in z-direction. Boundary conditions result in the eigenvalue equation $\cosh(\lambda_k)\cos(\lambda_k) = -1$ for the spatial eigenvalues λ_k. Taking the solutions λ_k the natural frequencies are:

$$\omega_k = \sqrt{\frac{EI_y}{\rho A}} \frac{\lambda_k^2}{L^2} \qquad k = 1, 2, \ldots$$

Torsional vibrations are described by:

$$\frac{\partial^2}{\partial t^2} \varphi(x,t) = \frac{GI_t}{\rho I_p} \frac{\partial^2}{\partial x^2} \varphi(x,t)$$

$\varphi(x,t)$ is the angular displacement. Natural frequencies are:

$$\omega_k = \left(k - \frac{1}{2}\right) \frac{\pi}{L} \sqrt{\frac{GI_t}{\rho I_p}} \qquad k = 1, 2, \ldots$$

Some of the analytical natural frequencies are shown in table 2.

It was tested how far the generated SABER models match the natural frequencies of vibration. Therefor AC-analyses were carried out in SABER. In a first simulation run the beams with 2, 4 and 8 elements were stimulated using a sinusoidal force F_y of $7.3*10^{-6}$ µN in y-direction and the frequency range was swept from 8 kHz to 1.2 MHz. The force was chosen 100 times smaller compared to table 1, due to the resonance effects expected. The results of the simulation are shown in fig. 6. The 2 element beam can model 2 natural frequencies, the 4 element beam 4 natural frequencies and the 8 element beam 8 natural frequencies. As known from finite element calculations, the precision of the models increases with the number of elements. Furthermore it can be seen that lower natural frequencies match better than higher ones. The reason for this is, that one beam element is able to model one region of curvature only. Higher order natural shapes of vibration are composed of a larger number of regions with different curvature and are therefore approximated with less accuracy.

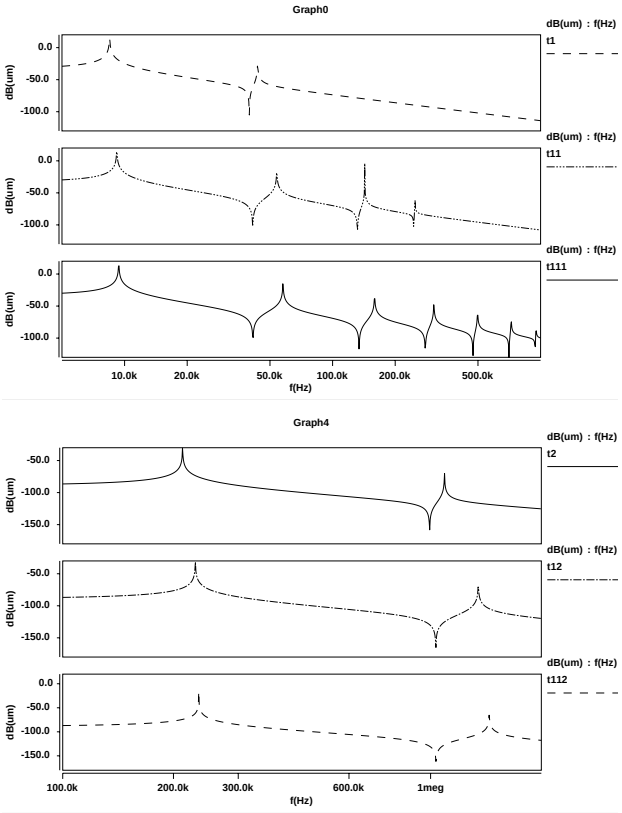


Fig. 6: Frequency response of the 2, 4 and 8 element beams stimulated by force in y-direction (top) and z-direction (bottom) respectively

Within the second simulation run a force F_z of the same value ($7.3*10^{-6}$ µN) was used as stimulus. The frequency range was swept from 100 kHz to 2 MHz. The beam has 2 natural shapes of vibration in this range, being modeled using the three different models. The natural shapes of vibration, generated by torsion around the x-axis, can be simulated using the provided models as well. Fig. 7 shows the results of a simulation with a torque M_x as excitation in the frequency range from 500 kHz to 1 MHz.

Table 2 compares the natural frequencies calculated by ANSYS, the natural frequencies of the provided SABER models and the natural frequencies of the analytical solution. The errors are always refered to the analytical solution.

Table 2: Modal analysis

Mode	Stim.	Analytic Solution [kHz]	ANSYS (100 Elements) [kHz]		SABER (2 Elements) [kHz]		SABER (4 Elements) [kHz]		SABER (8 Elements) [kHz]	
1.	F_y	9.4458	9.438	0.08%	8.47	10.3%	9.174	2.88%	9.374	0.76%
2.	F_y	59.196	59.143	0.09%	43.638	26.3%	53.906	8.94%	57.732	2.47%
3.	F_y	165.750	165.606	0.09%			142.8	13.8%	159.16	3.98%
4.	F_z	236.144	236.007	0.06%	211.63	10.4%	229.488	2.82%	234.35	0.76%
5.	F_y	324.803	324.539	0.08%			248.91	23.4%	307.0	5.48%
6.	F_y	536.923	536.564	0.07%					497.89	7.27%
7.	M_x	730.385			709.59	2.85%	725.07	0.73%	729.74	0.09%
8.	F_y	802.070	801.737	0.04%					721.86	10.0%
9.	F_y	1120.25	1120.0	0.02%					950.61	15.1%
10.	F_z	1479.89	1480.0	0.01%	1091.3	26.3%	1346.2	9.03%	1442.4	2.53%
11.	F_y	1491.45	1519.8	1.90%					1128.5	24.3%

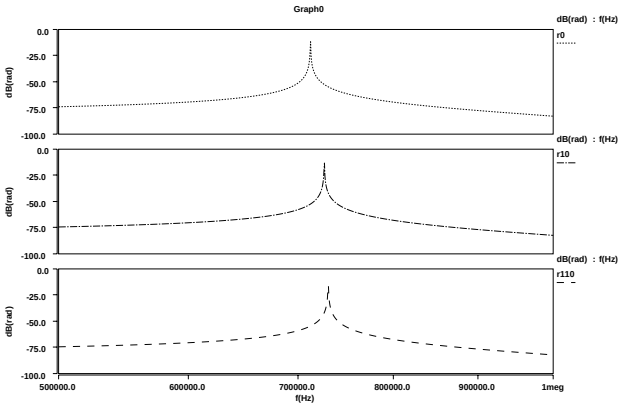


Fig. 7: Frequency response of the 2, 4 and 8 element beams stimulated by torque around the *x*-axis

6 Conclusion

The applicability of the introduced modeling approach to include mechanical components into the system simulation with SABER was shown using a resonating structure as an example. The approach gives an accurate description of the static and dynamic behaviour of linear spatial beams as well as more complex configurations constructed from those beams. The application of the described methods in addition with electrostatic comb drives and capacitive readout is discussed in separate publications ([6] and [12]). [6] uses a surface micromachined acceleration sensor as an example and [12] a vibrating angular rate gyroscope.

Acknowledgement

This work was supported by the german government, Department of Education and Research (BMBF), under grant number 16SV236/0.

References

[1] Arimoto, S.; Nakayama, T.: "Another Language for Describing Motions of Mechatronics Systems: A Nonlinear Position-Dependent Circuit Theory," *IEEE/ASME Transactions on Mechatronics* 1(1996)2, pp. 168-180.

[2] Bathe, K.-J., *Finite-Elemente-Methoden*, Springer Verlag, Berlin, 1990.

[3] Becker, U.; Schwarz, P.: "Modellierung und Simulation nichtelektrischer Systeme," part 1 and 2, *Elektronik* 1/97, 3/97, pp. 70.

[4] Dubbel; (Eds.: Beitz, W.; Küttner, K.-H.): *Taschenbuch für den Maschinenbau.* Springer Verlag 1987.

[5] Fedder, G. K.; Howe, R. T.: "Multimode Digital Control of a Suspended Polysilicon Microstructure", *Journal of Microelectromechanical Systems* 5 (1996) 4, pp. 283-297.

[6] Haase, J.; Reitz, S.; Schwarz, P.; Becker, U.; Lorenz, G.; Neul, R.: “Netzwerk- und Verhaltensmodellierung eines mechanischen Beschleunigungssensors”, *Proc. 6. Workshop: Methoden und Werkzeuge zum Entwurf von Mikrosystemen*, Paderborn, 1997, pp. 23-30.

[7] Hütte; (Ed.: Czichos, H.): *Die Grundlagen der Ingenieurwissenschaften*, Springer Verlag, Berlin, 1989.

[8] Knothe, K.; Wessels, H.: *Finite Elemente*, Springer Verlag, Berlin, 1992.

[9] Koenig, H. E.; Blackwell, W. A.: *Electromechanical System Theory*, McGraw-Hill, New York, 1961.

[10] Lenk, A.: *Elektromechanische Systeme* (Band 1 und 2), Verlag Technik, Berlin, 1973, 1974.

[11] Lenning, L. G.; Shah, A.; Özgüner, Ü.; Bibyk, S. B.: “Integration of VLSI Circuits and Mechanics for Vibration Control of Flexible Structures”, *IEEE/ASME Transactions on Mechatronics*, 2(1997)1, pp. 30-40.

[12] Lorenz, G.; Neul, R.: “Integration von mikrosystemtechnischen Komponenten in die Netzwerk- und Systemsimulation”, *Proc. 6. Workshop: Methoden und Werkzeuge zum Entwurf von Mikrosystemen*, Paderborn, 1997, pp. 39-46.

[13] Meißner, U.; Menzel, A.: *Die Methode der finiten Elemente*, Springer Verlag, Berlin, 1989.

[14] Neul, R.: “Modellbildung für mikromechanische Beschleunigungssensoren”, *Proc. 5. Workshop: Methoden und Werkzeuge zum Entwurf von Mikrosystemen*, Paderborn, 1996, pp. 111-117.

[15] Pelz, G.; Bielefeld, J.; Zappe, F.-J.; Zimmer, G. et al.: “MEXEL: Simulation of Microsystems in a Circuit Simulator Using Automatic Electromechanical Modeling”, *Proc. MICRO SYSTEM Technologies ‘94*, VDE-Verlag, Berlin, 1994, pp. 651-657.

[16] Petersen, K.E.: “Silicon as a Mechanical Material”, *Proc. IEEE*, Vol. 70, no. 5, May 1982, pp. 420-457.

[17] Reinschke, K.; Schwarz, P.: *Verfahren zur rechnergestützten Analyse linearer Netzwerke*, Akademie-Verlag, Berlin, 1976.

[18] Senturia, S. D.; Aluru, N.; White, J.: “Simulating the Behavior of MEMS Devices: Computational Methods and Needs”, *IEEE Computational Science & Engineering*, January-March 1997, pp. 30-43.

[19] Smits, J. G.; Ballato, A.: “Dynamic Admittance Matrix of Piezoelectric Cantilever Mimorphs”, *Journal of Microelectromechanical Systems* 3 (1994) 3, pp. 105-112.

[20] Woodson, H. H.; Melcher J. R.: *Electromechanical Dynamics*, John Wiley & Sons, New York, 1968.

[21] Wünsche, S., Schwarz P., Becker, U.; Neul, R.: “Ein Modellierungsansatz zur Einbeziehung mechanischer Mikrosystemkomponenten in die Systemsimulation”, *Proc. 5. Workshop: Methoden und Werkzeuge zum Entwurf von Mikrosystemen*, Paderborn, 1996, pp. 23-29.