A system level model in VHDL-AMS for a micromechanic vibration sensor array

Michael Schlegel, Göran Herrmann, Dietmar Müller
Department of Electrical Engineering and Information Technology,
Chemnitz University of Technology,
Chemnitz, Germany
michael.schlegel@infotech.tu-chemnitz.de
herrmann@infotech.tu-chemnitz.de
dietmar.mueller@infotech.tu-chemnitz.de

Abstract

A new capacitive vibration sensor using an array of laterally moving mass-spring systems is being developed at Chemnitz University of Technology [1]. The sensor operation is based on narrow-band resonance of the mass-spring elements. The natural frequency of each element can be tuned electrically. The sensor is intended for application in wear state recognition on highly stressed machine components.

The paper is focussing on high abstraction level CAD modeling of the sensor array. A new design approach [2], efficient choice of physical domains, resulting problems and their solutions will be shown in connection with the design of both sensor and analog signal processing models.

Keywords

sensor array, modeling, VHDL-AMS, design method

INTRODUCTION

Wear state recognition on machine tools during the normal work process is an important fundamental of product quality improvement. Today this is done by measuring the forces on the tool. An alternative method is to control machine and tool vibrations. Approaches to tool vibration measurement have existed for many years. But these approaches are using pieco ceramic wide band sensors which are quite expensive [4],[5].

The new sensor array which is being developed at Chemnitz University of Technology is fabricated by a SCREAM technology. Because of the narrow-band resonance working mode, the signal processing can be simplified due to a better signal to noise ratio and because no FFT is needed. So a low cost sensor system can be built.

The sensor system should be tested using the experimental prototype „vibration sensor array". But before the construction of the experimental prototype a simulation of the whole system is necessary to check the functionality of the individual components and their interaction. The function of the sensor was simulated by FEM simulation during the development of the sensor array. For simulation of the system including the sensor, analog and digital signal processing an FEM simulation takes too much time. Additionally, a simulation environment is necessary which allows simulation of mechanical, analog electrical and digital systems. So a model at a high abstraction level was developed using the VHDL-AMS hardware description language. The paper presents different variants of the sensor model, an analog signal processing model, an environmental model which emulates the environment of the sensor, and the simulation results.

THE SYSTEM EXPERIMENTAL PROTOTYPE „VIBRATION SENSOR ARRAY"

The sensor system consists of a sensor array containing eight individual mass-spring resonators, with an electrically tunable natural frequency each, an analog signal processing unit and a high voltage amplifier. The system is controlled by a micro controller. The system also includes a fuzzy pattern classification system [3].

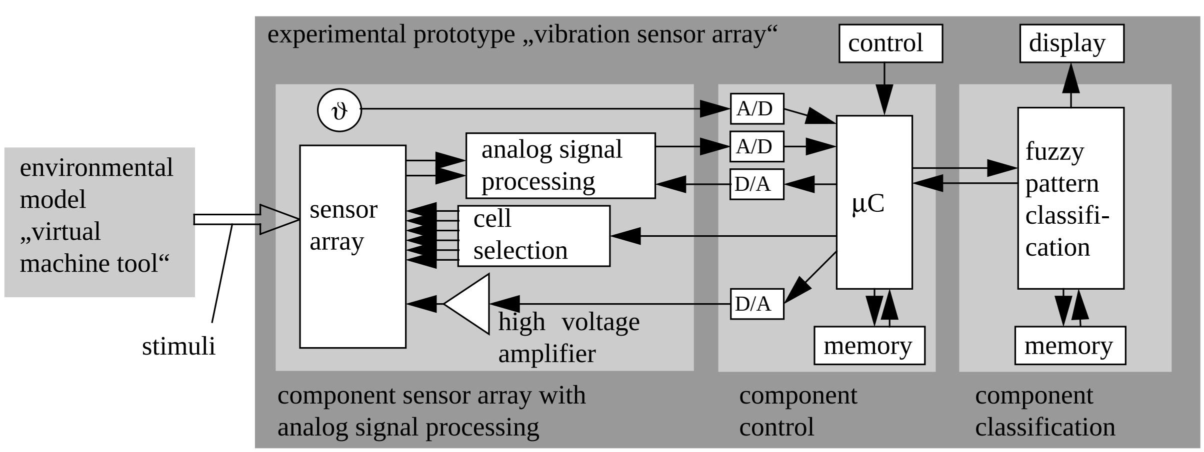

Figure 1. Block diagram of the experimental prototype „vibration sensor array"

An environmental model called „virtual machine tool“ provides the stimuli for simulation. It reproduces measured data in time or frequency domain or generates fictitious data. The vibration sensor converts this mechanical stimulation frequency-selectively into an electrical signal. Analog signal processing amplifies this signal and extracts the magnitude at a specific frequency. The individual resonators in the sensor can be activated separately, in groups or all together by a cell activation unit. The high voltage amplifier generates voltages up to 30 V for the natural frequency tuning of the sensor. A micro controller starts or stops the measurement, activates or deactivates resonators and starts self-calibration. It tunes the natural frequency of the resonators and transmits measured data to the classification unit. The classifier decides by a fuzzy pattern classification algorithm whether the data are produced by an sharp or worn out tool. This algorithm will be realized as FPGA.

MODELS OF THE SENSOR

The sensor consists of an array of laterally moving mass-spring resonators. They work in a frequency-selective resonant mode. To allow measurements at variable frequencies the spring constant and, therewith the natural frequency is tuned by an electrostatic force. The response of the structures is detected capacitively by comb electrodes at the seismic mass. Figure 2. shows a photograph of a test version of the sensor array.

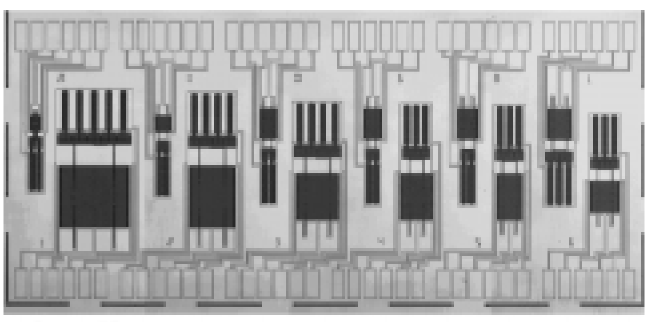

Figure 2. Photograph of a test version of the vibration sensor

Tuning by stress-stiffening

The first variant of tuning the natural frequencies is using the stress-stiffening effect. During its development the sensor was simulated and optimized by FEM simulation. But for system simulation, FEM simulation is too slow. So the sensor was modeled and simulated in the VHDL-AMS hardware description language. Figure 3. shows a schematic of the model. It consists of 8 springs, seismic masses and dampers. The advantage of using VHDL-AMS is that these components can be modeled in the mechanical domain directly without any analogy transformation. This means that the behavior of a spring can be described as $F = k \cdot s$ (force, spring constant, displacement). With a SPICE-like analogy transformation the modeling of the mechanical behavior is limited to electrical equivalents whereas in VHDL-AMS it is possible to describe every linear and nonlinear behavior between force and acceleration, velocity or displacement.

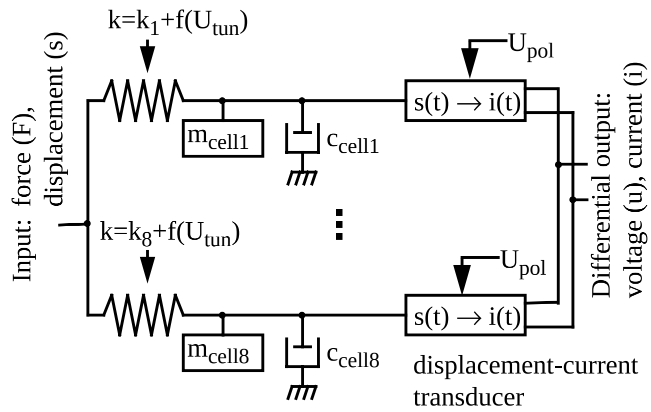

Figure 3. Schematic of the sensor array

Figure 4. shows the functional principle of the mass-spring system and the natural frequency tuning by the stress-stiffening effect.

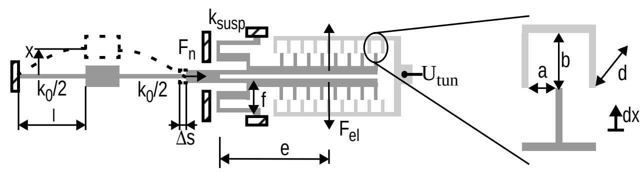

Figure 4. Extracted detail of a mass-spring resonator

The stress-stiffening unit is driven by an electrostatic force F_{el}. This force is caused by the voltage U_{tun} and can be calculated as follows:

$$F_{el} = \frac{dW}{dx}, \quad dW = \frac{1}{2} \cdot U_{tun}^2 \cdot dC \qquad \text{(eq. 1)}$$

The capacitance of the comb structure can be calculated by the simplification of a homogeneous field of a plate capacitor:

$$C = 2n \cdot \varepsilon \cdot \frac{x_{ss} \cdot d}{a}, \quad \frac{dC}{dx} = 2n \cdot \varepsilon \cdot \frac{d}{a} \qquad \text{(eq. 2)}$$

$$F_{el} = U_{tun}^2 \cdot \frac{n \cdot \varepsilon \cdot d}{a} \qquad \text{(eq. 3)}$$

where n is the number of combs. The force F_{el} generates by a lever mechanism (e, f) a normal force F_n [1].

$$F_n = 2 \cdot \frac{e}{f} \cdot F_{el} + \frac{6 \cdot k_{susp}}{5 \cdot l} \cdot x^2 \qquad \text{(eq. 4)}$$

This force influences the spring constant by the stress-stiffening effect:

$$k = k_0 + \frac{12 \cdot F_n}{5 \cdot l} \qquad \text{(eq. 5)}$$

The following part of the source code shows the implementation of this stress-stiffening effected spring in VHDL-AMS.

```
ENTITY stress_stiffening_spring IS
   GENERIC(n,a,b,d,e,f,k0,ksusp,l,eps:real);
   PORT( TERMINAL t1,t2: mech_F_s;
         TERMINAL t_tun: electrical);
END;

ARCHITECTURE behav of stress_stiffening_spring IS
   QUANTITY x ACROSS f THROUGH t1 TO t2;
   QUANTITY u_tun ACROSS t_tun;

   QUANTITY fel,fn: real:=0.0;
   QUANTITY k: real:=k0;
BEGIN
   fel==u_tun**2 * eps*n*d/a;
   fn==2.0*2.0*e/f*fel + 6.0*ksusp/5.0/l*x**2
      --two stress-stiffening actuators per spring
   k==k0+12.0/5.0*fn/l;

   f==k*x;
END;
```

Tuning by an electrostatic spring

Tuning by stress-stiffening has two disadvantages. The fabrication of this structure is difficult because of strain in the layers. Additionally, the lever mechanism is causing nonlinearities in the behavior (eq. 4). So an alternative approach was tested. In this approach the spring constant is kept constant. The tuning of the natural frequency is done by a curved comb structure where the attraction force is a linear function of the displacement.

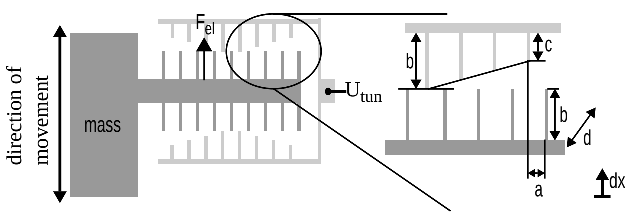

Figure 5. Extracted detail of the electrostatic softening

The curve was optimized by FEM simulation. For system simulation it can be assumed to be a linear function. Analogous to equation (eq. 1) the force in the electrical field of this comb structure can be calculated as follows:

$$F_{el} = \frac{1}{2} \cdot U_{tun}^{\ 2} \cdot \frac{dC}{dx} \qquad \text{(eq. 6)}$$

The capacitance of this comb structure can be calculated by the simplification of the homogeneous field of a capacitor within the range $-(b-c) \le x \le (b-c)$:

$$C = 2 \cdot \varepsilon \cdot \frac{d \cdot x}{a} \cdot \frac{x}{b-c} \cdot n \cdot \frac{1}{2} \qquad \text{(eq. 7)}$$

$$C = \varepsilon \cdot \frac{d \cdot x^2}{a} \cdot \frac{n}{b-c} \qquad \text{(eq. 8)}$$

$$\frac{dC}{dx} = 2 \cdot \varepsilon \cdot \frac{d \cdot x}{a} \cdot \frac{n}{b-c} \qquad \text{(eq. 9)}$$

$$F_{el} = U_{tun}^{\ 2} \cdot \varepsilon \cdot \frac{d \cdot x}{a} \cdot \frac{n}{b-c} \qquad \text{(eq. 10)}$$

Because F_{el} and the force of an ordinary spring are acting in opposite directions, the following can be written:

$$F = k \cdot x\,,\ k = -\left(U_{tun}^{\ 2} \cdot \varepsilon \cdot \frac{d}{a} \cdot \frac{n}{b-c}\right) \qquad \text{(eq. 11)}$$

And in the range $|x| > (b-c)$ applies:

$$C = 2n \cdot \varepsilon \cdot \frac{(x-(b-c)) \cdot d}{a} + C_0 \qquad \text{(eq. 12)}$$

$$\frac{dC}{dx} = 2n \cdot \varepsilon \cdot \frac{d}{a} \qquad \text{(eq. 13)}$$

$$F_{el} = U_{tun}^{\ 2} \cdot n \cdot \varepsilon \cdot \frac{d}{a} \text{ for } x > (b-c) \qquad \text{(eq. 14)}$$

$$F_{el} = -U_{tun}^{\ 2} \cdot n \cdot \varepsilon \cdot \frac{d}{a} \text{ for } x < -(b-c) \qquad \text{(eq. 15)}$$

Figure 6. shows the structure of the sensor array model with this kind of tuning the natural frequency.

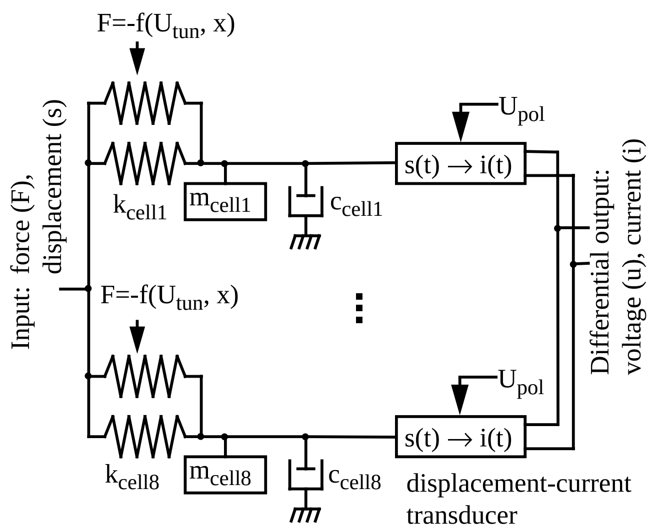

Figure 6. Schematic of the sensor with electrostatic softening

Displacement-current transducer

The displacement-current transducer is also made of a curved comb structure which was optimized by FEM. For system simulation this curve was neglected because of the low curvature.

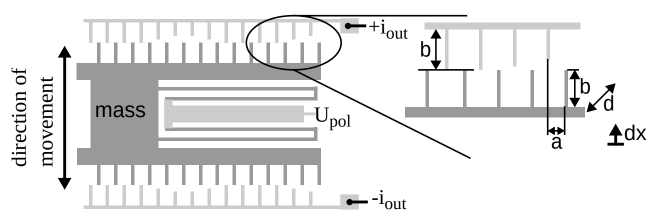

Figure 7. Displacement-current transducer

The currents caused by movement can be calculated by simplification of a homogeneous field of a capacitor as follows:

$$i(t) = U_{pol} \cdot \frac{dC}{dt} \qquad \text{(eq. 16)}$$

$$C = 2n \cdot \varepsilon \cdot \frac{x \cdot d}{a}, \quad \frac{dC}{dt} = 2n \cdot \varepsilon \cdot \frac{d}{a} \cdot \frac{dx}{dt} \qquad \text{(eq. 17)}$$

$$i(t) = U_{pol} \cdot 2n \cdot \varepsilon \cdot \frac{d}{a} \cdot \frac{dx}{dt} \qquad \text{(eq. 18)}$$

ANALOG SIGNAL PROCESSING

Analog signal processing includes a current to voltage converter and a differential amplifier which amplifies the low output current from the transducer. The usage of a differential output avoids non-linearities of the output signal.

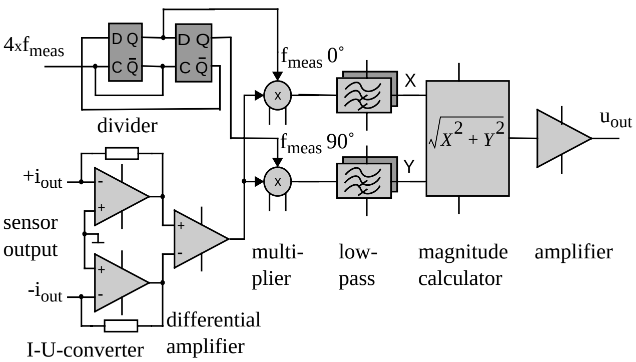

Figure 8. Block diagram of analog signal processing

The next part is a lock-in amplifier. This amplifier extracts the magnitude of the measured signal at a certain frequency.

For modeling this circuit a new design approach called **Multi Architecture Modeling** was used (for details see next section). The components were designed as functional blocks firstly. By using conservative nodes (*terminal*) and by insertion of power supply nodes even at this high abstraction level it is possible to exchange these abstract components partly or completely against models of real electrical elements.

SPECIAL MODELING METHODS

Deferred constants

Deferred constants are a powerful but seldom used feature of VHDL. As the source code above shows, even in this small model the parameter list (*generics*) contains numerous items. A *generic* list for the complete sensor model would contain about 50 items. For a wide range of applications the model must be parametrizable. But 50 parameters in a *generic* are very confusing, and putting these parameters into vectors is also not much better. The solution of this problem is to use deferred constants. This means that the parameters of the sensor are declared as a *constant* in a *package* without a default value. The default values of the constants are assigned in the *package body*. The parameters can be made visible in the model by using this *package*. If the constants would be declared with default values in the *package* a modification of one default value of a constant would make it necessary to recompile all models which use this *package*. By using deferred constants the value of the constant can be modified in the *package body* without the need to recompile anything else than the *package body*.

Multi architecture modeling

If MEMS are designed by a top down based method then digital, analog electrical and non-electrical models at different abstraction levels may appear during the design process. These components generally have interfaces to their environment. The abstraction level of these interfaces depends on the abstraction level of the component. If system models are developed within the scope of a system design then it could have been necessary until now to modify the interfaces of the system and component models at every design step of a component model. One solution of this problem is the Multi Architecture Modeling (MAM) [2]. The main idea of MAM is to use the interface, which will be necessary at low abstraction level, already at a high abstraction level. This avoids a lot of work because abstract models of a component can be replaced by detailed models or vice versa without any modification to the interface or the surrounding models.

This approach was applied in the development of the analog signal processing model. In the beginning, the components were modeled at a high abstraction level as functional blocks. Functional blocks normally communicate by non-conservative nodes (*quantity*). For correct results simulation as an electrical circuit (partly with simplified electrical elements) is necessary. An electrical circuit uses conservative nodes (*terminal*). So refined components cannot be inserted into a system model which is using *quantities*. With MAM the system model with functional blocks is using *terminals* as interface and it also contains nodes for power supply. This has no disadvantages as compared with the use of non-conservative nodes, whereas modeling overhead is limited. But now refined parts of the analog signal processing may be inserted into the system environment without any problems. This saves a lot of time and reduces error-prone adapting steps.

The domain for translatory movement

For the simulation of electrical components the use of current and voltage as characteristic *across* and *through quantity* goes without saying. But which *across* and *through quantity* should be used for the mechanical domain? If the design of mechanical components is done by an analogy transformation to electrical systems the *across* and *through quantity* is limited to velocity and force as analogy to voltage and current. In opposition to this, VHDL-AMS allows the description of mechanical behavior without analogy transformation. Table 1. shows possibilities for the *across* and *through quantity* in translatory behavior.

Table 1. Possible combinations of across and through quantities for translatory movement

across quantity	through quantity
displacement (s)	force (F)
velocity (v)	force (F)
acceleration (a)	force (F)

Even a description of displacement, velocity or acceleration as *through quantity* and force as *across quantity* is possible.

Measurement of vibration is normally done by measuring accelerations. So it was obvious to model the mechanical parts with acceleration and force. This is no problem until a resulting velocity or displacement is needed.

As is generally known:

$$v = \int a dt + C_0 , \qquad s = \int v dt + C_1 \qquad \text{(eq. 19)}$$

The integration constants C_0 and C_1 cause difficulties - they cannot be set explicitly by the VHDL-AMS `'integ` attribute. This, e. g., results in:

$$a = A \cdot \sin(\omega t) \qquad \text{(eq. 20)}$$

$$v = A \cdot \int \sin(\omega t) + C_0 = -\frac{A}{\omega} \cdot \cos(\omega t) + C_0 \qquad \text{(eq. 21)}$$

The simulator assumes:

$$v(0) = -\frac{A}{\omega} \cdot \cos(0) + C_0 , \quad C_0 = \frac{A}{\omega} \qquad \text{(eq. 22)}$$

This results in an offset of the calculated velocity which will be integrated when calculating the displacement. This means that in the simulation the sensor „pulls the machine tool away“.

The solution of the problem is to use displacement as *across quantity*. If a velocity or acceleration is needed then these values can be calculated definitely by derivation of displacement over time. The measured vibration data must be converted into displacement data but this can be done easily.

RESULTS

The models described above have been simulated by Mentor Graphics’ AdvanceMS on a SUN ULTRA60 workstation with UltraSPARC-II 296 MHz CPU.

The first step in system simulation was the simulation of the sensor array. Figure 9. shows the spring constant as a function of the tuning voltage with tuning by stress-stiffening effect. It can be seen easily that the spring constant increases with tuning voltage.

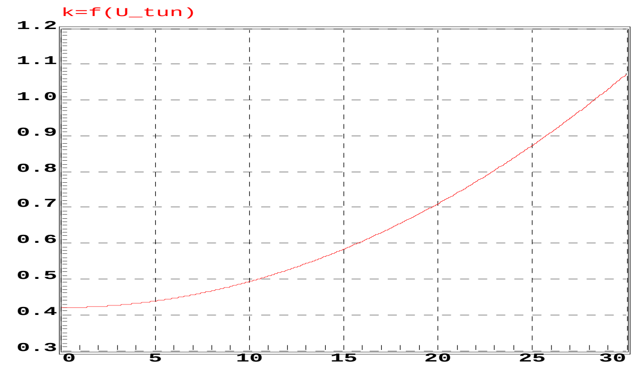

Figure 9. Spring constant with stress-stiffening effect as a function of tuning voltage

Figure 10. shows the spring constant of the electrostatic softening structure corresponding to equation (eq. 11).

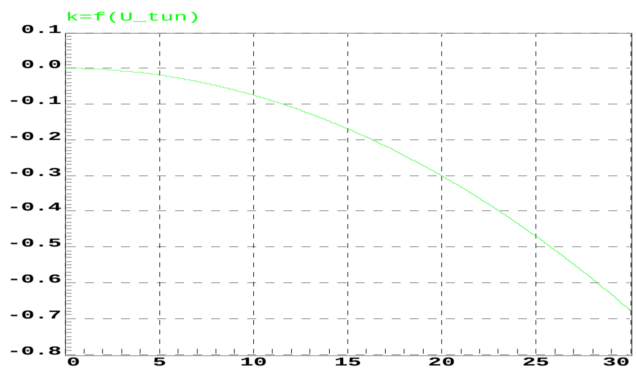

Figure 10. Spring constant of softening structure as a function of tuning voltage

It can be seen that the spring constant assumes a negative value. This means that this spring will produce an attraction force when it is being compressed. In opposition to this a normal spring would produce a repulsion force if compressed. For tuning the natural frequency of the mass-spring element this electrostatic softening structure is mounted parallel to the normal spring of the system (see Figure 6.) and so the spring constant of this structure can be modified by tuning voltage.

The simulation of these sensor parts required only a few seconds computing time. The simulation inaccuracy of the softening structure is 4 % as compared with the FEM simulation. For the stress-stiffening structure the error as compared with the FEM simulation is about 10 %. An error of 10 % may seem to be too high but for system simulation this accuracy should be sufficient.

The next steps in the simulation process were the simulation of the whole sensor array and the simulation of analog signal processing. Finally, the system consisting of sensor array, stimuli generation and analog signal processing was simulated completely. A simulation result can be seen in Figure 11. The first curve displays the stimuli of the sensor provided by the „virtual machine tool“. Curves two through four show the response of the first three mass-spring systems of the sensor array. The tuning voltage is set to 12 V so cell 1 is tuned to a natural frequency of 1 kHz. The last two curves show the electrical response of the sensor after passing the differential amplifier and the response of the lock-in amplifier.

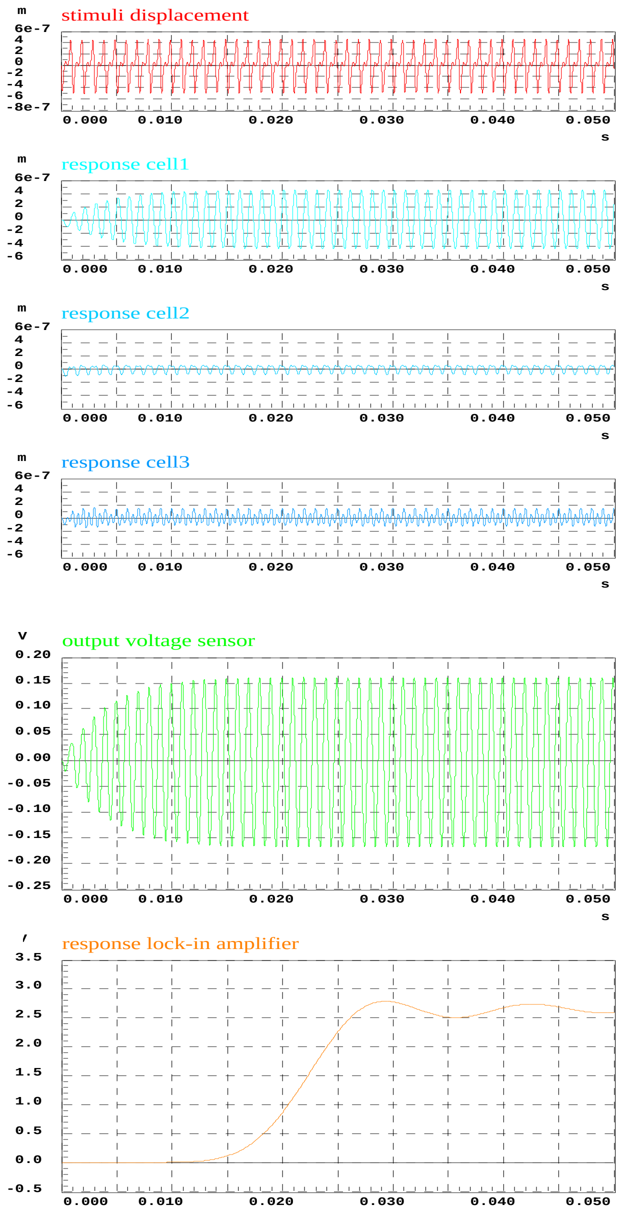

Figure 11. Stimuli and response of system simulation

The simulation time of this system is less than 5 minutes. In comparison to this, an FEM simulation takes about 20 minutes to simulate one comb structure only. The accuracy of the system simulation cannot be classified before the experimental prototype is build.

CONCLUSION AND OUTLOOK

Simulation of the sensor array in combination with analog signal processing and the environmental model allows to show the function of the system before the realization of the hardware. Due to the short simulation time an interactive optimization of the system is possible, and also the interaction between the components can be tested easily.

At the moment the models for the whole experimental prototype „vibration sensor array“ are not yet complete. The digital components „control“ and „fuzzy pattern classification“ [3] still have to be implemented. When this is done it will be possible to simulate whole measurement cycles in relative short times.

ACKNOWLEDGEMENTS

The work presented here has been done within project A2 „System Design“ of the SFB 379 (collaborative research center), which is funded by the German Science Foundation (DFG).

REFERENCES

[1] Mehner, J.; Scheibner, D.; Wibbeler, J.: *Silicon Vibration Sensor Arrays with Electrically Tunable Band Selectivity.* MICRO SYSTEM Technologies 2001, Düsseldorf, March 27-29, 2001, ISBN 3-8007-2601-7, pp. 267-272

[2] Schlegel, M.; Herrmann, G.; Müller, D.: *Multi Architecture Modeling Design Method for Mixed Signal and Multi Domain System Simulation - First Solutions*. The International MEMS Workshop 2001, Singapore, July 4-6, 2001, ISBN 981-04-4165-7, pp. 662-667

[3] Eichhorn, K.: *Entwurf und Anwendung von ASICs für musterbasierte Fuzzy-Klassifikationsverfahren.* Dissertation, Technische Univerität Chemnitz, 21. März 2000

[4] Strackeljan, J.; Behr, D.: C*ondition monitoring of rotating machinery using a fuzzy pattern recognition algorithm*. 6. International Conference on Vibrations in Rotating Machinery, September 9-12, 1996, ISBN 1-86058-009-2, pp. 507-516

[5] Lipovszky, G.; Solyomvari, K.; Varga, G.: *Vibration testing of machines and their maintenance.* Elsevier Science Publishers, Amsterdam 1990, ISBN 0-444-98808-4