Convergence and Speed Issues in Analog HDL Model Formulation for MEMS

Sitaraman Iyer[†], Qi Jing[†], Gary K. Fedder[†*] and Tamal Mukherjee[†]

[†]Department of Electrical and Computer Engineering and [*]The Robotics Institute
Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA
(Email: {sita, qjing, fedder, tamal}@ece.cmu.edu)

ABSTRACT

Behavioral simulation using high-level hardware description languages (HDLs) is becoming increasingly relevant for mixed-domain MEMS simulation. In this paper three issues which impact the convergence and simulation time in MEMS behavioral simulations are addressed. First, velocity and displacement are compared for the choice of the across variable in nodal simulation. The through variable is force. Second, three state space implementations of displacement as across variable are compared. Finally, relative scaling of different quantities in order to improve convergence properties is considered with respect to the rotational domain. A minimal equation matrix representation with low condition number using displacement as across variable and scaling of the rotational domain gives the best convergence and simulation time.

Keywords**:** behavioral modeling, analog hardware description language, convergence

1 INTRODUCTION

Researchers have encoded behavioral models of mechanical, electrostatic, optical and fluidic components in analog HDLs such as Verilog-A [1], MAST [2] and VHDL-AMS [3]. Analog HDLs provide a powerful methodology to combine different domains. Therefore, they are well-suited for integrated MEMS simulation.

Behavioral simulation provides the modeler with freedom to implement the physics of the component in a number of different ways. However, the high-level analog HDL code renders the final simulation matrix inaccessible to the modeler. Therefore, the choice of the best implementation is not immediately apparent to the modeler. Different implementations lead to different numbers of equations, convergence properties and simulation speed in transient analysis. Without a thorough understanding of the translation of the analog HDL code to the simulation matrix, the resulting simulation times may be non-optimal and the simulation may be non-convergent in the worst case. The available reference material mainly addresses syntax issues and does not provide insight into mapping of analog HDL representation to the equations for nodal analysis [4][5]. In this paper, analog HDL code is correlated to the matrix formulation during transient analysis. The convergence and simulation speed of transient analysis are then explained with the aid of the matrix formulation.

Comparisons between electrical circuit simulation and MEMS simulation for choice of *through* and *across* variables for nodal analysis have been suggested previously [3][6]. *Force* has been the preferred *through* variable. *Displacement* and *velocity* are possible candidates for the *across* variable. From a user perspective *displacement* as *across* variable is more convenient to observe motion. A preliminary evaluation of simulations using the two choices for the *across* variable showed that the *velocity* as *across* variable implementation has some convergence difficulties [7]. Simulation times have not been compared previously. A more comprehensive comparison of the convergence and simulation times for the two choices is presented here.

Mixed-domain behavioral simulation involves formulation of a set of matrix equations which may have widely different coefficients due to the different regions of interest in different domains. Wide range in matrix element values can lead to ill-conditioned matrices and thus to convergence difficulties. It is demonstrated in this paper that this problem can be eliminated by appropriate scaling of quantities.

The simulation environment is described in the background section. This is followed by an explanation of the different implementations along with the analog HDL code and the expected matrix implementation in the simulator. Scaling of quantities for better convergence is then discussed followed by simulation results and analyses. Finally, conclusions and suggestions for analog HDL modeling are presented.

2 BACKGROUND

The behavioral models in this paper are implemented as part of the NODAS [1] framework. NODAS is a library of parameterized models for elements such as *beam*, *plate*, and *comb-drive.* The user constructs a simulation representation of the device by composing a MEMS schematic using elements from this library. The differential equations for the elements in NODAS are encoded in Verilog-A. The Spectre™ [8] simulator from Cadence is used for simulations.

An elastically gimbaled z-axis CMOS-MEMS gyroscope [9], is used as the benchmark for simulations. For simulations the comb-drives are removed from the schematic in order to eliminate non-linear physical effects. Then, the gyroscope can be modeled as a linear nested spring-mass-damper system. By reducing the problem to a linear matrix, we focus on the mathematical representation of the system and its relation to convergence and speed of simulation.

3 MODEL FORMULATION

The Verilog-A module is based on *constitutive relationships*, which describe the behavior of the element, and *interconnection relationships*, which describe the structure of the network. The simulator combines *constitutive relationships* with *Kirchhoff's laws* in *nodal analysis* to form a system of differential-algebraic equations [10]. Numerical integration methods are employed to solve these equations for transient analysis. The simulator replaces the time derivative operator with a discrete-time finite difference approximation and solves at individual time points along the interval. Interval between time points (time step *h*) is controlled by simulator to ensure accuracy of the finite difference approximation.

Common integration methods include Backward Euler (BE), Trapezoidal rule (TR) and Gear. BE is used in this paper to illustrate the formulation of transient analysis matrix due to its simplicity, accuracy and stability. In BE, the node value at time *t+h* is computed based on the derivative value at *t+h*. For example, the equation $v = \dot{x}$ can be written as:

$$x(t) = \int_0^t v(\tau)d\tau \tag{1}$$

Solving for node variable *displacement x* using BE integration rule:

$$\frac{x(t)}{h} \cong \frac{x(t+h)}{h} - v(t+h) \tag{2}$$

The matrix representation is as shown in Implementation 1 of Section 3.1. Figure 1 is an illustration of BE integration method.

A mechanical spring-mass-damper system is governed by the equation: $F = Kx + B\dot{x} + M\ddot{x}$. Elastic, damping and inertial elements are discretized as described above and represented in this paper.

Five different behavioral model formulations of the above second order system are described below. Broadly they can be classified into two groups, one using *displacement* as the *across* variable and the other using *velocity* as the *across* variable, *force* being the *through* variable in all the cases. The latter bears a direct analogy to traditional electrical circuit simulation which uses *current* as the *through* variable and *voltage* as the *across* variable.

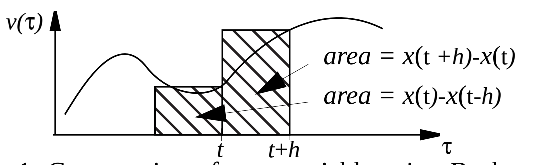


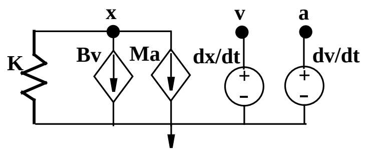
Figure 1. Computation of state variable using Backward Euler integration rule. The areas of the rectangles obtained by integration are shown.

Verilog-A code, equivalent circuit and matrix representations for transient analysis with BE are shown for each case. Extra state variables used by the modeler are listed in the beginning of the code. In addition, the simulator inserts additional states, some of which have trivial solutions. The non-trivial states inserted by the simulator are also included in the equivalent circuit and the matrix representations. The equivalent circuits represent the equations solved by the simulator at each time step. They are composed of conductances which enter the diagonal elements of the matrix and voltage controlled current sources which contribute to the off-diagonal terms in the transient analysis matrix.

3.1 *Displacement* as *across* variable

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

Verilog-A:
```
kinematic v, a;
Pos(v) <+ ddt(Pos(x));
Pos(a) <+ ddt(Pos(v));
F(x) <+ - M*Pos(a)
        - B*Pos(v)
        - K*Pos(x);
```
Equivalent circuit:

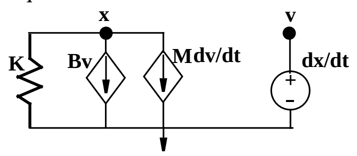


Matrix:

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

Implementation x2: One additional state (*v*) is used to hold velocity, leading to a more compact matrix. Note that the damping and inertia terms occur together in the matrix.

Verilog-A:
```
kinematic v;
Pos(v) <+ ddt(Pos(x));
F(x) <+ - M*ddt(Pos(v))
        - B*Pos(v)
        - K*Pos(x);
```
Equivalent circuit:

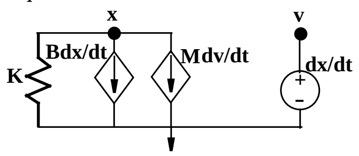


Matrix:

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

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

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



Matrix:

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

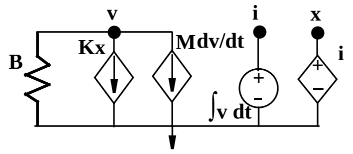
3.2 *Velocity* as *across* variable

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

Verilog-A:

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

Equivalent circuit:



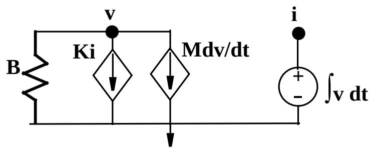
Matrix:

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

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

Verilog-A:

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



Matrix:

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

4 SCALING OF QUANTITIES

In mixed-domain simulation, implementation of behavioral models without insight into the simulation matrix can lead to ill-conditioned matrices. The wide variation of matrix elements is illustrated in Figure 2. The in-plane stiffness matrix for a typical MEMS beam is shown with and without scaling of the rotational domain. We can see that the resulting matrix is better conditioned with scaling.

5 SIMULATION RESULTS

Simulations of the gyroscope were done using the five different implementations. They are abbreviated as x1, x2, x3, v1 and v2. A 1 µN sinusoidal force was applied and transient analysis was done till 40 ms. The results of simulation are summarized in Table I.

The condition number for the matrix for a single spring-mass-damper goes asymptotically as *1/(Mh)* for x1 and as *(1/M)* for x2, x3, v1 and v2. This explains why x1 does not converge for the gyroscope simulation. x2 converges, but the result shows less damping than expected because the damping effects are shadowed by the *M/h* terms. Though the number of time-steps taken by v1 and v2 is nearly the same as those taken by x3, the overall time taken is larger.

$$\begin{array}{c|ccc} & \delta x & \delta y & \delta\theta \\ \hline Fx & \frac{EA}{l} & 0 & 0 \\ Fy & 0 & \frac{12EI}{l^3} & \frac{6EI}{l^2} \\ Mz & 0 & \frac{6EI}{l^2} & \frac{4EI}{l} \end{array}$$

(a) No scaling

$$\begin{bmatrix} 6600 & 0 & 0 \\ 0 & 2.64 & 1.32\times 10^{-4} \\ 0 & 1.32\times 10^{-4} & 8.8\times 10^{-9} \end{bmatrix}$$

Condition number: 3.0e12

(b) Scaling of $\delta\theta$ by 1e-6

$$\begin{bmatrix} 6600 & 0 & 0 \\ 0 & 2.64 & 132 \\ 0 & 132 & 8800 \end{bmatrix}$$

Condition number: 1.3e4

Figure 2. In-plane stiffness matrix for a beam [11]. Beam with *length* = 100 µm, *width* = 2 µm, *thickness* = 2 µm and Young's Modulus E = 165 GPa. A is the cross-section area and I is the moment of inertia. (a) The large span of the diagonal elements of the stiffness matrix is evident. (b) The stiffness matrix after scaling the rotational discipline by 10^{-6} has much smaller condition number.

Type	Conve rged	Correct	Time (min.)	No. of Eqns	Time-steps
x1	No	NA	NA	2751	NA
x2	Yes	No	180	1809	68811
x3	Yes	Yes	110	1809	59875
v1	Yes	Yes	133	2240	60023
v2	Yes	Yes	134	1556	60021

Table I Comparison of five implementations

6 CONCLUSIONS

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

1. Additional states ($a = \dot{v}$, $v = \dot{x}$) to hold derivatives lead to greater simulation times. They introduce large off-diagonal terms during the transient analysis and thereby lead to ill-conditioned matrices. Therefore, care must be taken when using additional states.
2. If additional states are not used, coefficients of derivatives (m) in the differential equations ($v = \dot{x}$, $F = kx + bv + m\dot{v}$) are multiplied by (*1/h*) during transient analysis. Coefficients of v and $\dot{v}$ must be separated or scaled such that (*1/h*) multiplication of the lower order coefficients (m) during transient analysis does not lead to complete insignificance of the higher order coefficient (b).
3. Appropriate scaling must be used when different domains are combined together so that the composite nodal analysis matrix remains well-conditioned.

ACKNOWLEDGEMENTS

This research effort is sponsored by the Defence Advanced Research Projects Agency (DARPA) and U. S. Air Force Research Laboratory, under agreement number F30602-97-2-0323 and F30602-99-2-0545 and in part by the National Science Foundation award CCR-9901171.

REFERENCES

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

[2] G. Lorenz, *Network Simulation of Microelectromechanical Systems*, Ph. D. Thesis, University of Bremen, December 1999

[3] P. Voigt, G. Schrag and G. Wachutka, "Electrofluidic full-system modelling of a flap valve micropump based on Kirchoffian Theory", *Sensors and Actuators A*, 66, pp. 9-14, 1998

[4] D. Fitzpatrick, I. Miller, *Analog Behavioral Modeling With the Verilog-A Language*, Kluwer Academic Publishers, November 1997

[5] MAST Reference Manual, Release 4.0, Analogy Inc. Beavorton OR, 1995

[6] B. Romanowicz, "Methodology for Modeling and Simulation of Microelectromechanical Systems"

[7] G. K. Fedder, "Mechanical Natures: A White Paper for the VHDL-AMS Interoperability Working Group", November 24, 1998

[8] Spectre User Guide, Cadence Design Systems, 555 River Oaks Parkway, San Jose, CA, http:// www.cadence.com

[9] H. Luo, G. K. Fedder and L. R. Carley, "An Elastically Gimbaled Z-Axis CMOS-MEMS Gyroscope," to be published in *Int. Sym. Smart Structures and Microsystems*, Hong Kong, October 19-21, 2000

[10] Spectre Manual: Affirma Verilog-A language Reference-nodal analysis, Cadence Design Systems, San Jose CA

[11] J.S. Przemieniecki, "Theory of Matrix Structural Analysis", Dover Publications Inc., New York, 1985