System Simulation of Mixed-Signal Multi-Domain Microsystems With Piecewise Linear Models

Steven P. Levitan, *Senior Member, IEEE*, José A. Martínez, *Student Member, IEEE*, Timothy P. Kurzweg, *Member, IEEE*, Abhijit J. Davare, *Student Member, IEEE*, Mark Kahrs, *Member, IEEE*, Michael Bails, *Student Member, IEEE*, and Donald M. Chiarulli

***Abstract*—We present a component-based multi-level mixed-signal design and simulation environment for microsystems spanning the domains of electronics, mechanics, and optics. The environment provides a solution to the problem of accurate modeling and simulation of multi-domain devices at the system level. This is achieved by partitioning the system into components that are modeled by analytic expressions. These expressions are reduced via linearization into regions of operation for each element of the component and solved with modified nodal analysis in the frequency domain, which guarantees convergence. Feedback among components is managed by a discrete event simulator sending composite signals between components. For electrical, and mechanical components, interaction is via physical connectivity while optical signals are modeled using complex scalar wavefronts, providing the accuracy necessary to model micro-optical components. Simulation speed vs. simulation accuracy can be tuned by controlling the granularity of the regions of operation of the devices, sample density of the optical wavefronts, or the time steps of the discrete event simulator. The methodology is specifically optimized for loosely coupled systems of complex components such as are found in multi-domain microsystems.**

***Index Terms*—Behavioral modeling, microelectromechanical (MEM) simulation, mixed-signal multi-domain (MSMD) simulation, modified nodal analysis (MNA), piecewise-linear (PWL) simulation, system simulation of microsystems.**

I. Introduction

THE development of an integrated design environment for mixed-signal multi-domain (MSMD) microsystems is motivated by a convergence of new integration techniques for optical, mechanical, and electronic devices and by consumer demand for system applications that require new functionality, higher performance, and lower cost. Next generation systems will combine digital very large scale integration (VLSI) technologies with sensors, actuators, communication, and control devices incorporating analog electronics, optics, and mechanics into a single system design. Current examples of commercial MSMD microsystems include automotive air-bag controls, digital projectors, and optical network switches.

Design tools that work in a single domain such as digital CMOS have traditionally relied on abstraction to manage scale and complexity. However, in this abstraction methodology there is an underlying assumption of domain expertise on the part of the designer. In MSMD systems, there are few designers with expert-level knowledge across domains as varied as optics, mechanics, and electronics. Thus, MSMD design tools must both abstract detail and provide a consistent cross-domain modeling methodology that can be accessible to a designer with limited knowledge of a specific domain.

The creation of simulation tools for MSMD systems is also difficult because these systems span the physical domains of electronics, photonics, and mechanics, as well as multiple orders of magnitude in both time and length scales. The difficulties are compounded by the fact that computational performance and accuracy are directly related to the level of detail in the underlying models.

Two commercial tools for mixed-signal design and development are from Coventor [1], based on Saber, and MEMSCAP [2], based on the HDL-A language. Both of these tools provide simulation support for electrical and mechanical components with extensions for optical, RF, and fluidic devices. Coventor's CoventorWare software suite also provides physics-based FEM/BEM simulation of components for detailed analysis and extraction for some macro-component models.

However, creating a complete design flow for MSMD systems still presents numerous challenges. Simulators must use consistent modeling methodologies across domains and between abstraction levels and also must exhibit fast yet accurate simulation at the behavioral and system levels. Furthermore, design flows typically depend on extraction from the physical level to the behavioral level via multiple runs of finite element or boundary element solvers. There is also a scarcity of synthesis methods, typically based on simple library model composition. Finally, there is a critical lack of metrology and validation of device, component, and system models.

In this paper, we address the first two of these concerns: a consistent modeling methodology across domains and fast yet accurate behavioral simulation models that span multiple domains. We focus on the three domains of optics, electronics, and

Manuscript received June 5, 2002; revised September 11, 2002. This work was supported in part by Defense Advanced Research Projects Agency, under Grant F49620-01-1-0536 and in part by National Science Foundation, under Grant C-CR9988319. This paper was recommended by Guest Editor G. Gielen.

S. P. Levitan, J. A. Martínez, M. Kahrs, and M. Bails are with the Department of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA 15261 USA (e-mail: steve@ee.pitt.edu; jmarti@ee.pitt.edu; kahrs@ee.pitt.edu; mikeb@ee.pitt.edu).

T. P. Kurzweg was with the Department of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA 15261 USA. He is now with Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA 19104 USA (e-mail: kurzweg@ece.drexel.edu).

A. J. Davare was with the Department of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA 15261 USA. He is now with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: davare@eecs.berkeley.edu).

D. M. Chiarulli is with the Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260 USA (e-mail: don@cs.pitt.edu).

Digital Object Identifier 10.1109/TCAD.2002.806604

mechanics, with an understanding that this methodology could be extended to other domains. The results presented here have been implemented in an MSMD modeling and simulation environment described below.

We begin with a background of this environment with emphasis on system-level and behavioral-component models. We then present our piecewise linear (PWL) behavioral simulation methodology followed by a description of our modified nodal analysis-based (MNA) fast solver for electronic and mechanical models. Next, we describe our techniques for fast optical signal propagation. Finally, we present a system-level modeling example of a digital display that incorporates optical, mechanical, and electronic components and demonstrates the multi-domain integration of behavioral component models.

II. Background

We identify three levels of abstraction for an MSMD design environment: the system level, which is concerned with the ensemble performance of complete systems composed of black box parameterized components; the behavioral or component level, which captures the input/output transformations within multi-domain components with an abstract description; and the physical or device level, which models the same transformation as a result of the physical processes that underlie the operation of the device.

A. System-Level Simulation

The simulation of multi-domain systems involves signals with different properties (e.g., voltage for electronics and intensity for optics) and with varied dynamics. The use of an object-oriented framework permits a large degree of abstraction and flexibility for the simulation of such systems [3]. At the highest level, the system is composed of component modules that are individually characterized and joined together by the mutual exchange of information. As shown in Fig. 1(a) each module, i, processes some vector of input messages, $X_i(t)$, updates its vector of internal state variables, $S_i(t)$, and generates sets of output messages. The nature of these messages can be optical, electrical, or mechanical. Using a discrete event simulator, each module's execution is based on the availability of new data values for its inputs [3]. The simulation scheduler provides the system with a buffering capability, which allows the system to keep track of all the messages arriving at one module when multiple input streams of data are involved. This allows modeling of dynamic systems where each component can have variable rates of consumed or produced data during simulation.

In general, the components come from a parameterized model library. Some examples include CMOS analog amplifiers, vertical-cavity surface-emitting lasers (VCSEL), micro-mechanical cantilevers, lenses, and microelectromechanical (MEM) mirrors. The components are modeled at the behavioral level where they are represented either by analytic expressions or as a tightly coupled network of elements (E_i) such as shown in Fig. 1(b). In either case, at the system level there is a loosely coupled network of tightly coupled component models. This corresponds well with the general structure of mixed-signal microsystems where multi-domain components interact with few signals, while, at the same time, the behavior of each component is based on its underlying physical processes. These models are discussed next.

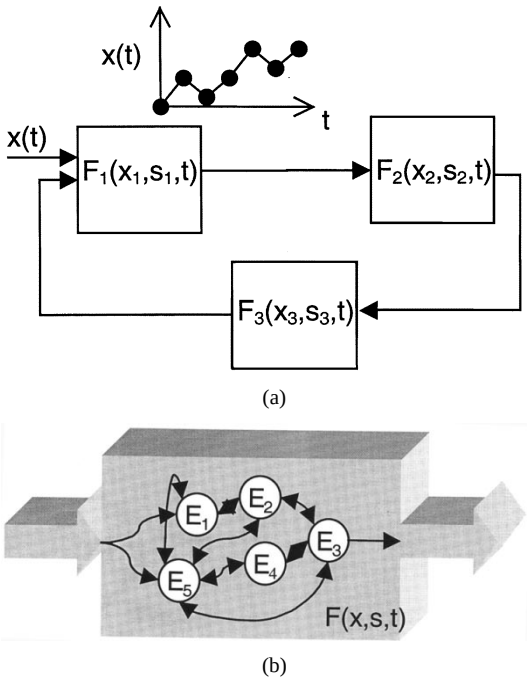


Fig. 1. (a) System-level discrete event simulation. (b) Behavioral model of one component.

B. Physical Device Versus Behavioral Component Models

We make a distinction between device-level and component-level modeling. Device-level models focus on explicitly modeling the processes within the physical structure of a device such as electromagnetic fields, fluxes, mechanical stresses, and thermal gradients. These are typically described by partial differential equations in both space and time. Conversely, in behavioral-level models these distributed effects are captured in terms of parameters, and the models focus on the relationships between these parameters and state variables (e.g., optical intensity, phase, current, voltage, displacement, or temperature) as a set of temporal linear or nonlinear differential equations.

Device-level simulation techniques offer the degree of accuracy required to model fast transients and fabrication geometry dependencies, as well as steady-state solutions in the device [4]. However, modeling these processes requires specialized techniques and large computational resources. Further, these simulations produce results that are generally not compatible with the simulators required for other domains. For instance, it is difficult to model the behavior of a laser in terms of carrier population densities while modeling the emitted light in terms of its electromagnetic fields.

In order to deal with the problem of physical device simulation in a multi-domain environment it is possible to use a simulator for each unique domain, coupled to the other domains through a higher level coordinating process that manages their behavior in terms of their common physical processes in energy and time. However, this technique has all the drawbacks previously

mentioned for the device-level simulation and the additional computational requirement to coordinate both simulators and make them converge to a common point of operation [5], [6]. For system-level simulation the computational costs of this method are prohibitive.

On the other hand, behavioral component models can be designed to capture the key behaviors of a device from physical processes. While this technique sacrifices some fidelity, the behavioral models can still provide enough accuracy for system-level simulation. Therefore, our approach is to incorporate the transient solution, along with other second order effects, of the device analysis within the behavioral component model. Different methodologies can be used to translate the device-level expressions, which characterize the device operation (e.g., for semiconductors, Poisson's equations, the carrier current, and the carrier continuity equation) into a set of temporal linear or nonlinear differential equations that are used in the behavioral model [4], [7].

The advantage of having this representation is that we can simulate electronic, mechanical, and optical models in a single mixed-domain simulation environment. It supports an abstract representation of the system consisting of a set of modules interchanging information (in terms of electronic, optical, or mechanical signals) as discussed above. And, it provides for a mechanism for varying the degree of accuracy of the simulation without changing the environment or the models. However, this approach brings the challenge of choosing which behavioral modeling techniques will be best for accurate and fast characterization of the varied components used in multi-domain microsystems.

C. Behavioral Component Modeling

When choosing a behavioral modeling methodology for MSMD systems, not only do we have to consider the sets of interactions between components of different technologies, we also have to consider the performance of the simulation environment, which depends on the simulation method and the type of signal characterization chosen. Much research has been conducted to offer a suitable methodology for the simulation of these systems. Here, we classify them into two different approaches: *functional modeling* and *equivalent circuit methods*.

Functional modeling is a flexible and general methodology that allows hierarchical support and mixed signal simulation. Hardware description languages with extensions to support analog signals such as VHDL-AMS or Verilog-A can be used to describe the system [8], [9]. In this approach, the degree of abstraction provided by the hardware description language simplifies the designer's task for the description of the system in terms of analytic expressions including differential equations [10], [11].

Even though the functional modeling approach appears to be a promising option for the modeling of MSMD systems, it is necessary to clarify the difficulties and limitations present in this technique. During the description of the system, an "expert" designer must specify the relations that define the interaction between the different signals in the system. The definition of these relationships is nontrivial for multi-domain components since it involves the characterization of ports, defined as transducers (energy conversion devices) and elements, defined as actuators (unidirectional energy flow devices) [12]. Additionally, since this technique takes advantage of the abstraction levels and language constructs offered by the mixed-signal simulation framework, it shares their drawbacks as well.

The second method for behavioral modeling is based on finding an equivalent circuit representation for the nonelectrical domain to be simulated. The electrical equivalent can be simulated using any of the well-known and established circuit simulators (e.g., SPICE, iSMILE [4], or Saber). This method has been used for the simulation of micro-mechanical devices, where a mapping of these devices to a SPICE netlist is proposed [13]–[15]. Yang [16] simulated optoelectronic interconnection links using iSMILE as the circuit simulator engine. The limitations of the equivalent circuit technique are the lack of support for hierarchical design and co-simulation. Additionally, because the simulation is coupled to an analog simulator, digital simulation is not supported.

For both the functional- and circuit-based approaches, the fundamental limitation for system-level simulation comes from the algorithm used for the analog simulation. MSMD microsystems, which consist of a very large number of elements at the system level, will produce a large computational load for typical mixed-signal simulators, based on conventional analog simulators solving large sets of coupled differential equations.

As an alternative to traditional circuit simulation, nonlinear network modeling techniques using PWL models have been developed [17], [18]. This technique has been applied with success in simulators such as NECTAR 2 [19], PLANET [20], and PLATO [21]. These simulators are much more stable when compared to traditional circuit simulators and provide flexibility for their use in hierarchical design.

Conventional PWL simulators use integration techniques to solve the transient response of the system because they use continuous analog behavior for input signals. This is an accurate but computationally demanding approach because it requires integration techniques to solve the set of linear differential equations.

In our approach, we extend the PWL technique to also represent the discrete event signals in the system. The input signals are linearized and, consequently, the transfer function for each of the circuit elements in the components can be obtained explicitly. This decreases the computational requirements because it avoids the integration process required in the conventional algorithms.

Additionally, a literal representation for the equivalent circuit representation of linear and nonlinear elements is used as the PWL formulation. This avoids the computational overhead of using a superset, or ensemble, of PWL models for the representation in the linear numerical analysis solver, which is the case for other PWL simulator implementations. In our case, the different configurations of the network are changed according to the change of regions of operations over individual nonlinear elements and not through the use of ideal switches that configure the superset model. Boundary conditions in individual nonlinear elements are used to determine the switching behavior between

configurations. In the next section, we present our implementation of this method.

III. Behavioral Modeling Methodology

Once we have chosen to use a PWL modeling technique, there are three basic approaches to the modeling of a component composed of linear and nonlinear elements in an equivalent circuit model. The first is to use an interconnection of available linear and nonlinear circuit elements, such as ideal diodes or current sources, which have been precharacterized as PWL devices in a library. The second approach is needed if we do not have appropriate nonlinear models in the library. Then, the behavioral modeler must explicitly model each nonlinear element in the device as a PWL function. That is, they must specify the number of linear regions of operation, their boundaries, and the functions that map device parameters (such as length) to changes in the behavior. This static model would be combined with linear models for intrinsic or extrinsic parasitics to provide a large signal model for the component. The third approach starts with a set of analytical equations that characterize the nonlinear element and then performs an automatic linearization of these relations to generate a similar static model.

Each of these three methods meets the needs of various types of design methodologies. The first is applicable where equivalent large signal models are known, while the other two methods provide a flexible methodology to model new devices. In the following sections, we present the automatic approach based on the relationships among the ports of each device. First, we review the mathematical justification for PWL approximations in multi-domain systems.

A. PWL Approximation

Whereas at the system level each component is a black box, each component can be described as a network of linear and nonlinear elements. For our modeling methodology, the component is first decomposed into a nodal representation of elements, as was shown in Fig. 1(b), where elements are interconnected through nodes. Every port is a pair of nodes in the component. This can be done for components in the electrical, mechanical, or optical domains, and for components which themselves span multiple domains.

Next, the behavior of each element is captured in terms of the analytic relationships among variables which define the state of its nodes. The two basic types of variables in nodal analysis are *across* and *through* variables. Across variables are measures of the values of field potential in the physics of the device (e.g., electrical potential, temperature, fluid pressure). Through variables are measures of flux intensity at nodes (e.g., electrical current, thermal flux, fluid velocity).

The nodal analysis principle can be traced back to the basic conservation laws of energy and bond graph theory [22]. In an enclosed volume with finite interfaces, an energy conservation relationship can be established using the energy flow through the interfaces and the internal energy density.

Consequently, for any element in the nodal representation, a function f can be found that relates the total flows (across variables) through its interfaces (nodes) as equal to zero.

If we define the state of the element $\hat{s}(t)$ as being a vector of all its across variables $\hat{x}_{cr}(t)$, through variables $\hat{x}_{\rm th}(t)$, and their associated (n) derivatives for all its m nodes at time t, then the nodal function f is defined as

$$f(\hat{s}) = 0 \tag{1}$$

where $\hat{s} = [(\hat{x}_{cr}, \hat{x}_{cr}, \hat{x}''_{cr}, \ , \ldots, \hat{x}^{(n)}_{cr}), \ (x_{th}, x'_{th}, \hat{x}''_{cr}, \ldots, \hat{x}^{(n)}_{cr})]^T$, the across variables are $\hat{x}_{cr} = [V_1(t), \ldots V_m(t)]$ and the through variables $\hat{x}_{\rm th} = [I_1(t), \ldots, I_m(t)]$.

The linearization of this function can be obtained through a Taylor expansion around a point $\hat{s}_c$ where the function is differentiable

$$f(\hat{s}) = f(\hat{s}_c) + f(\hat{s}_c)'(\hat{s} - \hat{s}_c) + \frac{f(\hat{s}_c)''}{2}(\hat{s} - \hat{s}_c)^2 + \frac{f(\hat{s}_c)^{(n)}}{n!}(\hat{s} - \hat{s}_c)^n + \cdots. \tag{2}$$

Using only the first-order term

$$f(\hat{s}) \cong f(\hat{s}_c) + f(\hat{s}_c)'(\hat{s} - \hat{s}_c) \cong 0. \tag{3}$$

Equation (3) is a set of ordinary differential equations (ODE) of order n, in a vector form, that represents the PWL equivalent of the device at time t. Additionally, this expression is in a nodal form that can be mapped directly to an MNA formulation [23]. The relevance of this formulation is that it involves multi-domain variables and nonlinear elements. The additional complexity of order n for the ODE can be resolved using an appropriate variable change that reduces the expression to first order, as shown in Section III-E.

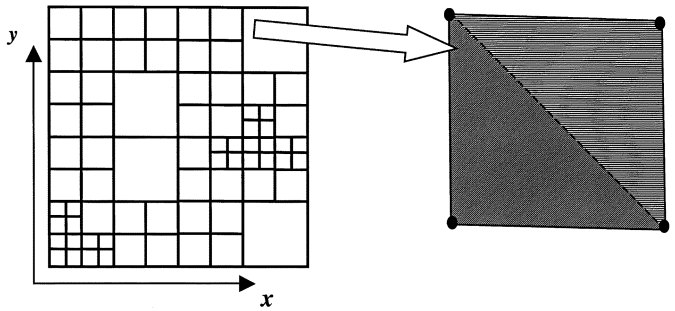
B. Generation of Multidimensional Approximations

For the simulation, we need a way to provide a linear approximation of $f_i(s)$ for each of the $i\,(1 < i < m)$ nodes that make up the ports of the element. In general, this will be a hyperplane in the dimensionality of the domain of s, plus one for the range of f. However, since we are using a linear approximation, rather than a single plane we decompose this space into regions, with a PWL approximation in k, the dimensionality of s. This gives us the ability to approximate the function to the degree of accuracy required for the range of operation of interest.

While there are many choices for the decomposition, the resulting PWL approximation of the function should have the following characteristics: it must be linear in the dimensionality of the function (e.g., planes in three-space for functions of two variables); it should approximate the function within specified absolute and relative tolerances; it should make the fewest partitions possible; and, it must be mathematically continuous in the underlying function value at the transition points between the different regions of operation.

This last point reduces the probability that the device model, in simulation, will oscillate between regions of operation due to poor convergence. It also makes simple, curve-fitting (e.g., based on minimizing rms error) techniques less applicable. Rather, we use a triangulation approach based on recursive decomposition of the function space.

We define a $k+1$ dimensional space for the domains of the k independent variables of s and the range of f. We recursively decompose the k dimensional projection, from the independent



(a) (b)

Fig. 2. (a) Partitioned two-dimensional (2-D) projection of three-dimensional (3-D) function $f(x, y)$. (b) One patch decomposed into two planar regions.

variables, of the space into k dimensional hypercubes. The decomposition need not be symmetric; we can refine one region of the space more than others. Each hypercube defines the region of operation for the element. The advantage of cubic decomposition is that the bounds for each region of operation can be tested efficiently in the simulator. Recursion stops when we reach our tolerance level in every region. An example of such a partition of a 2-D projection is shown in Fig. 2(a).

The next step is to triangulate the hypercubes, since we want a linear approximation of the function; however, the 2^k corner points of the hypercube over-constrain a linear function in $k+1$ dimensions. That is, four points over-constrain a plane in three dimensions, giving a saddle rather than a plane. In three dimensions, this is solved by breaking the four-point patches into two planes, defined by two triangles, as shown in Fig. 2(b).

For functions of more than two variables, the recursive hypercube decomposition remains straightforward, however, the triangulation of the resulting hypercubes is not simple [24]. While, the best decomposition of the 3-D cube is five tetrahedra and four-dimensional hypercubes decompose into 16 hyper-tetrahedra, in higher dimensions the decomposition grows to be quite large [25] and finding the optimal is a very difficult problem. In this work, we use a straightforward vertex index permutation approach.

In general, a k-simplex is the convex hull of $k+1$ points in k-dimensional space. A k-dimensional hypercube is triangulated if it is partitioned into finitely many k-simplices with disjoint interiors. In particular, we need a triangulation where the vertices of all the k-simplices are also vertices of the original cube and the intersection of any two k-simplices is a face of each of them. This is called "face-to-face vertex triangulation." Using a general vertex index permutation algorithm, we can perform the triangulation for higher dimensional functions. This gives us $k!$ nonoverlapping hyperlinear partitions of the hypercube where every vertex is from the original hypercube and the boundaries of the region are defined by the faces of the simplex, defined by linear equations.

As an example, in Fig. 3 we show the linearization of the simple n-channel MOS (nMOS) transistor equation [26]

$$I_{ds} = \begin{cases} 0 & V_{\rm th} > V_{gs} \\ \beta\left[(V_{gs}-V_{\rm th})\,V_{ds} - \dfrac{V_{ds}^2}{2}\right] & V_{ds} > (V_{gs}-V_{\rm th}) \\ \dfrac{\beta}{2}\,(V_{gs}-V_{\rm th})^2 & V_{ds} < (V_{gs}-V_{\rm th}). \end{cases} \quad (4)$$

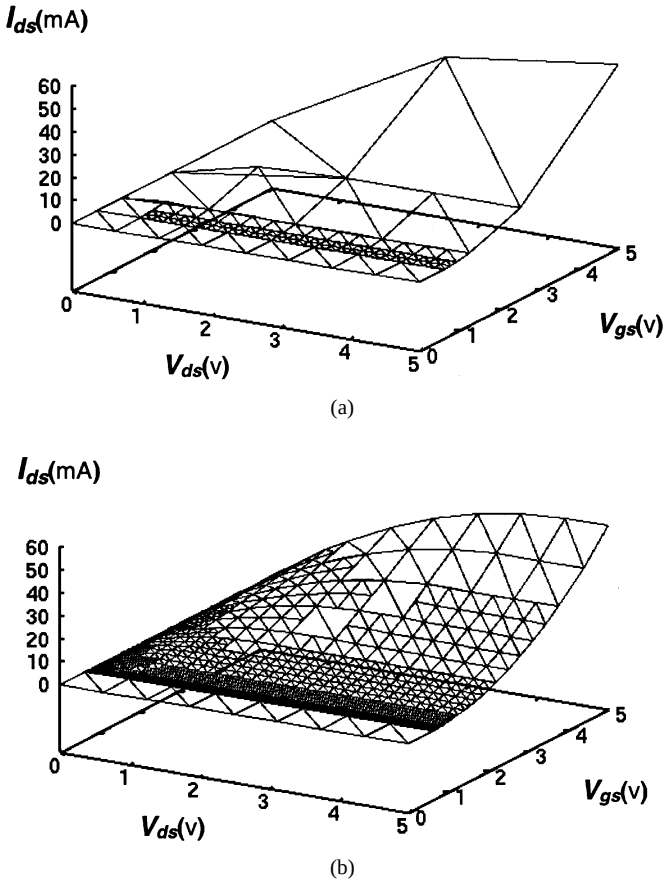


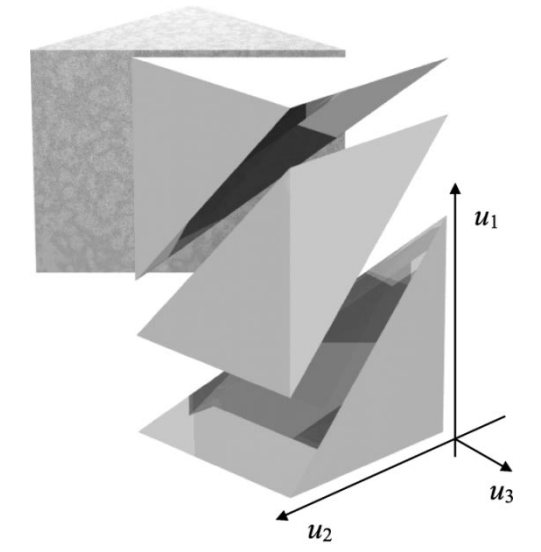
Fig. 3. Linearization of NMOS transistor, I_{ds} vs. V_{gs} and V_{ds} for (a) 25% and (b) 1% relative accuracy.

For $V_{gs} > 0$ V and $V_{ds} < 5$ V. Fig. 3(a) shows the linearization for 25% relative accuracy while Fig. 3(b) shows the same device modeled to 1% relative accuracy. We note that at 25% accuracy the figure shows a discontinuity caused by two patches having been decomposed to different levels of granularity. This problem can be addressed by edge coherence techniques [27].

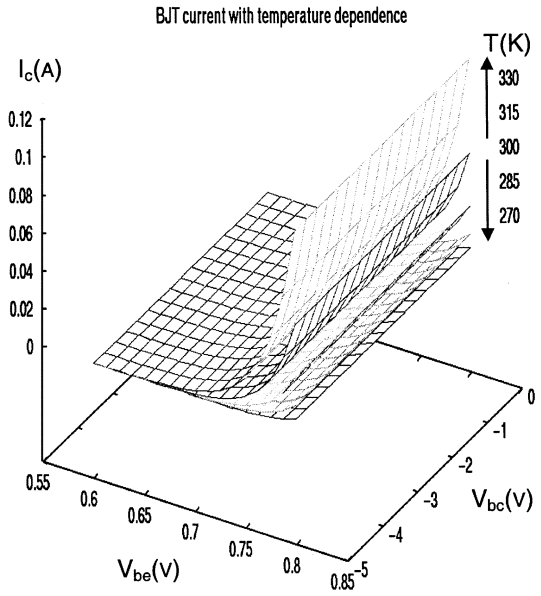
Fig. 4 illustrates the case for three dimensions. After recursive decomposition of the space into cubes, each cube represents a domain for a function of three variables and each tetrahedron defines a PWL approximation of the function. In Fig. 4(a), we show the tetrahedral triangulation of a single 3-D cube. In Fig. 4(b), we use this triangulation to graph a function for collector current in a bipolar transistor, I_c, as a function of $0.55\,(\text{V}) < V_{be} < 0.88\,(\text{V})$, $-5.0\,(\text{V}) < V_{bc} < 0\,(\text{V})$, and $270\ \text{K} < T_{op} < 330\ \text{K}$, using the Ebers–Moll formula [26]

$$I_c = I_s\left(e^{V_{be}/V_t} - 1\right) - \left(\frac{I_s}{\alpha_r}\right)\left(e^{V_{bc}/V_t} - 1\right),$$
$$\text{with } V_t = V_{t\ \rm nom}\left(\frac{T_{op}}{T_{\rm nom}}\right)$$
$$I_s = I_{s\ \rm nom}\left(\frac{T_{op}}{T_{\rm nom}}\right)^{XTI} \exp\left(\frac{g}{V_t}\left(\frac{T_{op}-T_{\rm nom}}{T_{\rm nom}}\right)\right). \quad (5)$$

The sets of linear regions of operations, together with the definitions of the boundaries for each region, for each element is captured in a "template" data structure as shown in Fig. 5. We



(a)



(b)

Fig. 4. (a) Decomposition of cube into six tetrahedrons representing a PWL function in three-space. (b) Equal-depth 3-D linearization of NPN BJT, I_c vs. V_{be} and V_{bc} for temperatures 270 K–330 K.

note that there can be multiple templates for each device, depending on the choice of underlying analytic expressions, the accuracy chosen during the linearization process and the physical parameter dependencies.

The modeling methodology described above provides us with a technique for modeling new devices based on analytic expressions for their input/output relationships. However, as mentioned above, it is often convenient to model systems where some components are purely electrical and have already been captured as a SPICE netlist in terms of elements in a library. Therefore, we have also provided this interface to the behavioral modeler.

C. Library Interface

We have implemented a template library-based interface to the simulator in order to provide a method for parsing component descriptions into the behavioral simulator. In general, the library can contain elements from various domains including mechanical and optical. Currently, the library enables the user to import existing SPICE netlists for electrical components while performing simulations with mechanical and optical components. The flow for this is shown in Fig. 6. A SPICE netlist is parsed to extract the interconnection structure of elements. As elements are read in, the template library is searched for structural descriptions of the elements and their physical parameter dependencies. For linear components this is all that is needed to build the MNA matrix representation of the circuit. For nonlinear components, we also need to extract the definitions of the regions of operation to be used during simulation. These definitions can come from a predefined library or from the process described above. While the structural description and physical parameters do not change during simulation, for nonlinear elements the regions of operation do change. Managing these changes is done by the solver interface as part of the behavioral simulator.

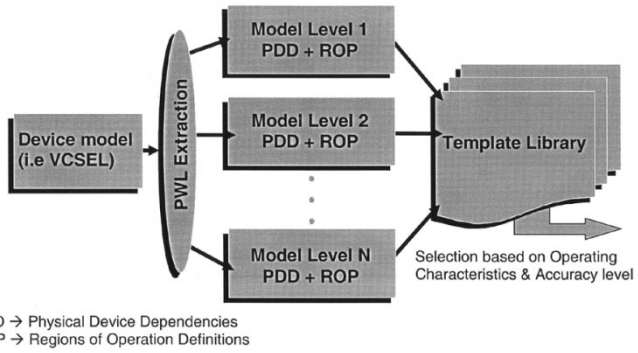


Fig. 5. Template library creation.

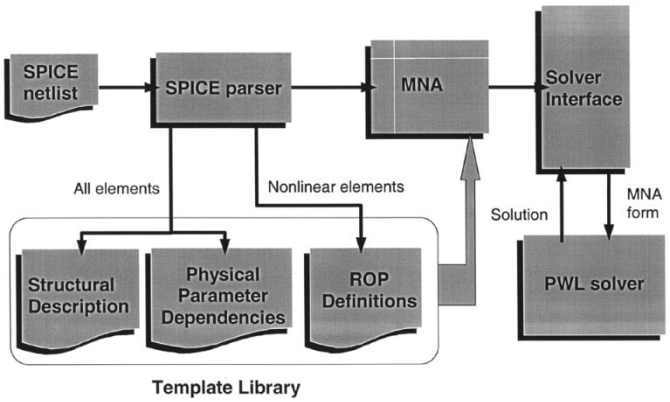
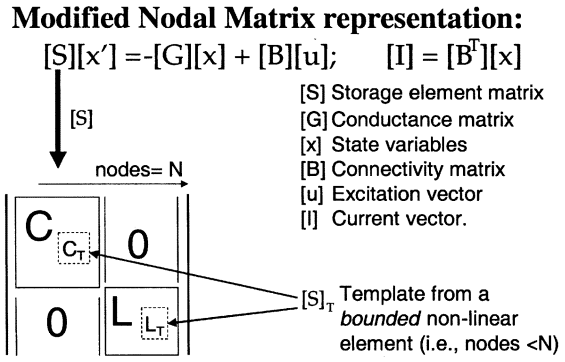


Fig. 6. Library interface for PWL behavioral solver.

D. Simulation

As mentioned above, MNA [23], [28] is used to create a matrix representation for each component. As shown in Fig. 7(a) for electrical components, [S] is the storage element matrix, [G] is the conductance matrix, [x] is the vector of state variables, [B] is a connectivity matrix, [u] is the excitation vector, and [I] is the current vector [23], [28], [29].

The linear sub-block elements are directly placed into this representation. The structures of the nonlinear elements are also incorporated directly but in the form of placeholders for their



(a)

General motion equation

$$F = [K][U] + [B][\dot{U}] + [M][\ddot{U}]$$

Standard ODE Transformation

$$\begin{bmatrix} 0 & M \\ M & B \end{bmatrix}\begin{bmatrix} \ddot{U} \\ \dot{U} \end{bmatrix} + \begin{bmatrix} -M & 0 \\ 0 & K \end{bmatrix}\begin{bmatrix} \dot{U} \\ U \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} F$$

Templates for every basic element (e.g. beam)

$$X = \begin{bmatrix} \dot{U} \\ U \end{bmatrix}; \quad [Mb]\dot{X} + [Mk]X = [E]F$$

(b)

Fig. 7. MNA description for (a) electrical and (b) mechanical components.

templates. The templates give us the ability to change these models for the nonlinear devices depending on the changes in conditions in the circuit, and thus the regions of operation.

Once the integrated MNA is formed, a linear analysis in the frequency domain can be performed to obtain the solution of the system. The MNA representation is initialized by the application of values for the state variables of the component. The regions of operation definitions for the nonlinear elements are compared against these state variables, and the MNA representation is updated accordingly. This MNA representation is then passed to the solver, which returns the new set of state variables.

During each time step in the simulation, the state variables in the module will change and might cause the nonlinear elements to change regions of operation. Therefore, we recompute the solution caused by changes between piecewise models. In general, depending on the number of regions of operation used in the PWL model, there are a large number of time steps during which the system representation is unchanged, justifying the computational savings of this technique.

Understanding that the degree of accuracy of PWL models depends strongly on the step size chosen for the time base, an adaptive control method is used [29]. For any nonlinear element, a coarse discretization may cause the element to move out of the valid range space for its model. If this occurs, the state variables are restored to the last successful time point, and the time step is reduced. The current iteration is then rerun with the reduced time step. If this time–step reduction results in the element moving to a valid region of operation, then the time step is accepted; otherwise, the same process is performed recursively. The algorithm also discards nonsignificant time samples, which do not appreciatively affect the output. The inclusion of the samples during fast transitions or suppression of samples during "steady-state" periods optimizes the number of events used in the simulation.

Due to the continuous nature of the analytic expressions, an element is likely to remain in its current region of operation or move to an adjacent region. The data structure for storing the regions and the search for the new region of operation uses this information to improve performance. However, sometimes, an element moves through its space "too fast." If it skips over neighbor regions and moves instead to (still valid) regions that are further away, the result may lead to inaccurate results or extra transitions for other elements in the device. In this case, a better result may be obtained if the time step is preemptively reduced, even though it is not critical at the current time. Both of these heuristics are used to trade off between computation time and accuracy.

For electrical components, the inputs and outputs of the component are identified nodes of the network. The output nodes have characteristic output impedances, providing impedance matching between electrical components. This impedance, together with a PWL voltage waveform is passed to other components by the discrete event simulation engine at the system level.

While we have used electronic components for the preceding discussion, these same techniques apply to other domains. For mechanical components, we can derive a similar template-based structure for composition into an MNA formulation as explained next. Then we present an overview of our optical signal representation and propagation methods followed by a system simulation example.

E. Mechanical Behavioral Modeling

The same general solver using PWL techniques can be used for mechanical models as well as electrical components. The model for a mechanical device can be summarized as a set of differential equations that define its dynamics as a reaction to external forces. This model can be converted to the same form as in the electrical case to be given to the PWL solver for evaluation.

With damping forces proportional to the velocity, the equation of motion for a mechanical structure with viscous damping effects is $F = KU + BV + MA$ [30] where, K is the stiffness matrix, U is the displacement vector, B is the damping matrix, V is the velocity vector, M is the mass matrix, A is the acceleration vector, and F is the vector of external forces affecting the structure. Obviously, knowing that the velocity is the first derivative and the acceleration is the second derivative of the displacement, the above equation can be recast to $F = KU + BU' + MU''$.

Similar to the electrical modeling case, this equation represents a set of linear ODEs if the characteristic matrices K, B, and M are static and independent of the dynamics in the body. If the matrixes are not static and independent (e.g., the case of aerodynamic load effects), they represent a set of nonlinear ODEs.

Using a modification of Duncan's reduction technique for vibration analysis in damped structural systems [31], we reduce the above general mechanical motion equation to a standard first order form, similar to electrical model which gives a complete characterization of a mechanical system, as shown in Fig. 7(b).

Each mechanical element (beam, plate, etc.) is characterized by a template consisting of the set of matrices Mb and Mk, composed of matrices B, M, and K.

If the dimensional displacements are constrained to be small and the shear deformations are ignored, the derivation of Mb and Mk is simplified and independent of the state variables in the system. Typically, this element is only a part of a bigger device made from individual components that are characterized using similar expressions. The generalization of the previous case to an assembly of elements or mechanical structures is fairly straightforward [29], [30].

We use dynamic control of the sampling rate in the mechanical domain based in the Nyquist criteria of the highest significant modal frequency for the structure. The allowed sampling rate is lower than half of the period of the highest modal frequency. This allows us to optimize the samples used in this domain while still completely characterizing its dynamic behavior. There can be several orders of magnitude reduction in the sampling rate compared to the electrical domain because of the difference in dynamics.

The use of a PWL general solver for mechanical simulation decreases the computational task and allows for a tradeoff between accuracy and speed. The additional advantage of using the same technique to characterize electrical and mechanical models allows us to easily merge both technologies in complex devices that interact in mixed domains.

For the optical domain, however, we need to explicitly consider the propagation medium as well as the optical components themselves. This is because in free space, optical signals do not simply propagate point to point.

IV. Optical Propagation

When optical wavefronts interact with the small feature sizes of microsystems, many of the common optical propagation modeling techniques become invalid, and full vector or scalar solutions to Maxwell's equations are required for accurate simulation [32]. However, these accurate solutions are computationally intensive, making interactive design between system designer and CAD tool almost impossible. As more optical components are introduced into microsystems and the systems become more complex, the demand for computationally efficient simulation tools increases. Therefore, the problem of optical modeling in MSMD microsystems is twofold: first, a rigorous model is needed to model optical propagation, and, second, the model must be computationally efficient.

To reduce the computational resources of modeling the optical wavefront completely by the vector solution of Maxwell's equations, a scalar representation is commonly used. Scalar optics are defined by summarizing the electric field vector, $\vec{E}$, and the magnetic field vector, $\vec{H}$, by a single complex scalar, U. This replacement is valid if the propagation medium is dielectric, isotropic, homogenous, nondispersive, and nonmagnetic. Propagation through free space meets these requirements.

This complex scalar must satisfy the Helmholtz wave equation, $(\nabla^2 + k^2)U = 0$, where, the wave number, $k = 2\pi/\lambda$. With use of Green's theorem, the Rayleigh–Sommerfeld formulation is derived from the wave equation for the propagation of

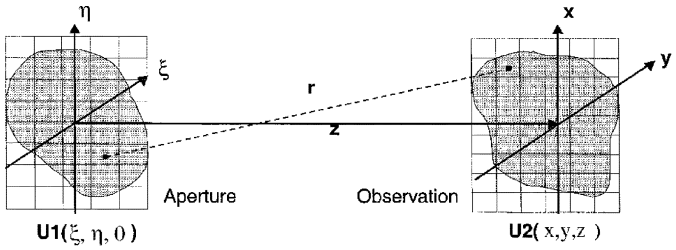


Fig. 8. Aperture and observation coordinate system in the Rayleigh–Sommerfeld approximation.

light in free space from the aperture plane $(\xi, \eta, 0)$ to a parallel observation plane (x, y, z), as seen in Fig. 8 [33]

$$U(x, y, z) = \frac{z}{j\lambda} \iint_{\Sigma} U(\xi, \eta, 0) \frac{\exp(jkr)}{r^2} \, \partial\xi \, \partial\eta \quad (6)$$

where, $r = \sqrt{z^2 + (x-\xi)^2 + (y-\eta)^2}$, Σ is the area of the aperture, and z is the distance that the light is propagated from an aperture plane $(z = 0)$ to the observation plane. The formulation is valid as long as both the propagation distance and the aperture size are greater than the wavelength of light. These restrictions are based on the boundary conditions of the Rayleigh–Sommerfeld formulation, and the fact that the electric and magnetic fields cannot be treated independently at the boundaries of the aperture [33]. To compute the complex wavefront at the observation plane, each plane is discretized into an $N \times N$ mesh. Using a direct integration technique, the computational order of the Rayleigh–Sommerfeld formulation is $O(N^4)$.

The far (Fraunhofer) and near (Fresnel) field approximations of the scalar formulation reduce the computational demand, using a fast Fourier transform (FFT) for optical propagation. However, we have shown that these techniques are not valid for typical microsystem dimensions [32]. In the interest of reducing the computational load of using a full scalar technique, we have recast the Rayleigh–Sommerfeld formulation using an angular spectrum technique.

A. Angular Spectrum Technique

As an alternative to direct integration over the surface of the wavefront, the Rayleigh–Sommerfeld formulation can also be solved using a technique that is similar to solving linear, space-invariant systems. Reexamining the Rayleigh–Sommerfeld formulation, it can be seen that the equation is in the form of a convolution between the complex wavefront and the propagation through free space [34]. The FFT of the complex optical wavefront results in a set of plane waves traveling in different directions away from the surface [33]. Each plane wave is identified by the components of the angular spectrum. At the observation plane, the plane waves are summed together by performing an inverse FFT, resulting in the propagated complex optical wavefront at the observation plane. Brief details of the technique follow.

To solve the Rayleigh-Sommerfeld formulation with the angular spectrum technique, we first examine the complex wavefront at the aperture plane. The wave function $U(x, y, 0)$ has a

2-D FFT, $A(v_x, v_y, 0)$, in terms of angular frequencies, v_x and v_y

$$A(v_x, v_y, 0) = \iint U(x, y, 0) \exp[-j2\pi(v_x x + v_y y)]\, \partial x\, \partial y \quad (7)$$

where, $v_x = \sin\theta_x/\lambda$ and $v_y = \sin\theta_y/\lambda$.

From the equation, the plane waves are defined by $\exp[-j2\pi(v_x x + v_y y)]$ and the spatial frequencies define the directional cosines, $\sin(\theta_x)$ and $\sin(\theta_y)$, of the plane waves propagating from the origin of the aperture plane's coordinate system.

The free-space transfer function in the frequency domain has been computed by satisfying the Helmhotz equation with the propagated complex wave function, $U(x, y, z)$

$$A(v_x, v_y, z) = A(v_x, v_y, 0) \exp\left\{jz2\pi\sqrt{\frac{1}{\lambda^2} - v_x^2 - v_y^2}\right\}. \quad (8)$$

This describes the phase difference that each of the plane waves, differentiated by the angular, or spatial frequencies, experiences due to the propagation between the parallel planes. Therefore, the wave function after propagation can be transformed back into the spatial domain with the following inverse FFT

$$U(x, y, z) = \iint A(v_x, v_y, 0) \exp\left\{jz2\pi\sqrt{\frac{1}{\lambda^2} - v_x^2 - v_y^2}\right\} \times \exp[j2\pi(v_x x + v_y y)]\, \partial v_x\, \partial v_y. \quad (9)$$

The advantage of using the angular spectrum to model light propagation is that the method is based on the FFT. The computational order of the FFT for a 2-D input is $O(N^2 \log_2 N)$.

In continuous theory, the angular spectrum method is an exact solution of the Rayleigh–Sommerfeld formulation. However, when using a discrete FFT, the accuracy of the angular spectrum method depends on the resolution of the aperture and observation plane mesh.

We have determined in 2-D space that with a mesh spacing of $\Delta x, \Delta y < \lambda/2$, the angular spectrum decomposition will ensure plane waves propagating from aperture to observation plane in a complete half circle, that is, between -90 and $+90$ degrees [35]. For many simulation systems without large degrees of tilt and hard diffractive apertures, the resolution can be coarser. In systems with high tilts, the resolution is most sensitive.

Now that we have introduced our modeling techniques for electrical, mechanical components, and optical propagation, we present an example microsystem, which spans these three domains.

V. Example System: A Grating Light Valve (GLV) Projector

To provide motivation for our MSMD CAD tool, we examine one of the more promising optical MEM components, the GLV [36]. This device has many display applications, including digital projection, HDTV, and vehicle displays. The GLV is simply a MEM phase-grating made from parallel rows of reflective ribbons. When all the ribbons are in the same plane, incident light that strikes normal to the surface reflects 180° off the GLV creating the so called zeroth mode of a diffraction pattern. However, if alternating ribbons are moved down a quarter of a wavelength $(\lambda/4)$ of the incident optical light, a "square-well" diffraction pattern is created, and the light is reflected at an angle from that of the incident light, into the odd ($\pm$1st, $\pm$3rd) diffractive modes. The angle of reflection depends on the width of the ribbons and the wavelength of the incident light. Fig. 9 shows the ribbons, from both a top and side view, and also the reflection patterns for both positions of the ribbons.

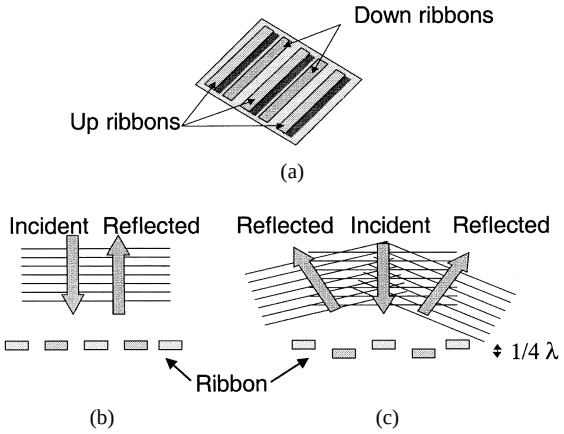


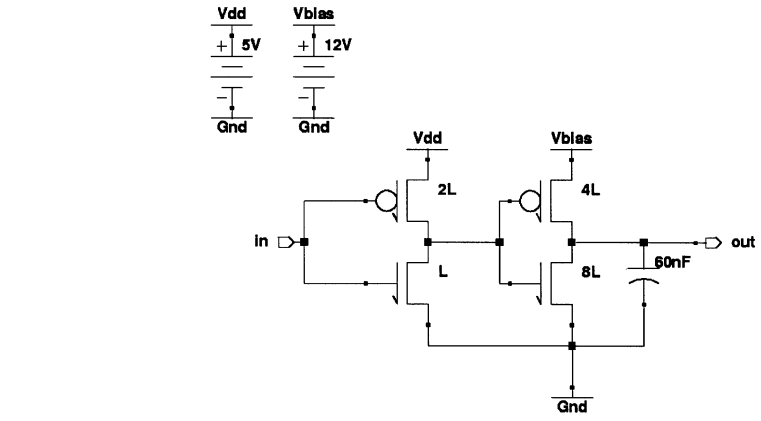
Fig. 9. GLV device. Top view (a) and side views: ribbons all up (b) and alternating ribbons pulled down (c).

The GLV component is fabricated using standard silicon VLSI technology, with ribbon dimensions approximately 3–5 μm wide and 20–100 μm long. Each ribbon moves through electrostatic attraction between the ribbon and an electrode fabricated underneath the ribbon. This electrostatic attraction moves the ribbons only a few hundred nanometers, resulting in an approximate switching time of 20 ns. Since the simulation of a GLV system relies on the optical wavefront, the mechanical displacement of the ribbons, and the electrostatic attraction between the ribbons and the substrate, a CAD tool that can model the multidomains and interactions between these domains is required.

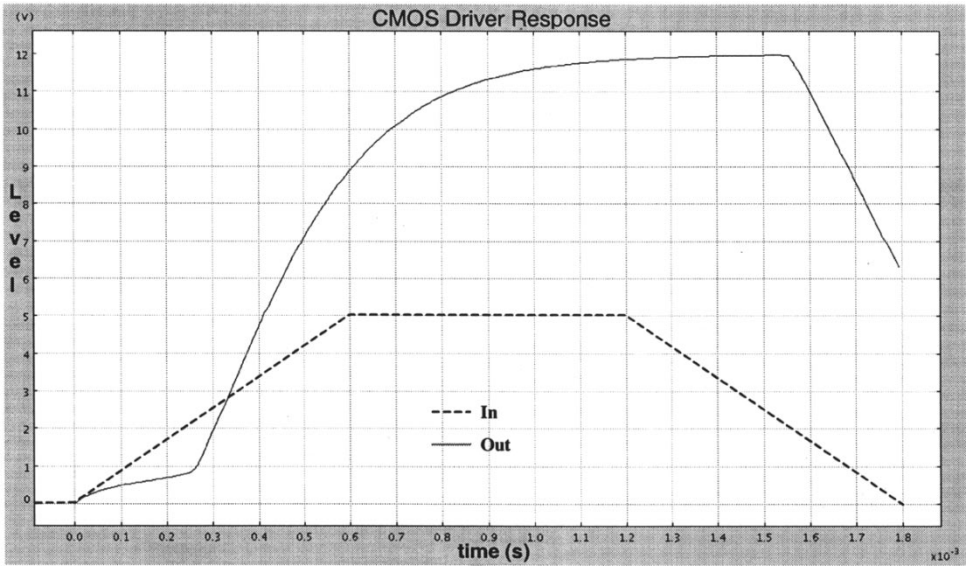
A. GLV Simulation

In this section, we present simulation and analysis of the GLV system. For the simulations of the GLV, we examine one optical pixel. A projected pixel is diffracted from a GLV composed of four ribbons, two stationary and two that are movable [36]. In our simulations, each ribbon has a length of 60 μm, a width of 5 μm, and a thickness of 1.5 μm, for a total GLV pixel size of 60 $\times$ 20 μm. The ribbons are made of silicon nitrite (density 3290 Kg/m^3, Young's modulus 290 $\times$ 10^9 N/m^2), and coated with aluminum for smoothness and reflectivity. In these simulations, we assume there is no gap between the ribbons, however, in reality, a gap is present and is a function of the feature size of the fabrication.

The model of the GLV is twofold: an electromechanical model simulating the movement of the ribbons toward the substrate, and the optical model simulating the reflection of the



(a)



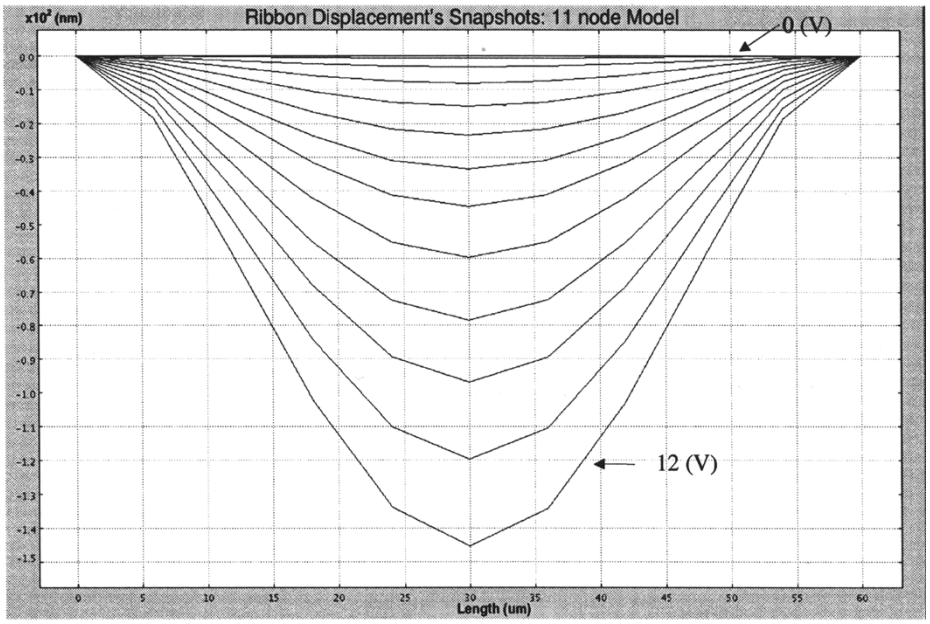
(b)

Fig. 10. (a) Two-stage CMOS driver for GLV and (b) driver input–output response.

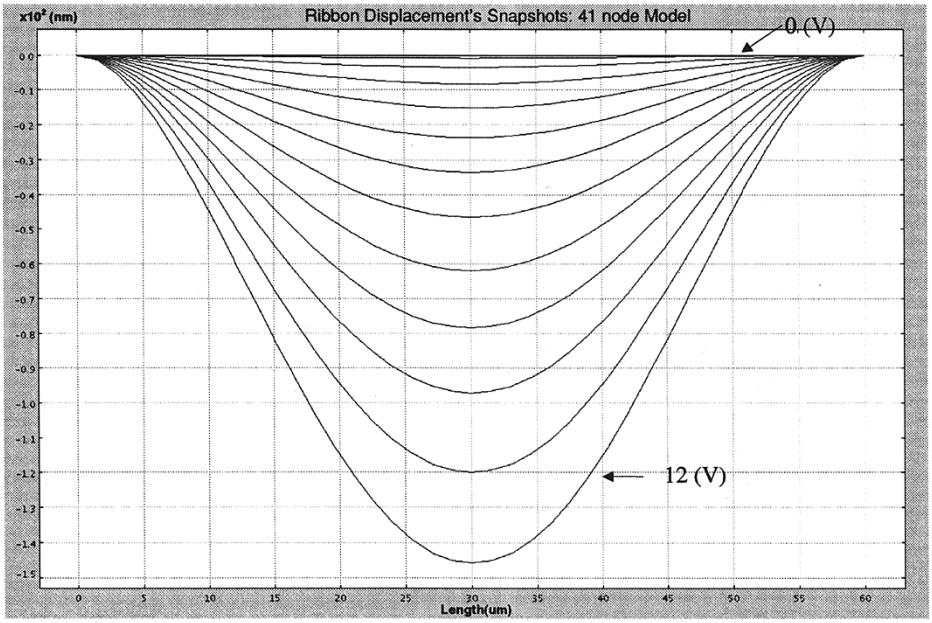
optical wavefront off of the ribbons. The ribbon is modeled as a thin cantilever beam anchored on each end. The beam is modeled in PWL segments, and is electrostatically attracted to the silicon substrate, which is covered with 500 nm of silicon dioxide. The voltage is applied between the ribbon and substrate electrode by a two-stage CMOS amplifier seen in Fig. 10(a). This electrical driver is modeled as described previously, and its response to a 0–5-V input ramp is also shown in Fig. 10(b). The air gap between the ribbons and the surface is 0.65 μm. This electrostatic model is connected to the optical GLV model through a "wire" containing the displacement of each node that comprise the model of the ribbon. A linear interpolation between the nodes is required for the optical mesh points that do not fall on the ribbon's nodes. The effect of the ribbon movement is optically modeled as a phase grating, where the light that strikes the down ribbons propagates further than the light that strikes the up ribbons. In our model, light reflecting from the down ribbons is multiplied by a phase term. The phase term is similar to a propagation term through a medium: $U_{\text{down\_ribbon}} = U\ \exp(j2kd)$, where, d is the distance that the ribbon is moved toward the substrate and k is the wave number, $k = 2\pi/\lambda$.

Since the ribbon ends are anchored, the alternating ribbons are not flat as they are electrostatically attracted to the substrate. As expected the beams are curved. In the simulations, the ribbon is composed of an equal sized number N of segments or basic beams, totaling $N + 1$ nodes. The layered shape of the ribbon with forces and movement limited to one plane justify the use of the basic beam element for the modeling of the mechanical structure. The analysis is reduced to a 2-D problem in the plane of the displacement. The accuracy of the mechanical simulation can be increased if a larger number of these basic elements are used at the cost of an increase in computation time. The resolution of higher fundamental nodal frequencies is proportional to the number of these segments. Simulation output data show the shape of the curved beams as the voltage between the ribbons and the substrate electrode is ramped between 0 and 12 V. For these simulations, we examine cases with 5, 11, 21, and 41 nodes. The mechanical deformation of the ribbon for the 11 and 41 node case is displayed in Fig. 11(a) and (b). Note that the y axis is in nanometers and the x axis is in microns.

We first perform simulations, in which ideal alternating flat, nonanchored ribbons move toward the substrate. We assume an incident plane wave of green light ($\lambda = 520$ nm) striking the



(a)



(b)

Fig. 11. Ribbon displacement at 1 V increments, (a) 11 node model and (b) 41 node model.

grating, with the square-well diffraction period defined by the ribbon width. We simulate the GLV in both cases, that is, when all the ribbons are on the same plane and when the alternating ribbons are moved downward a distance of 130 nm, or $\lambda/4$. In this example, the light is reflected off of the grating and propagated 1000 μm to an observation plane. An optical window of 400 $\times$ 400 μm is used, with an optical meshing equal to 256 $\times$ 256. Intensity contours of the optical waveform at the observation plane are presented in Fig. 12(a) for the case when the ribbons are all aligned, and when alternating ribbons are pulled down a distance equal to a quarter of the wavelength of the incident light, Fig. 12(b). Notice that the output optical waveform's height and widths are not equal. This is due to the rectangular shape of the GLV pixel, 60 $\times$ 20 μm. Also notice that the optical waveform appears to be in two lobes. This is a near-field optical effect of light propagating through a rectangular aperture and demonstrates that in this system, light propagating 1000 μm is not in the far field. This near-field effect highlights that the common scalar approximations, such as the Fraunhofer far-field approximation, would provide inaccurate simulation results, and the full Rayleigh–Sommerfeld formulation is required for accurate results.

The example is now resimulated with realistic curved ribbons. When curved ribbons are attracted down toward the substrate, the diffractive optical output is no longer ideal, as can be seen in the intensity contour of Fig. 12(c). Since the beam is curved from the anchors, a square-well diffraction pattern is no longer achieved, and the optical intensity contour appears to be a mix

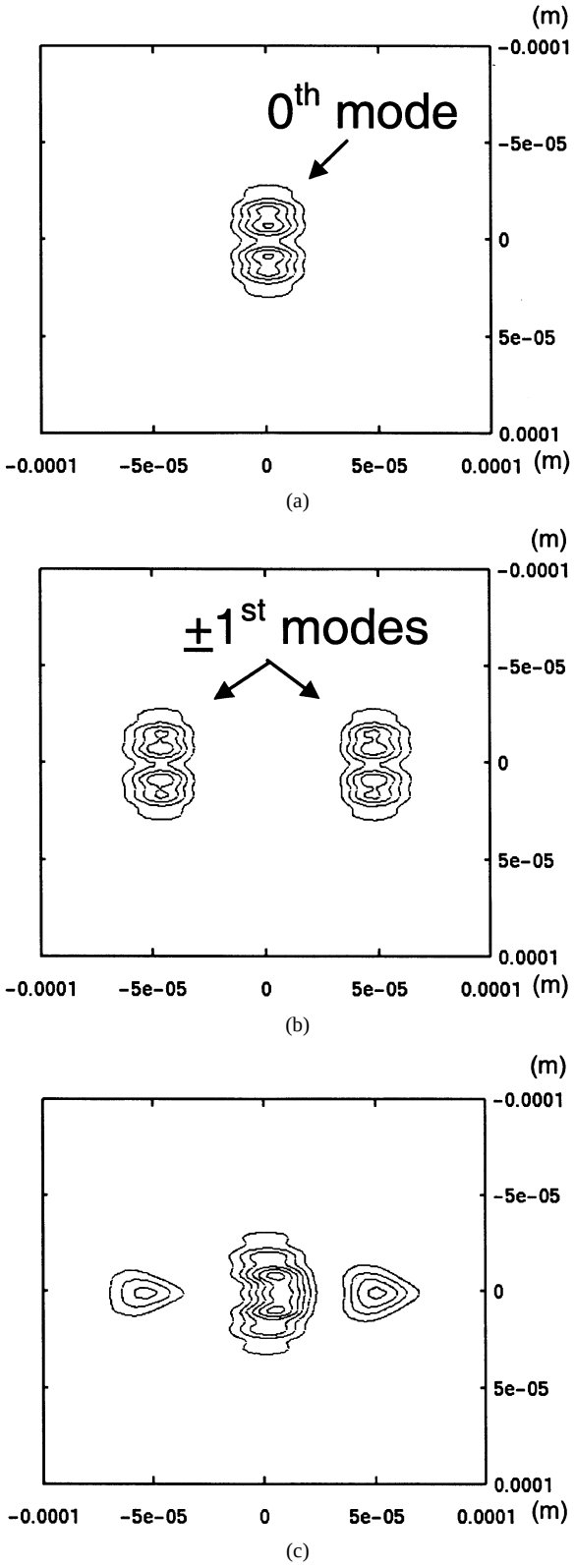


Fig. 12. GLV operation: (a) ribbons all up, (b) ideal ribbon displacement, and (c) curved ribbon displacement.

of the ideal cases seen in Fig. 12(a) and (b). The light reflecting from the middle of the ribbon, which is pulled down approximately $\lambda/4$ (130 nm), creates the $\pm$1st diffractive modes. These modes are now more circular, since effectively a 20×20 μm square-well is created in the center of the GLV device. The remainder of the light reflecting off the ribbons reflects straight off the GLV and creates the light found in the 0th mode.

In the next simulation, we performed a transient sweep of the applied voltage between the ribbon and the substrate electrode, from 0 to 12 V, with a complete switch occurring in 600 μs. The rest of the system setup is exactly the same as before. However, this time, we simulate the encircled power captured in the $+$1st diffraction mode for different ribbon depths. To simulate this, a circular detector (radius $= 10$ μm) is placed on the $+$1st mode.

Fig. 13 shows two graphs. The first graph shows the displacement of the center ribbon node and the input voltage with respect to time. From this result, we present the second graph in which we show how the ribbon movement affects the (normalized) encircled energy captured on the first mode detector. We can see, as the ribbons are attracted to the substrate, more optical power is diffracted into the nonzero modes. As the ribbons reach the $\lambda/4$ point (130 nm), the diffractive power peaks in the $\pm$1st mode. Beneath the two graphs in the figure are intensity contours of selected wavefronts during the transient simulation, along with markings of the system origin and circular detector position. From these wavefronts, interesting diffractive effects can be observed. As expected, when there is little voltage applied, all the light is in the 0th mode. As the ribbons move downward about $\lambda/8$ (65 nm), the energy in the $\pm$1st modes is clearly defined. As the gratings move closer to the $\lambda/4$ point, more power is shifted from the 0th mode into the $\pm$1st modes. As the ribbons start to return to their original position, the optical power shifts back into the 0th mode.

B. System-Level Simulation Performance

Using the same simulation environment we conducted the following tests to illustrate the speed/fidelity tradeoffs that can be done with a system–level simulation tool.

For reference, Fig. 14(a) shows the diffraction pattern at maximum ribbon displacement for a ribbon modeled with 5 segments and the scalar wavefront modeled with a 128×128 mesh. Fig. 14(b) shows the same system modeled with a 41-segment ribbon and a 512×512 mesh. What can be seen is the improvement in the degree of resolution of the wavefront, in particular the appearance of low-power third-order modes.

Additionally, Fig. 15 shows the dynamic response of the ribbon driven at a high switching frequency. The high stiffness of the structure gives it a fast response time as observed. However, under this stimulus, resonant effects are observed in the displacement of the nodes. The visible pattern of damped oscillations shows that the stiffness affects maximum operating speed of this device.

The accuracy of the mechanical simulation was also compared to modal analysis of the ribbon using ANSYS. The 11-node model matches the nine first-modal frequencies with a maximum difference of 2.24% at the highest frequency. For the 21-node model the tenth modal frequency differs by less than 0.59%. The 41-node model reduces this difference to 0.15%. As expected, to accurately capture higher modal frequencies, a larger discretization is required. Similar performance in the mechanical simulation of MEMs using this technique and its verification against NODAS [13] and ANSYS has previously been reported [29], [23].

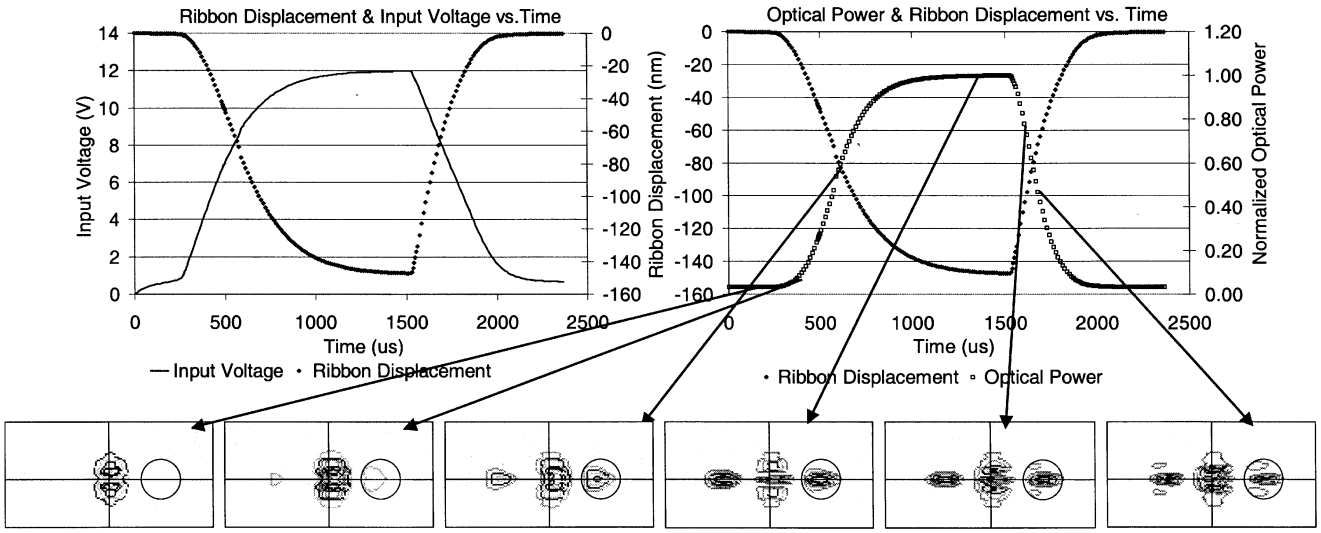


Fig. 13. GLV simulation graphs and intensity contours.

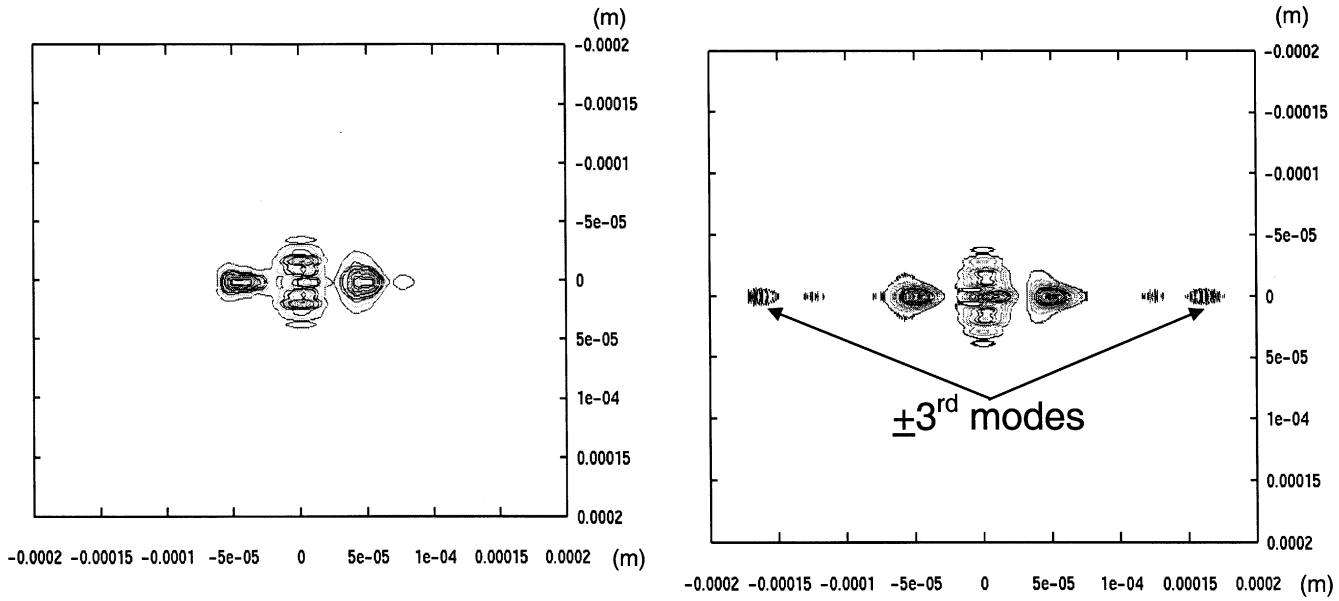


Fig. 14. Diffraction pattern at maximum ribbon displacement. (a) Ribbon model with 5 segments and 128 × 128 mesh for optical wavefront. (b) Ribbon model with 41 segments and 512 × 512 mesh for optical wavefront.

Table I shows system simulation time[1] as a function of both the scalar mesh resolution and the number of segments in the ribbon. The mechanical subsystem time includes the initialization of the MNA as well as the solution times for the entire movement for the 2.4 ms stimulus. The optical subsystem time includes both the scalar propagation time and the detector power integration time. The optical propagation time averaged 30 ms for the 128 × 128 case and 490 ms for the 512 × 512 case, while the integration time went from 2 to 41 s, respectively. We note that for typical systems, optical detection need only be done at the receivers after several stages of optical propagation. The system time included the electrical simulation of the CMOS driver, as well as initialization overhead. In previous work, we reported that the PWL electrical simulator was able to simulate simple CMOS circuits with a relative accuracy of 5% when compared to SPICE and with speedup factors of up to 100 [29].

What is interesting to note here, is the range of simulation time, 3 seconds for the 5-element, 128 × 128 case to 168 seconds for the 41-element, 512 × 512 case and the commensurate increase in fidelity of the resulting optical waveforms shown in Fig. 14(a) and (b). Similarly, Fig. 15 shows a high-fidelity description of the ribbon. What this illustrates is that we can use the same behavioral descriptions, in the same system-level simulation environment, to perform both interactive "what if" design exploration as well as more detailed investigations of higher order effects by simply changing the simulation parameters (e.g., optical mesh size, number of mechanical nodes, number of regions of operation for nonlinear elements, and

[1]For the simulations, a dual Pentium 1.7 GHz/Xeon processor with 4 GB RAM/PC800, running under Red Hat linux 7.1. was used.

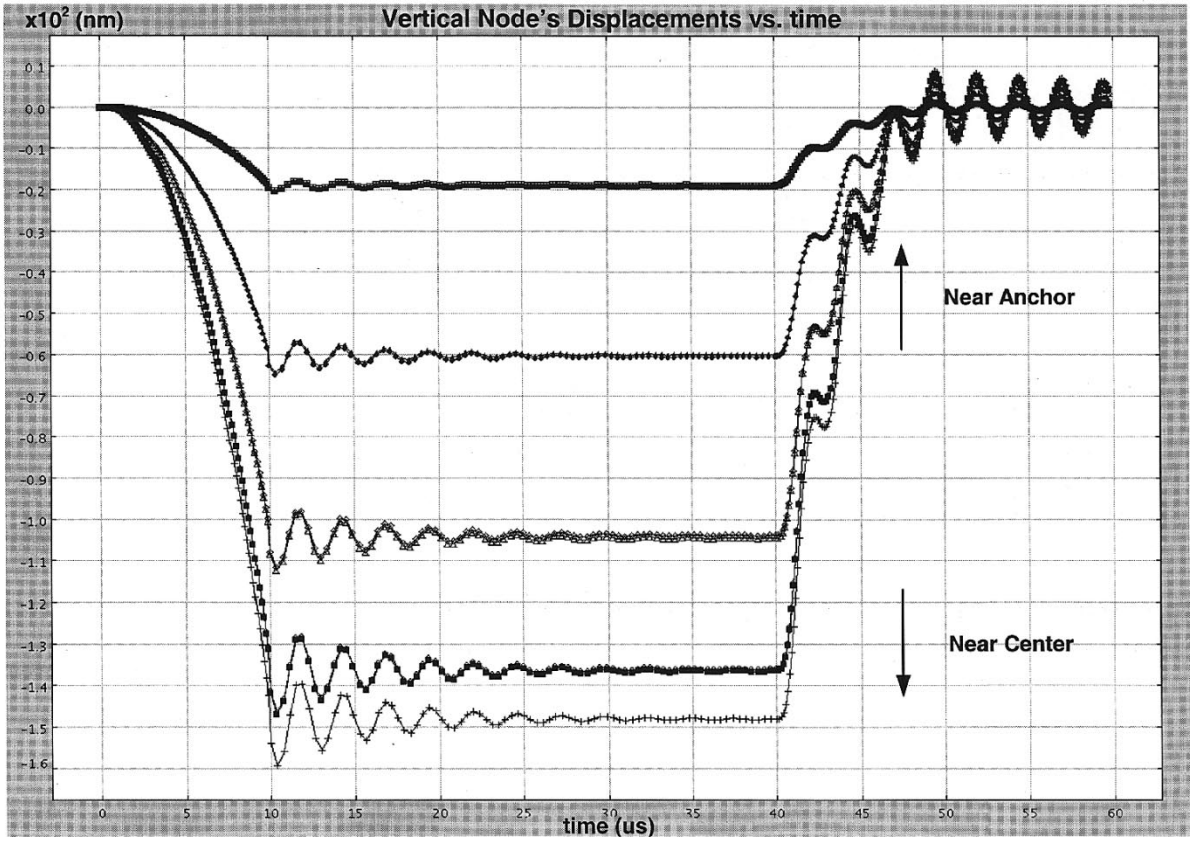


Fig. 15. Nodal displacements in 11-node ribbon model with high-frequency drive signal (10 μs-switching time). Note that symmetrical nodes in the structure (e.g., nodes 2 and 10) show identical responses.

TABLE I
GRATING LIGHT VALVE SYSTEM SIMULATION TIME

	Mesh	128x128			512x512		
Segments		Mech	Optical	System	Mech	Optical	System
5		0.14	1.99	3.33	0.16	40.79	42.37
11		2.02	2.02	5.30	2.02	41.01	44.49
21		15.60	1.98	19.19	15.43	40.78	58.19
41		119.94	1.99	128.81	124.40	40.33	167.68

minimum time step) without recourse to lower level simulation tools.

VI. SUMMARY AND CONCLUSION

We have introduced the challenges in modeling MSMD microsystems. In this paper, we have addressed the need for a consistent behavioral modeling methodology that spans the multiple technologies of electronics, optics, and mechanics. We have also shown how to use PWL models to capture the behavior of nonlinear elements in these domains. Our simulation method at the component level, based on MNA, allows the designer to trade modeling and simulation accuracy for simulation speed. Our angular spectrum scalar representation for free space optical signal propagation allows us to model microoptical components in the near-field and still perform system-level simulations in reasonable time, supporting the system designer in performing design tradeoffs in an interactive design environment.

REFERENCES

[1] Coventor Inc.. [Online]. Available: http://www.coventor.com
[2] MEMSCAP S.A.. [Online]. Available: http://www.memscap.com
[3] J. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt, "Ptolemy: A framework for simulating and prototyping heterogeneous systems," *Int. J. Comput. Simul.*, vol. 4, pp. 155–182, 1994.
[4] J. Morikuni and S. Kang, *Computer-Aided Design of Optoelectronic Integrated Circuits and Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1997, ch. 6.
[5] E. M. Buturla, P. E. Cottrell, B. M. Grossman, and K. A. Salsburg, "Finite element analysis of semiconductor devices: The FIELDAY program," *IBM J. Res. Develop.*, vol. 25, pp. 218–231, 1981.
[6] M. R. Pinto, C. S. Rafferty, and R. W. Dutton, "PISCES II—Poisson and continuity equation solver," Stanford Univ., Stanford, CA, Stanford Electronics Lab. Tech. Rep., Sept. 1984.
[7] P. C. H. Chan and C. T. Sah, "Exact equivalent circuit model for steady-state characterization of semiconductor devices with multiple energy-level recombination centers," *IEEE Trans. Electron Devices*, vol. ED-26, pp. 924–936, 1979.
[8] E. Christen and K. Bakalar, "VHDL-AMS: A hardware description language for analog and mixed-signal applications," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1263–1272, Oct. 1999.
[9] *Verilog*, IEEE Standard 1364-2001.
[10] G. Jacquemod, K. Vuorinen, F. Gaffiot, A. Spisser, C. Seassal, J.-L. Leclercq, P. Rojo-Romero, and P. Viktorovitch, "MOEMS modeling for opto-electro-mechanical co-simulation," *J. Modeling Simulation Microsyst.*, vol. 1, no. 1, pp. 39–48, 1999.
[11] J. J. Morikuni, P. V. Mena, A. V. Harton, and K. W. Wyatt, "The mixed-technology modeling and simulation of optoelectronic microsystems," *J. Modeling and Simulation of Microsystems*, vol. 1, no. 1, pp. 9–18, 1999.
[12] B. Romanowicz, *Methodology for the Modeling and Simulation of Microsystems*. Norwell, MA: Kluwer, 1998, pp. 24–39.
[13] J. E. Vandemeer, "Nodal design of actuators and sensors (NODAS)," M.S. thesis, Dept. Elect. Comput. Eng., Carniege Mellon Univ., Pittsburgh, PA, 1997.
[14] R. K. Gupta, E. S. Hung, Y. J. Yang, G. K. Ananthasuresh, and S. D. Senturia, "Pull-in dynamics of electrostatically-actuated beams," in *Proc. Solid-State Sensor Actuator Workshop, Late News Session*, Hilton Head, SC, June 3–6, 1996, pp. 3–6.

[15] G. K. Ananthasuresh, R. K. Gupta, and S. D. Senturia, "An approach to macromodeling of MEMS for nonlinear dynamic simulation," in *Proc. Dynamics Systems and Controls, ASME Int. Mechanical Engineering Congr. Exposition*, vol. 59, Atlanta, GA, Nov. 1996, pp. 401–407.
[16] A. T. Yang and S. M. Kang, "iSMILE: A novel circuit simulation program with emphasis on new device model development," in *Proc. 26th ACM/IEEE Design Automation Conf.*, 1989, pp. 630–633.
[17] R. Kao and M. Horowitz, "Timing analysis for piecewise linear RSIM," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1498–1512, Dec. 1994.
[18] A. Salz and M. Horowitz, "IRSIM: An incremental MOS switch-level simulator," in *Proc. 26th Design Automation Conf.*, 1989, pp. 173–178.
[19] K. Kawakita and T. Ohtsuki, "NECTAR 2 a circuit analysis program based on piecewise linear approach," in *Proc. ISCAS*, Boston, MA, 1975, pp. 92–95.
[20] M. H. Zaman, S. F. Bart, J. R. Gilbert, N. R. Swart, and M. Mariappan, "An environment for design and modeling of electro-mechanical microsystems," *J. Modeling Simulation Microsyst.*, vol. 1, no. 1, pp. 65–76, 1999.
[21] D. M. W. Leenaerts and W. M. G. Bokhoven, *Piecewise Linear Modeling and Analysis*. Boston, MA: Kluwer, 1998, ch. 4.
[22] D. L. Karnoop and R. C. Rosenberg, *System Dynamics: A Unified Approach*. New York: Wiley, 1975.
[23] J. A. Martinez, "Piecewise linear simulation of optoelectronic devices with application to MEMS," M.S. thesis, Dept. Elect. Eng., Univ. Pittsburgh, , Pittsburgh, PA, 2000.
[24] P. S. Mara, "Triangulations for the cube," *J. Combin. Theory Ser.*, vol. A 20, pp. 170–177, 1976.
[25] W. D. Smith, "A lower bound for the simplexity of the N-cube via hyperbolic volumes," *Eur. J. Combin.*, vol. 21, pp. 131–137, 2000.
[26] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, fourth ed. New York: Oxford Univ. Press, 1998.
[27] L. Velho, L. H. de Figueiredo, and J. Gomes, "A methodology for piecewise linear approximation of surfaces," *J. Brazilian Comput. Soc.*, vol. 3, no. 3, pp. 30–42, 1997.
[28] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.
[29] J. A. Martinez, T. P. Kurzweg, S. P. Levitan, P. J. Marchand, and D. M. Chiarulli, "Mixed-technology system-level simulation," *Analog Integrated Circuits Signal Process.*, vol. 29, pp. 127–149, Oct./Nov. 2001.
[30] J. S. Przemieniecki, *Theory of Matrix Structural Analysis*. New York: McGraw-Hill, 1968.
[31] W. J. Duncan, "Reciprocation of triply partitioned matrices," *J. R. Aeron. Soc.*, vol. 60, pp. 131–132, 1956.
[32] T. P. Kurzweg, S. P. Levitan, J. A. Martinez, P. J. Marchand, and D. M. Chiarulli, "Diffractive optical propagation techniques for a mixed-signal CAD tool," in *Proc. Optics in Computing (OC2000)*, Quebec City, CA, June 18–23, 2000, pp. 610–618.
[33] J. W. Goodman, *Introduction to Fourier Optics*, second ed. New York: McGraw-Hill, 1996.
[34] N. Delen and B. Hooker, "Free-space beam propagation between arbitrarily oriented planes based on full diffraction theory: A fast Fourier transform approach," *JOSA*, vol. 15, no. 4, pp. 857–867, Apr. 1998.
[35] T. P. Kurzweg, "Optical propagation methods for system-level modeling of optical MEMS," Ph.D. dissertation, Dept. Elect. Eng., Univ. Pittsburgh , Pittsburgh, PA, 2002.
[36] D. M. Bloom, "The grating light valve: Revolutionizing display technology," *Proc. SPIE*, vol. 3013, pp. 165–171, 1998.

Steven P. Levitan (S'83–M'83–SM'95) received the B.S. degree from Case Western Reserve University, Cleveland, OH, in 1972, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts, Amherst, in 1979 and 1984, respectively.

He is the John A. Jurenko Professor of Electrical Engineering at the University of Pittsburgh, Pittsburgh, PA. From 1972 to 1977, he was with Xylogic Systems designing hardware for computerized text processing systems.

Prof. Levitan is a member of Optical Society of America, Association for Computing Machinery, and a Senior Member of IEEE/Computer Society. He is an Associate Editor for the *ACM Transactions on Design Automation of Electronic Systems*. In 1998, he became Chair of the ACM Special Interest Group on Design Automation and a member of the Design Automation Conference Executive Committee.

José A. Martínez (S'89) in electrical engineering from the Universidad de Oriente (UDO), Barcelona, Venezuela, in 1993, and the M.S. degree in electrical engineering in 2000 from the University of Pittsburgh, Pittsburgh, PA, where he is currently pursuing the Ph.D. degree in electrical engineering.

Since 1997, he has been with the Optoelectronic Computing Group, University of Pittsburgh. His research interests include behavioral simulation, reduction order techniques, modeling of MEMs and OMEMs, CAD, VLSI, and computer architecture.

Mr. Martínez was granted the José Feliz Rivas' Medal for high academic achievement by the Venezuelan government, in 1993, and scholarships by the Venezuelan Fundayacucho Society, in 1993, and CONICIT-UDO, in 1994. He is a member of IEEE/Lasers & Electro-Optical Society and Optical Society of America.

Timothy P. Kurzweg (S'97–M'03) received the B.S. degree in electrical engineering from The Pennsylvania State University, University Park, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pittsburgh, Pittsburgh, PA, in 1997 and 2002, respectively.

He is an Assistant Professor in the Electrical and Computer Engineering Department, Drexel University, Philadelphia, PA. In the summer of 1999, he worked at Microcosm (now Coventor), in Cambridge, MA, and developed an optical methodology to interface within their system-level analysis tool enabling optical MEM simulation. His research interests include modeling and simulation, optical MEMS, CAD, free-space optics, VLSI, and computer architecture.

Dr. Kurzweg is a member of IEEE/Lasers & Electro-Optical Society, Association for Computing Machinery/Special Interest Group on Design Automation, and Optical Society of America.

Abhijit J. Davare (S'01) received the B.S. degree in computer engineering from the University of Pittsburgh, Pittsburgh, PA, in 2002. He is currently pursuing the M.S./Ph.D. degree in electrical engineering and computer sciences at the University of California, Berkeley.

At the University of Pittsburgh, he participated in the Fessenden Honors in Engineering Program and, as an NSF REU-funded Undergraduate Researcher, he was with the Department of Chemical and Petroleum Engineering and the Department of Electrical Engineering.

Mr. Davare is a recipient of the 2002–2003 California Microelectronics Fellowship, E.M. Heck Honors, and the Kevin Cecil Engineering scholarship. He is a member of Eta Kappa Nu.

Mark Kahrs (S'77–M'84) received the A.B. degree in applied physics and information science (with high honors) from Revelle College, University of California, San Diego, in 1974, and the Ph.D. degree in computer science from the University of Rochester, Rochester, NY, in 1984.

He has held positions at Stanford University, Stanford, CA, Xerox PARC , Institute de Research Acoustique Musique in Paris, Bell Laboratories, and Rutgers University. In 2001, he was a Fulbright Scholar at the Acoustics Laboratory, Helsinki University of Technology. He is currently a Visiting Associate Professor at the Department of Electrical Engineering, University of Pittsburgh, Pittsburgh, PA. His CAD-related interests include Hardware Description Languages for RF and microwave design and circuit synthesis from high level languages.

Michael Bails (S'00) received the B.A. degree in economics in 1995 from the University of Vermont, Burlington, and the B.S. degree (*cum laude*) in electrical engineering in 2002 from the University of Pittsburgh, Pittsburgh, PA, where he is currently pursuing the M.S. degree in electrical engineering.

He was an Undergraduate Researcher in optical MEMs for Benchmark Photonics, a Pittsburgh-based start-up company from 2001 to 2002.

Mr. Bails is a recipient of the Rath Fellowship, Department of Electrical Engineering, University of Pittsburgh. He is a student member of IEEE/Lasers & Electro-Optical Society.

Donald M. Chiarulli received the B.S. degree in physics and the Ph.D. degree in computer science from Louisiana State University, Baton Rouge, in 1976 and 1986, respectively, and the M.Sc. degree in computer science from Virginia Polytechnic Institute and State University, Blacksburg, in 1979.

He is a Professor of Computer Science at the University of Pittsburgh, Pittsburgh, PA, where he has been since 1986. He was an Instructor/Research Associate at Louisiana State University from 1979 to 1986. His research interests are in photonic and optoelectronic computing systems architecture. He is also the co-inventor on three patents relating to computing systems and optoelectronics. He has served on the technical program committees of numerous conferences for both research and education issues.

Dr. Chiarulli's research has been recognized with Best Paper Awards at the International Conference on Neural Networks (ICNN-98) and the Design Automation Conference (DAC-00). He is a member of The International Society for Optical Engineers and Optical Society of America. He serves on the editorial board of the *Journal of Parallel and Distributed Systems*.