

Integration of Mechanical CAD and Behavioral Modeling

Rajarishi Sinha¹, *Student Member, IEEE*, Christiaan J.J. Paredis^{1,2}, *Member, IEEE*, and Pradeep K. Khosla^{1,2}, *Fellow, IEEE*

¹Institute for Complex Engineered Systems, Carnegie Mellon University, Pittsburgh, PA 15213, USA

²Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

This article introduces the concept of combining both form (CAD models) and behavior (simulation models) of mechatronic system components into component objects. By composing these component objects, designers automatically create a virtual prototype of the system they are designing. This virtual prototype, in turn, can provide immediate feedback about design decisions by evaluating whether the functional requirements are met in simulation.

To achieve the composition of behavioral models, we introduce a port-based modeling paradigm where systems consist of component objects and interactions between component objects. To maintain the consistency between the form and behavior of component objects, we introduce parametric relations between these two descriptions. In addition, we develop algorithms that determine the type and parameter values of the interaction models; these models depend on the form of both components that are interacting.

The composable simulation environment has been implemented as a distributed system in Java and C++, enabling multiple users to collaborate on the design of a single system.

Keywords. Mechatronics, VHDL-AMS, Multi-body Simulation, Simulation-Based Design, Mechanical CAD

1 Introduction and Motivation

Because of the intense competition in the current global economy, successful companies must react quickly to changing trends in the market place. For example, the need for a new product can be triggered by the introduction of new technologies, changes in customer demands, or fluctuations in the cost of basic materials and commodities. To capitalize on these imbalances in the market, a company must conceive, design, and manufacture new products quickly and inexpensively. Because the design process consumes a significant portion of the total development time, a shorter design cycle provides a distinct competitive advantage.

The design cycle can be shortened significantly through virtual prototyping. A virtual prototype enables the designers to test whether the design specifications are met by performing simulations rather than physical

experiments; a physical prototype is only needed for final testing. Not only does virtual prototyping make design verification faster and less expensive, it provides the designer with immediate feedback on design decisions. This in turn promises a more comprehensive exploration of design alternatives and a better performing final design. To fully exploit the advantages of virtual prototyping, however, simulation models have to be accurate and easy to create.

Virtual prototypes need to model the behavior of the equivalent physical prototype accurately; otherwise, the predicted behavior does not match the actual behavior resulting in poor design decisions. But creating accurate models is a hard problem. Only recently has computing performance reached a level where high fidelity simulation models are economically viable. For instance, it is now feasible to evaluate dynamic simulations of finite element models for crack propagation.

Equally important to accuracy is the requirement that simulation models be easy to create. Creating high-fidelity simulation models is a complex activity that can be quite time-consuming. To take full advantage of virtual prototyping, it is necessary to develop a modeling paradigm that supports model reuse, that is integrated with the design environment, and that provides a simple and intuitive interface which requires a minimum of analysis expertise. This paper introduces such a paradigm, composable simulation, which is based on model composition from system components.

2 Composable Simulations

To provide better support for simulation-based design of mechatronic systems, we have developed a modeling and design paradigm based on composition. A wide variety of products, ranging from consumer electronics to cars, contain mostly off-the-shelf components and components reused from previous design generations; for instance, in cars, the portion of completely new components is often less than twenty percent. As a result, the design of such systems consists primarily of the configuration or assembly of existing components.

The modeling of such systems can also be viewed as composition. We can obtain a system level simulation

model, by combining the component models with the models that define the interactions between the components. Assuming that the models for individual components already exist in a component library, and that the physics of the interactions between the components have been modeled in a library of interaction models, a system level simulation model can be generated through the composition of existing component and interaction models.

To take advantage of the parallelism between the design and modeling activities—both consist of the composition of system components—we have developed a modeling and design framework in which the form and the behavior of a component are combined into a single component object. By composing component objects into systems, a designer simultaneously designs and models new artifacts. This is already common practice in electrical CAD software; when creating a chip layout, the instantiation of a transistor or logic gate creates the geometry for the silicon layers as well as the corresponding simulation model.

In mechanical CAD, the integration between design and simulation is not as common. For purely mechanical systems, most commercial CAD packages do provide an optional module for multi-body simulation, but these modules lack sufficient support for multi-disciplinary systems.

Our goal is to extend this design paradigm to multidisciplinary systems, specifically mechatronic systems. The traditional design approach for multidisciplinary systems has been a sequential design-by-discipline approach: First design the mechanical system, then the sensors and actuators, and finally the control system [16]. This approach imposes artificial constraints by fixing the design at various points in the design sequence. In mechatronic design, on the other hand, synergy between the different disciplines is achieved by designing all disciplines concurrently. To evaluate whether a mechatronic design prototype meets the design specifications, the designer must consider the component interactions in all energy domains. This would be prohibitively expensive without the intensive use of simulation.

Existing simulation tools for multidisciplinary systems are very general, stand-alone tools that are not integrated with the design environment. The main goal of the simulation and design environment that we have developed is to support multidisciplinary simulation-based design within an integrated software environment. Specifically, the framework has the following characteristics, which we will address in detail in the subsequent sections:

A port-based modeling paradigm: To take advantage of the compositional nature of both design and modeling of mechatronic systems, we have developed a port-based modeling paradigm in which the user can compose system-level simulations from component models. By connecting the ports of the subcomponents, the user defines the interactions between them. In Section 4, we will describe the port-based modeling paradigm in more detail.

Simulation integrated with CAD: The building blocks in our simulation and design environment are component objects; they describe both the form and the behavior of system components. In Section 5, we describe how the CAD description of the form may be used to extract the lumped parameters of the behavioral models. In addition, we have developed algorithms that instantiate models of mechanical interactions based on the form of the interacting components.

3 Related Work

3.1 Simulation-based Design

Many companies are resorting to simulation tools to improve their design process. A well-publicized example of virtual prototyping is the design of the Boeing 777 airplane. Boeing switched from a paper-based design process to a fully digital representation, allowing them to perform performance analysis (using CFD software) and assemblability analysis without the need for building physical prototypes. This resulted in a significantly shorter design and testing period. A similar all-digital approach is being adopted by car manufacturers.

Although the success of simulation-based design has already been demonstrated commercially, many unresolved research issues remain to be addressed. Ongoing research includes model validation [1], automatic meshing and model creation [22], integration of simulation engines in different domains [21], architectures for collaboration [9], and visualization using virtual reality technology [4]. In this paper, we focus on simplifying the process of model creation, by integrating form and behavior into component objects.

Our approach is based on the characterization of a design prototype by its form, function, and behavior [14]. The form is a description of the physical embodiment of an artifact, while function is the purpose of the artifact—the behavior that the designer intended to achieve. As is illustrated in Figure 1, the actual behavior does not depend on the function, but only on the form. During design or synthesis, we transform function into form, while, during

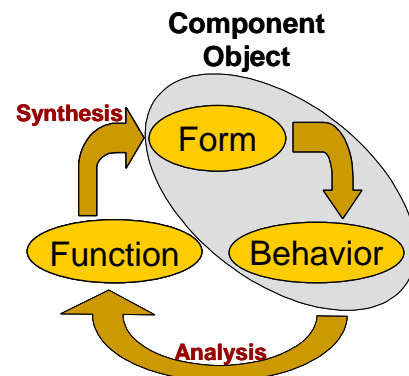


Figure 1. Encapsulation of form and behavior in a component.

design verification, we derive the behavior from the form and verify whether this behavior matches the function. In the context of virtual prototyping, the behavior is described by mathematical models and design verification is achieved by performing simulation experiments with these models.

3.2 Multibody Simulation

Baraff [2] used algorithmic methods to simulate the mechanical dynamics of multi-body systems with constraints. Such an approach involves setting up the ordinary differential equations (ODEs) that govern the dynamics of the multi-body system, and solving them using variable step numerical methods. Our framework extends this approach by allowing for the composition of models, or hierarchical systems, and for the easy definition of joint constraints. Orlandea et al. [12, 13] showed that springs and dampers could be modeled using sparse systems of linear equations. This work was subsequently incorporated in the ADAMS system [11]. However, unlike ADAMS, our framework can automatically derive the behavioral models of the components from the geometry.

3.3 Parameter Extraction from CAD

Techniques have been developed to predict the instantaneous degrees of freedom from the CAD models of parts composed of polygonal planar faces [10]. These techniques handle only parts with planar faces; most mechatronic devices have curved parts. When curved parts are approximated as piecewise planar parts, erroneous results are likely. In previous work [17, 18], we reason about the degrees of freedom at each joint, based on surface mating constraints that are in turn obtained from analyzing the nature of body-to-body contact. We obtain a set of properties that must be satisfied by a general contact surface in order to obtain linear models. We describe a method by which the space of allowable motions in the device can be described concisely.

4 Port-Based Modeling Paradigm

To achieve composability of behavioral models, we have developed a port-based modeling paradigm. This paradigm is based on two concepts: ports and connections [5].

Ports correspond to the points where a component exchanges energy with the environment - one port for each separate interaction point. The interactions between components are represented by connections between ports. Each connection imposes algebraic constraints on the port variables. These constraints are the equivalents of the Kirchoff voltage and current laws in electrical circuits. One type of constraint requires that the across variables be equal, the other that the sum of the through variables be zero.

As a reflection of the underlying physics, both connections and ports are undirected. In mathematical terms, this requires that the components and connections be modeled as declarative equations rather than assignments.

Many recent simulation languages are declarative, including Modelica [7], VHDL-AMS [8], and Dymola [6]; SimuLink [20], on the other hand, is procedural.

All the ports combined form the interface of the model. This interface defines how the component can interact with the other components in the system, but does not contain any information about the internal behavior of the component. Instead, the interface encapsulates the implementation of the model, which defines the internal behavior of the component

The port-based modeling paradigm cannot be applied to all systems; it is limited to systems with lumped interactions. When an interaction is distributed in nature, as between a boat and the water on which it floats, it must be approximated by a large number of lumped interactions. The internal model of a component, however, may still be distributed. Consider, for example, a flexible beam attached to a structure by its two ends. A finite element model may describe the internal behavior of the beam, but the interaction with the structure can still be captured with only two ports. For mechatronic systems, the primary interactions between components tend to be lumped, so that the port-based modeling paradigm is applicable. Only when very detailed models are required, may we have to consider phenomena, such as thermal interactions, that are distributed in nature.

We have implemented the port-based modeling approach for the electrical, mechanical, and signal domains [5, 19]. This paper focuses on the mechanical behavior.

5 Mechanical Behavioral Modeling

The mechanical behavior of a three dimensional rigid body is completely described by the position and orientation of the body (across variables), and the forces and torques acting on the body (through variables). Since a rigid body component has only one set of across and through variables, it has a single mechanical port. A common behavioral model for a rigid body is the Newton-Euler model that relates the across and the through variables of the port. This model is parameterized by a point mass at the center of gravity and an inertia tensor that captures the mass distribution. All positions and orientations of the body are expressed relative to a global frame of reference.

A rigid body in a multi-body system is constrained in its motion by other rigid bodies. The constraint is imposed on the port variables of the bodies and is represented by a joint component with two ports. The behavioral model of the joint component relates the variables of the two ports and captures a kinematic or dynamic relationship between the rigid body components. The joint types that are currently supported include the lower pairs, gear pairs and rack-and-pinion.

Two rigid body models are never connected directly to each other; they are connected through a joint component. When the port on a mass component is connected to a port on a joint component, a *node* is implicitly created, and the

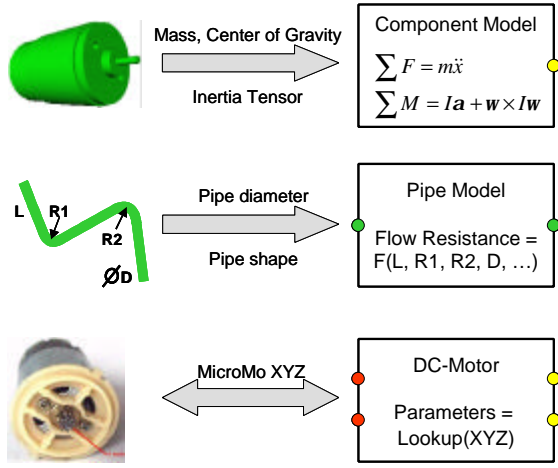


Figure 2: The relation between form and behavior parameters.

two ports are connected to this node. A node introduces the following connection constraints:

$$\begin{aligned} \vec{p}_A &= \vec{p}_B = \dots = \vec{p}_i \\ R_A &= R_B = \dots = R_i \end{aligned} \quad (1)$$

and

$$\begin{aligned} \sum_i \vec{F}_i &= \vec{0} \\ \sum_i \vec{\tau}_i &= \vec{0} \end{aligned} \quad (2)$$

where $\left\langle \underbrace{p_i, R_i}_{\text{across}}, \underbrace{F_i, \tau_i}_{\text{through}} \right\rangle$ are the mechanical port variables for mass i .

In a joint component model, the two ports of the joint are related through an algebraic or a differential-algebraic equation. A rigid joint, for instance, would define a constant transform between the positions of the two ports. A revolute joint has a single degree of freedom—a pure rotation about an axis. Written in the homogeneous transform notation, we have:

$${}^{F_1}T_{F_2} = R_K(\mathbf{q}) \quad (3)$$

where $R_K(\mathbf{q})$ is the rotation transform about the rotation axis $\hat{A} = [k_x, k_y, k_z]$.

6 Relation between Behavior and Form

Composable simulations are based on the concept of component objects that combine form and behavior. By composing component objects into systems, a designer simultaneously designs and models new artifacts. The previous two sections introduced a modular modeling paradigm that supports such composition. In this section, we focus on guaranteeing that these behavioral models are

consistent with the corresponding form descriptions as represented by a CAD model.

We distinguish between two different types of behavioral models: models representing physical components, and models representing interactions between components. Examples of physical components are motors, screws, shafts, or controllers. Their component objects contain a description of both form and behavior. Interaction models, on the other hand, do not have any associated form. Yet, their model parameters can be extracted from the form of the two interacting components. Examples of interaction models are lower pairs that result from surface-to-surface mechanical contact, contact resistance in an electrical switch, or magnetic forces between two magnets.

6.1 Form and Behavior of Component Objects

A component object contains a description of the form of the component as well as a model describing its behavior. Ideally, behavioral models are generated from the form automatically. This requires combining information about geometry and materials with knowledge of the physical phenomena occurring in the component. Creating such models automatically is too difficult in the general case, but can be achieved for certain classes or families of components. For example, the mechanical behavior of the set of rigid bodies with homogeneous material properties is completely defined by the mass and inertial parameters, as is shown in Figure 2. These rigid bodies are so common in mechatronic systems, that it makes sense to develop a procedure that computes the inertial parameters from the density and the geometry of the component, as defined in a CAD model. As a result, the behavior models of homogeneous rigid bodies can be derived automatically for any material and arbitrary geometry.

Besides rigid bodies, we can automatically generate behavioral models for parametric models. In a parametric CAD model, the designer establishes relationships between certain geometric dimensions or parameters. As a result, the form is completely defined by a limited set of characteristic parameters or features. Behavioral models also contain parameters, which, in turn, can be related to the CAD parameters. These relations can be simple, as in the rigid body example, or can be quite complicated, as for a hydraulic pipe. As is illustrated in Figure 2, the flow resistance of the pipe depends on its length, diameter, and bending radii. Although these dimensions may not be defined explicitly in the CAD model, they can be extracted through parametric relations captured as procedures [3, 15].

Finally, one can consider the case in which both geometric and behavioral parameters are determined through a lookup table. For instance, given the model type of a DC motor, a lookup table provides all the parameters for a detailed behavioral model. Similarly, a parametric CAD model is instantiated from parameter values in the lookup table. What makes this example significantly different from the previous example is that there may not be any direct relation between the geometric parameters and

the behavioral parameters. The geometry may simply be a high-level abstraction of the DC-motor, capturing only the external geometry through which the motor can interact with other components. These simplified geometric representations of the form no longer contain any relevant information from which an internal behavioral model can be extracted.

6.2 Form and Behavior of Component Interactions

In addition to the behavioral models of component objects, systems include models describing the interactions between component objects. For each pair of interacting component objects, there is an interaction model that relates the port variables of the two objects to each other.

Any interaction in any energy domain requires an interaction model. However, for the electrical domain, the interaction model is usually very simple. An electrical connection between two components is modeled sufficiently accurately by constraining the voltage at the two connecting ports to be equal and the current through them to add to zero. Because this interaction model is so common, we allow it to be omitted in our modeling paradigm, as is shown in Figure 3. In the mechanical domain, the equivalent default model is rarely appropriate. Even when connecting two components rigidly, their reference frame is usually in a different location so that a model representing the coordinate transformation is needed.

Besides rigid connections, other common mechanical interaction models are the lower pair kinematic constraints. We have developed algorithms to extract the type and parameters of a lower pair from the geometry of the interacting components [19].

Previously, kinematic analysis was limited to parts with only planar faces [10]. Since most engineering devices contain curved surfaces, these analyses either did not apply or failed to recognize certain degrees of freedom when approximating the curved faces with polygonal facets.

In our work [17, 18], we have extended these results to curved contacts, as is shown in Figure 4. When two rigid parts share a surface-to-surface contact, every contact point is subject to a non-penetration condition. This condition requires that the instantaneous velocity between the two bodies does not have a component in the direction opposite to the surface normal at the contact point. We write this

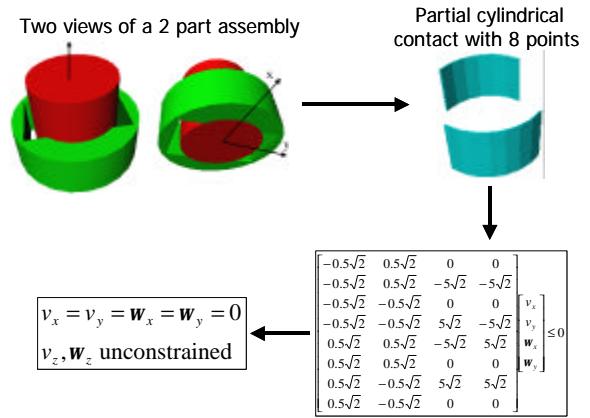


Figure 4: Extracting the type and parameters for lower pair interaction models.

condition as a linear inequality of the form:

$$(\vec{v} + \vec{w} \times \vec{r}) \bullet \vec{n} \geq 0, \quad (4)$$

where \vec{v} and \vec{w} are the relative translational and angular velocities between the two bodies, \vec{r} is the position of the point, and \vec{n} is the normal to the contact surface. Imposing Equation (4) at every point on the contact surface is equivalent to imposing the constraint at the vertices of the convex hull. Therefore, the analysis results in a linear relationship of the form:

$$J_{assembly} \begin{bmatrix} \vec{v} \\ \vec{w} \end{bmatrix} \geq 0, \quad (5)$$

where each row of $J_{assembly}$ represents a non-penetration constraint, as in Equation (4). From the properties of the $J_{assembly}$ matrix, we can determine the kinematic constraints between two interacting component objects. For example, the basis vectors of the nullspace of $J_{assembly}$ define the contact-preserving degrees of freedom.

Our method can infer behavior from devices with curved geometry, while at the same time resolving global, multi-part interactions. We have developed procedures that derive the $J_{assembly}$ matrix directly from the CAD models, and from it determine the type and parameters of the interaction models.

7 Conclusions

To support simulation-based design, we have developed a design environment in which design and modeling are tightly integrated. This integration is based on *component objects* that combine descriptions of both form and behavior of system components. By composing component objects into systems, the design team simultaneously designs and models new artifacts. To enable this composition, we have developed a modular port-based modeling paradigm. The integration between

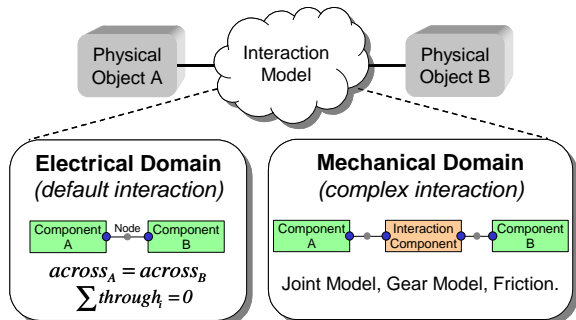


Figure 3: Interactions between system components.

form and behavior is further enhanced by defining relationships between CAD and behavioral parameters. To extract the parameters of interaction models from the form of interacting components, we have developed procedures that automatically determine the type and parameters of lower pair mechanical interactions. We instantiate parameterized VHDL-AMS behavioral models for rigid bodies and the interactions between rigid bodies in a multi-body system.

8 Acknowledgments

This research was funded in part by DARPA under contract ONR # N00014-96-1-0854, by the National Institute of Standards and Technology, by the National Science Foundation under grants # CISE/115/KDI 98 73005 and # EIA-97 29827, by the Pennsylvania Infrastructure Technology Alliance, and by the Institute for Complex Engineered Systems at Carnegie Mellon University.

9 References

- [1] Balci, O., "Principles of Simulation Model Validation, Verification, and Testing," *Transactions of the SCS*, vol. 14, pp. 3-12, 1998.
- [2] Baraff, D., "Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies," *SIGGRAPH: Computer Graphics*, vol. 23, pp. 223-232, 1989.
- [3] Bettig, B., Summers, J. D., and Shah, J. J., "Geometric Exemplars: A Bridge between Cad and Ai," presented at The Fourth IFIP Working Group 5.2 Workshop on Knowledge Intensive CAD (KIC-4), Parma, Italy, 2000.
- [4] Dani, T. H. and Gadh, R., "Networked Virtual Reality for Collaborative Design," presented at 31st CIRP International Seminar on Manufacturing Systems, Berkeley, CA, 1998.
- [5] Diaz-Calderon, A., Paredis, C. J. J., and Khosla, P. K., "A Composable Simulation Environment for Mechatronic Systems," presented at SCS 1999 European Simulation Symposium, Erlangen, Germany, 1999.
- [6] Dynasim AB, "Dymola," Lund, Sweden, 1999.
- [7] Elmqvist, H., Mattsson, S. E., and Otter, M., "Modelica: The New Object-Oriented Modeling Language," presented at The 12th European Simulation Multiconference, Manchester, UK, 1998.
- [8] IEEE, "1076.1 Working Group Analog and Mixed-Signal Extensions for Vhdl,": IEEE, 1999.
- [9] Ivezic, N., "Simulation-Based Support for Early Collaborative Design," in *Civil and Environmental Engineering*. Pittsburgh: Carnegie Mellon University, 1995, pp. 158.
- [10] Mattikalli, R., Baraff, D., Khosla, P., and Repetto, B., "Gravitational Stability of Frictionless Assemblies," *IEEE Transactions on Robotics and Automation*, vol. 11, pp. 374-388, 1995.
- [11] Mechanical Dynamics Inc., "Adams,"., 1999.
- [12] Orlandea, N., Chace, M. A., and Calahan, D. A., "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - Part 2," *Journal of Engineering for Industry*, pp. 780-784, 1977.
- [13] Orlandea, N., Chace, M. A., and Calahan, D. A., "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - Part 1," *Journal of Engineering for Industry*, pp. 773-779, 1977.
- [14] Pahl, G. and Beitz, W., *Engineering Design: A Systematic Approach*, 2nd Edition ed. London, U.K.: Springer-Verlag, 1996.
- [15] Shah, J. J. and Mantyla, M., *Parametric and Feature-Based Cad/Cam: Concepts, Techniques, Applications*. New York, NY: John Wiley & Sons, 1995.
- [16] Shetty, D. and Kolk, R., *Mechatronics System Design*, 1997.
- [17] Sinha, Gupta, Paredis, and Khosla, "Extracting Articulation Models from Cad Models of Parts with Curved Surfaces," Carnegie Mellon University, Pittsburgh, Research Report April 8 1999.
- [18] Sinha, R., Paredis, C. J. J., Gupta, S. K., and Khosla, P. K., "Capturing Articulation in Assemblies from Component Geometry," presented at ASME Design Engineering Technical Conference, Atlanta, GA, 1998.
- [19] Sinha, R., Paredis, C. J. J., and Khosla, P. K., "Kinematics Support for Design and Simulation of Mechatronic Systems," presented at The Fourth IFIP Working Group 5.2 Workshop on Knowledge Intensive CAD (KIC-4), Parma, Italy, 2000.
- [20] The Mathworks Inc., "MATLAB/SIMULINK,"., 1999.
- [21] Tsai, Chang, and Wang, "Integration Infrastructure for a Simulation-Based Design Environment," presented at ASME Computers in Engineering Conference, 1995.
- [22] Wilson, Pinsky, and Dutton, "Investigation of Tetrahedral Automatic Mesh Generation for Finite-Element Simulation of Micro-Electro-Mechanical Switches," presented at International conference on modeling and simulation of microsystems, semiconductors, sensors and actuators, San Juan, 1999.