Behavior Generation using Model Switching
A Hybrid Bond Graph Modeling Technique

Pieter J. Mosterman
Center for Intelligent Systems
Vanderbilt University
Nashville, Tennessee 37235
pjm@vuse.vanderbilt.edu

Gautam Biswas
Department of Computer Science
Vanderbilt University
Nashville, Tennessee 37235
biswas @vuse.vanderbilt.edu

Abstract

This paper discusses a technique for modeling discontinuous physical systems that combines bond graph energy-flow modeling and signal-flow modeling schemes. This enables the generation of complex, multi-mode behaviors without violating the energy flow principles imposed by bond graphs. Mode switching is achieved by *controlled junctions* which assume one of two states: *on* and *off*. The control function of the junctions are specified as state transition graphs or tables. The modeling methodology presents a common framework for modeling abrupt switching elements, piecewise linear components, and subsystems that undergo structural changes.

1 Introduction

Recent advances in model-based and qualitative reasoning have led to researchers developing large scale models of complex, continuous systems, such as power plants, aircraft, and space station sub-systems. Configuration and individual component description changes cause multi-mode behavior in complex systems[4]. A primary reason for component description changes is the representation of complex non linear behaviors as simpler, discontinuous, piecewise linear behaviors[13].

Biswas and Yu[2] have discussed the effectiveness of the bond graph modeling language in defining formal but generic models of physical systems. These models are based on energy transfers between mechanisms of the system and do not gracefully accommodate mode-switching of components as part of the modeling framework. This paper extends previous work on (i) finite state models for representing discontinuous behavior by Broenink and Wijbrans[4], and (ii) ideal switching elements by Strömberg, Top, and Söderman[16, 17] by developing a signal flow model on primary bond graph models to incorporate *mode-change* behaviors in systems. Signal flow models incorporating finite state automata have been used extensively in real-time system analysis[7]. Finite state automata can also model ideal switching elements with complex sequential behavior while avoiding spurious junctions, incorrect causal relations, but preserving the relation between bond graph mechanisms and corresponding physical components of the system. Therefore, they provide an integrated framework for hierarchical modeling and behavior analysis of complex, continuous systems.

2 Modeling Physical Systems

The ability to dynamically construct parsimonious system models is the key to developing computationally efficient behavior generation methods. This requires model building methodologies that satisfy *compositionality*, *graceful extendibility*, and *genericity*. *Bond graphs*[14, 15] introduce the notion of generic physical state variables in modeling and provide a well-defined formal vocabulary for describing system components and their interactions. This creates a framework for building physical models over a wide range of domains[18].

Systematic procedures exist for building bond graph models of physical systems, deriving causal structure from the model[15], and building complex system models by composition[2]. Analytic models which express system behavior as a set of first order differential equations[15] form the basis for deriving a spectrum of qualitative causal to quantitative models from this framework. Compared to other qualitative reasoning approaches, bond graphs use additional energy conservation and continuity constraints to dramatically reduces the number of spurious behaviors generated[16, 18]. Bond graphs support hierarchical model construction schemes by allowing compilation of these models from a library of sub-systems[2].

Typically, qualitative simulation methods (e.g., [6, 10]) impose *continuity constraints* to ensure meaningful behavior generation. Further, to reduce computational complexity, constraint behavior models are *linearized.* However, in a number of real world situations, physical components, such as electric switches, hydraulic valves, diodes, and mechanical clutches, exhibit *abrupt* changes in behavior. In other situations, complex non linear behavior, such as non linear oscillators and transistors, is simplified and represented as *multiple-piecewise* linear behaviors. The corresponding physical component or device is said to operate in *multiple-modes*[1]. A third kind of change that can occur is abrupt changes in structure, such as the breaking of pipes and wires, and overflowing tanks. In qualitative simulation systems, such as QSIM[10], a set of QDE's define the continuous behavior of a single mode of the system. Discontinuous changes are handled by defining multiple QDE sets, and a higher level control structure (meta-model) determines when to switch QDE sets during behavior generation.

Broenink and Wijbrans[4] and Nishida and Doshita [13] classify discontinuities into different classes and propose individualized behavior generation techniques for each class. We contend that all discontinuities modeled in physical systems can be attributed to *abstracting* component behavior to simplify the relations among parameters or to simplify the time-scale of the interactions (e.g., bouncing ball). Therefore, our goal is to derive a uniform approach to analyzing discontinuous system behavior.

Nishida and Doshita[13] point out two properties of discontinuous systems: (i) the causal structure of the system may change during a discontinuous change, and (ii) a number of discontinuous changes may occur one after another. The change of causality, which may be drastic when a series of changes are involved, can be derived once the resulting energy-flow bond graph is established[3]. It is often not clear how to propagate a series of discontinuous changes. Since ordinary equations do not support this kind of information, this is a source of problems when employing a related modeling approach. Therefore, a dedicated formalism addressing this issue in the bond graph framework is required.

3 Bond Graphs and Discontinuous Behavior

Like qualitative simulation approaches, the bond graph methodology is geared toward modeling and analysis of of continuous physical systems. To accommodate abrupt switching and non linear behavior, basic switches modeled as non linear resistive elements have been introduced to augment the bond graph framework.

The first approach uses *modulated transformers* (MTF)[9, 15] with parameter m assigned a value 1 for a *closed* switch and a value 0 for an *open* switch. Top[19] has shown that the open switch configuration may produce incorrect flow values at the junction the transformer is connected to. Causal relations are also violated when a switch changes state.

A second approach, *implicit switching*[8], is based on almost *ideal* switching behavior using non linear R elements. Additional constraints are added to the input of the resistor in one of the switch modes. Fixed causal structure is achieved, and except for the additional constraints imposed on the resistor, equations defining system behavior are derived by standard methods. However, in the conductive *(on)* mode of the switch, the resistor dissipates energy, and, therefore, ideal switching behavior cannot be modeled. Moreover, unlike traditional bond graphs the R element is no longer strictly acausal; it imposes causal direction on the rest of the

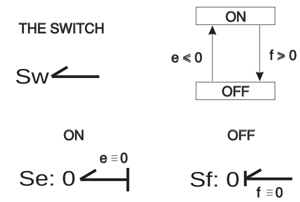

Figure 1: **The Switch**

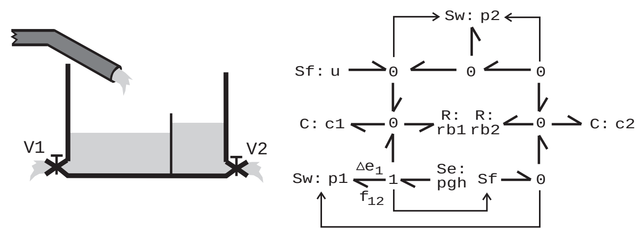

Figure 2: **The bi-tank and its initial bond graph**

model.

Broenink and Wijbrans[4] introduce the concept of *switching bonds*, which are used to connect or disconnect sub-models based on principles of energy transfer. The switching bond is connected between two junctions and their positions are determined by a systematic modeling process. A *control box*, implemented as a global finite-state automaton, connected to the switching bond determines its *on* and *off* states. The shortcomings of this methodology are that it can cause hanging junctions, and changing boundary conditions due to switching are incorrectly handled.

To address these problems, Strömberg, Top and Söderman[16, 17] introduce an ideal switching element into the bond graph modeling language. A graphical description of the switch is shown in Fig. 1. Discontinuous changes are handled by enforcing either *zero flow* (switch *off*) or *zero effort* (switch *on*) on a junction. In both these modes, $power = effort \times flow = 0$, which is recognized to be the key in representing physical discontinuities using an ideal switching element[19]. Note that the *on* and *off* modes of the switch are degenerate forms of the effort- and flow-source elements of the bond graph, respectively.

A successful application of this approach, modeling the discontinuities of a bi-tank system (Fig. 2), is described in [17]. This system consists of three basic modes of operation: (i) both the sub-containers are filled below the top of the interconnecting wall, (ii) the left container is overflowing, and (iii) both the sub-containers have a fluid-level that is over the top of the wall. Switching between these modes is taken care of by the two switches with associated control relations p_1 and p_2 (Fig. 3). c_1 and c_2 are the capacitance of the left and right compartments, respectively, rb_1 and rb_2 are corresponding valve resistances, $S_f : u$ is the inflow pipe which is modeled as a flow source. h is the height of the compartment wall.

This representation of the switch handles a rudimen-

Figure 3: **Control relations for switches**

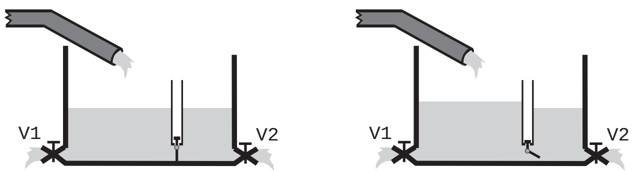

Figure 4: **The bi-tank system with memory**

tary form of *combinatorial control logic*, but does not address aspects of sophisticated *sequential control logic* that can occur in physical systems. For example, consider a more complex bi-tank system shown in Fig. 4. To avoid unnecessary complexity, the effect of the left compartment overflowing is not considered in the discussion that follows. The latch between the tanks can go either way from its initial upright position when the *pressure differential* between the tanks exceeds a threshold value p_t. An open switch can close, but it does not open in the opposite direction. For example, if the pressure differential caused the switch to open to the right, it remains open in that direction but closes if the pressure differential becomes 0. If the pressure differential reverses, the latch does not open to the left. This behavior can only be modeled using sequential logic, because the *open state* of the switch determines all future behavior. Fig. 5 illustrates the corresponding state-transition diagram.

Ideal switches implement pure combinational logic, therefore, cannot implement the control logic of the latched bi-tank system. Furthermore,

- it is unnatural to consider switches as bond graph elements; unlike other bond graph elements they have no energy-related functions,
- switches represent *transient* elements; their behavior is based on control logic rather than physical concepts, and
- a related point is that they obscure hierarchical

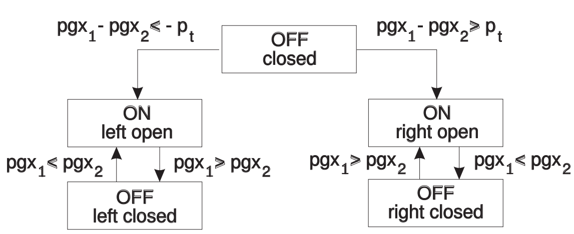

Figure 5: **CSPEC State Transition Diagram: Complex bi-tank system**

compositional modeling schemes, which allow a system to be defined in terms of multiple models, each model representing an individual mode of system behavior. Dynamic behavior generation may involve model-switching to account for the different behavioral modes. A number of ideal switching elements may be employed to achieve model switching dynamically, but, in this representation, the link between bond graph elements and parts of the physical system become obscure and non-intuitive.

Our solution to this problem is to employ a hybrid representation scheme that combines

1. traditional energy-related bond graph elements to model the physical components of the system, and
2. control flow models based on switching junctions whose *on–off* characteristics are modeled by finite state automata.

This work extends intuitive notions presented in [1].

4 The Hybrid Modeling Approach

Previous work on real-time systems[20] demonstrates that system behavior is a function of time and event dependency, therefore, these system models must incorporate dynamic views with different system configurations and states. In general, a system with n such components each with k behavior modes, can assume k^n overall configurations or behavior modes, however, in practical situations, only a small fraction of the modes are physically realized. In previous approaches, e.g., [1, 4, 10], this knowledge has been exploited to pre-enumerate global behavior modes and specify transition functions between the modes as rules or state transition tables. However, this approach will not work for systems whose range of behaviors have not been pre-enumerated. Recent compositional modeling approaches[5, 12] overcome this problem and build system models *dynamically* by composing model fragments. Our approach adopts this methodology, and implements a dynamic but rigorous model switching methodology in the bond graph modeling framework.

Instead of identifying a global control structure and pre-enumerating bond graph models for each of these modes, we translate the overall physical model to one bond graph model that covers the energy flow relations within the system. Next, the discontinuous mechanisms and components in the system are identified. These discontinuities are modeled locally as a *controlled junction* which can assume one of two states – *on* and *off*[1]. A local control mechanism, implemented as a *finite state automaton* and represented as a *state transition graph* or *table*, is used to model each discontinuity. A controlled junction behaves like an *idealized switch*. The input to its control mechanism are *effort* values from selected *0-junctions* and *flow* values from selected *1-junctions* and its output is a control signal that determines the *on/off*

[1]Traditional bond graph junctions are always in the *on* state.

state of the junction. The set of local control mechanisms associated with controlled junctions constitute the *signal flow model* of the system. The signal flow model performs no energy transfers, therefore, it is distinct from the bond graph model that deals with the dynamic behavior of the physical system variables. Interactions between the two models are established by tapping appropriate effort and flow energy variables from bond graph junctions and generating output signals that determine the *on* and *off* states of controlled junctions. To summarize, bond graph models deal with energy-related behavior whereas signal flow models describe the *transient*, i.e., mode-switching behavior of the system.

A *mode* of a system is determined by the combination of the on/off states of all the controlled junctions in its bond graph model. Each mode defines a physically valid energy model whose bond graph elements can be mapped back to components and mechanisms of the physical system. Note that the system modes and transitions are dynamically generated, and do not have to be pre-enumerated.

4.1 The Modeling Language

The modeling language has two components:

1. traditional energy-related bond graphs[9, 15], and
2. finite state automata for modeling controlled junctions.

Bond graphs are *well-grounded* in physical reality[2, 15, 19] and finite state automata represent a formal methodology for modeling time-varying discrete systems. The combination of these two models which interact through the controlled and regular junctions define overall system behavior.

4.2 Controlled Junctions

When active (*on* state), controlled junctions behave like normal 0- or 1-junctions. Note that for a 0-junction the effort value associated with all connected bonds are equal and the flow values sum to 0. For a 1-junction the flow values are equal and the effort values sum to 0. A deactivated (*off*) 0-controlled junction forces the effort value at all connected bonds to become 0, implying that there is no energy transfer across this junction. Similarly, a deactivated 1-junction forces flows to become 0. In both cases, the controlled junction exhibits ideal switch behavior, and modeling discontinuous behavior in this way is consistent with bond graph theory[19]. Deactivating controlled junctions can affect the behaviors at adjoining junctions.

Controlled junctions are marked with subscripts (e.g., 1_1, 0_2) and the corresponding control logic is depicted as dark circles in the bond graph representation (Fig. 6). Their input, effort and flow energy variables tapped from 0- and 1-junctions, respectively are shown as arrows. Their output, a single control signal (shown as a dotted line), sets the associated controlled junction to *on* and *off*. The junctions define the *interactions* between the energy-flow and signal-flow models of the system.

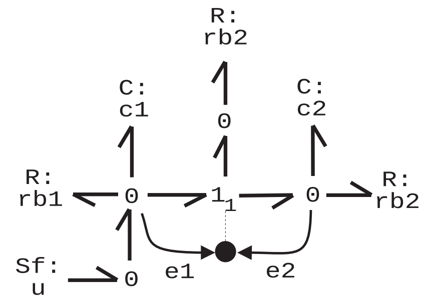

Figure 6: **Bond Graph Model: Complex bi-tank system**

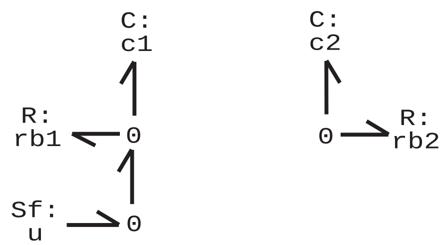

Figure 7: **Bond Graph Model: Initial System State**

The control logic for determining the *on* and *off* state for a controlled junction is implemented as a finite state automata, and represented as a state transition graph or table. The two representations are logically equivalent. We call this *control specification* function *CSPEC* which is actually a finite state machine whose internal communication is asynchronous and overwriting. The CSPEC representation for the complex bi-tank system is shown in Fig. 5.

Each controlled junction has one associated state transition diagram with at least two states describing its *on* and *off* modes. However, one junction may have several internal states that map onto the *on* and *off* modes.

The use of controlled junctions is illustrated in the bond graph model (Fig. 6) of the complex bi-tank system (Fig. 4). If the latch is closed, i.e., there is no flow of liquid, the 1_1 junction is deactivated, and the flow at this junction is now 0. Energy transfer across this junction becomes 0, which implies that the bonds incident on the junction can be eliminated in this system mode. The net result is that the two compartments of the tank become independent subsystems thus producing a seamless implementation of the mode-switching process.

5 Simulating the latched bi-tank

The simulation model of the latched bi-tank system was developed under Microsoft Windows using Visual Basic 3.0 Professional Edition[11]. The system model is incorporated as a matrix of equations which are derived from the bond graph model manually and inverted using Mathematica. Note that this derivation process is already fully automated in systems like CAMAS[3]. In-

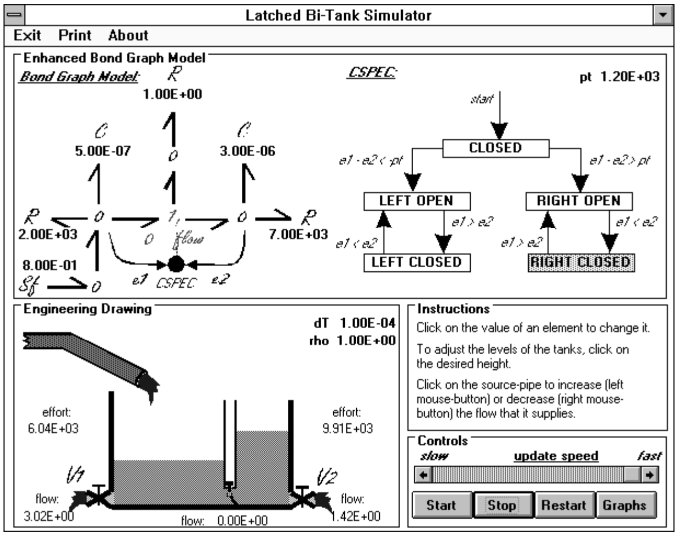

Figure 8: **The Simulator Interface**

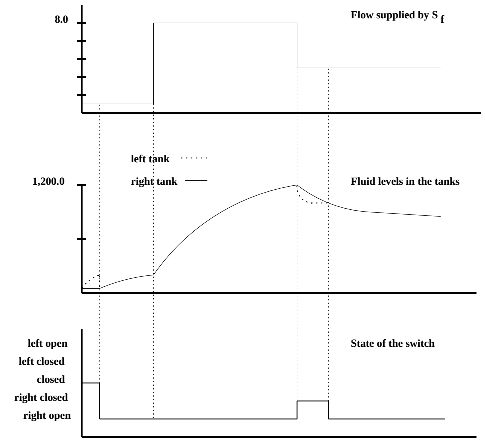

Figure 9: **Simulation Results**

tegration is implemented as a zero-order approximation and careful selection of the time step produced good results inspite of this simplifying approximation. The mode switch implementation sets the flow in the controlled junction to 0, which eliminates this variable from the set of equations in the active mode. The simulation uses a color code to depict the current state of the controlled junction (*green* implies the switch is *on* and *red* implies that it is *off*). Fig. 8 shows the latched bi-tank simulator window.

Simulation of the system using the parameters displayed in Fig. 8 produces the behavior description shown in Fig. 9. At the start, the latch is closed and there is no flow between the compartments. This represents the closed state in the CSPEC (Fig. 5). The resulting bond graph consists of two independent containers (Fig. 7). Both of the containers are connected to dissipating elements (valves) and the left container has an input flow source.

When the level of fluid in the left container becomes high enough, the threshold pressure (p_t) is exceeded and the latch opens to the right. The system moves into its *right open* state and the 1_1 junction now behaves as a regular 1-junction. Because of the low resistance connection (the parameters are defined in such a way that the time constant ($R \cdot C = 5 \times 10^{-7}$) of the flow through the valve is small compared to the time step chosen (10^{-4})), the levels of fluid in the two compartments will almost instantaneously move to the same value. Fig. 9 shows this initial exponential increase and sharp decrease of fluid-level of the left container as a little hub. The levels of fluid in both of the containers now move to an equilibrium steady-state. Being a first order system (the two energy storing elements are dependent), there is no overshoot at the equilibrium level and the latch does not oscillate.

Next, the flow supplied by the source S_f is increased to 10 times its initial value. The latch is already in its *right-open* state, therefore, the system exhibits a first order exponential behavior towards a new equilibrium level for both containers. Then the input flow is reduced to half its original value, and the level of fluid in the left compartment falls much faster than the right compartment because of its smaller time constant (10^{-3} for left compartment and 21×10^{-3} for the right compartment). As soon as the pressure in the right compartment exceeds that of the left, the valve closes in its *right-closed* mode, i.e., it cannot open to the left. The right compartment now achieves a new equilibrium level. The resulting bond graph is that of the two independent compartments, i.e., the initial state (Fig. 7). When the level of fluid in the right compartment falls below the level in the left compartment, the latch again opens up to the right. The levels of fluid in the left and right compartments then achieve a new equilibrium level together.

6 Conclusions

This paper has discussed a comprehensive modeling methodology for complex physical systems that exhibit multiple discontinuous behavior modes. Our primary contribution is the introduction of systematic and uniform mode-switching methods into a formal compositional modeling scheme based on the bond graph methodology. The goal, to provide a modeling language that ensures the development of consistent, rigorous and complete models, is achieved by combining: (i) the bond graph scheme to model the energy-related aspects of system behavior, and (ii) finite-state automata to model signal-flows that cause configuration changes in the bond graph model to produce discontinuous behavior. Interaction between the two components of the model are restricted to physical signals that act on so-called *controlled junctions*. Physical signals interacting at bond graph junctions are an integral part of the bond graph language.

The combined use of bond graphs and finite state automata was discussed first in [4]. In their method, global automata control sub-system switching and model interactions between signal and bond graph models through switching bonds, but this can cause hanging junctions. Controlled junctions represent an extended implementation of the *energetic switch* concept introduced by Strömberg, Top, and Söderman[17]. The extended implementation allows for more sophisticated *sequential* control logic as opposed to purely *combinational* logic. Its use has been demonstrated for the more complex bi-tank system. Also, the controlled switch is a transient or control element rather than an energy-related bond graph element.

The strict definition of the interaction between the energy-flow and signal-flow components of the modeling methodology is of paramount importance in generating valid physical models. The approach presented supports modeling discontinuities caused by (i) abrupt switching, such as in idealized valves and diodes, (ii) mode switching caused by parameter value changes, such as the change from laminar to turbulent flow in a pipe when the Reynolds number goes above a threshold value, and (iii) configuration switching caused by changes in subsystem models.

Furthermore, focusing on the energy model instead of external control models (as is done in QSIM and CC) allows for dynamic model composition. This is extremely important for complex systems that include a large number of discontinuous components. As discussed earlier, in general such systems can exhibit an exponential number of modes, therefore, pre-enumerating modes is not a feasible modeling approach. Moreover, a large number of these modes are not physically achievable, but that cannot always be determined before hand.

In future, this approach needs to be applied to comprehensive modeling and analysis of more complex systems. Typical examples would be chemical plants that involve phase changes and chemical reactions among materials, and complex electronic circuits that have a large number of switching elements and piecewise linear components. Our long-term goal is to employ these methodologies in building robust models for automated diagnosis and design of complex, engineering systems.

References

[1] G. Biswas, S. Manganaris and X. Yu, "Extending Component Connection Modeling for Analyzing Complex Physical Systems," *IEEE Expert*, vol. 8, no. 1, pp. 48-57, 1993.

[2] G. Biswas, and X. Yu, "A Formal Modeling Scheme for Continuous Systems: Focus on Diagnosis," *Proc. IJCAI-93*, Chambery, France, pp. 1474-1479, August 1993.

[3] J.F. Broenink, Computer Aided Physical Modeling and Simulation: Bond-Graph Approach, *Ph.D. thesis*, University of Twente, Febodruk, Enschede, The Netherlands, 1990.

[4] J.F. Broenink and K.C.J. Wijbrans, "Describing Discontinuities in Bond Graphs," *Proc. Intl. Conf. on Bond Graph Modeling*, pp. 120-125, San Diego, CA, 1993.

[5] B. Falkenhainer and K. Forbus, "Compositional Modeling: Finding the right model for the job," *AI Journal*, vol. 51, pp. 95-143, 1991.

[6] K.D. Forbus, "The Qualitative Process Engine," in *Readings in Qualitative Reasoning about Physical Systems,* D. Weld and J. deKleer, eds., pp. 220-235, Morgan Kaufmann, San Mateo, CA, 1990.

[7] D.J. Hatley, and I.A. Pirbhai, Strategies for Real-Time System Specification, Dorset House Publishing Co., New York, N.Y., 1988.

[8] D.C. Karnopp, "General method for including rapidly switching devices in dynamic simulation system models," *Trans. Society for Computer Simulation*, col. 2, pp. 155-168, 1985.

[9] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg, Systems Dynamics: A Unified Approach, 2nd edition, John Wiley, New York, 1990.

[10] B. Kuipers, "Qualitative Simulation," *Artificial Intelligence*, Ol. 29, pp. 289-338, 1986.

[11] Microsoft Corporation, Visual Basic, Version 3.0 Professional Edition, One Microsoft Way, Redmond, WA, 98052-6399, 1993.

[12] P. Nayak, L. Joscowicz and S. Addanki, "Automated Model Selection using Context-Dependent Behaviors," *Proc. AAAI-91*, AAAI/MIT Press, Menlo Park, CA, pp. 710-716, 1991.

[13] T. Nishida, and S. Doshita, "Reasoning about Discontinuous Change,", *Proc. AAAI-87*, pp. 643-648, Seattle, WA, 1987.

[14] H.M. Paynter, Analysis and Design of Engineering Systems, The M.I.T. Press, Cambridge, MA, 1961.

[15] R. Rosenberg, and D. Karnopp, Introduction to Physical System Dynamics, McGraw Hill, New York, NY, 1983.

[16] J.E. Strömberg, J. Top and U. Söderman, "Variable causality in bond graphs caused by discrete effects," *Proc. Intl. Conf. on Bond Graph Modeling*, pp. 115-119, San Diego, CA, 1993.

[17] J.E. Strömberg, U. Söderman and J. Top, "Bond graph supported estimation of discretely changing parameters," *Proc. Intl. Conf. on Bond Graph Modeling*, pp. 47-52, San Diego, CA, 1993.

[18] J.L. Top, J.M. Akkermans, and P.C. Breedveld, "Qualitative reasoning about physical systems: an artificial intelligence approach," *Journal of the Franklin Institute*, vol. 328, no. 5/6, pp. 1047-1065, 1991.

[19] J.L. Top, Conceptual Modeling of Physical Systems, *Ph.D. thesis,* University of Twente, CIP-DATA Koninklijke Bibliotheek, Den Haag, 1993.

[20] S.M. White and J.Z. Lavi, "Embedded Computer System Requirements Workshop," *Computer*, April 1985.