Efficient Simulation of Component-Based Hybrid Models Represented as Hybrid Bond Graphs

Matthew Daigle, Indranil Roychoudhury, Gautam Biswas, and Xenofon Koutsoukos

EECS Department/ISIS, Vanderbilt University
Nashville, TN 37235, USA
matthew.j.daigle@vanderbilt.edu

1 Introduction

Modern engineering systems consist of a large number of interacting components with nonlinear, hybrid behaviors. Building accurate and computationally efficient simulation models for these systems is a challenging task. Researchers have adopted component- [1] and actor-oriented [2] frameworks for modeling large hybrid systems. Mathematical models specify individual component behaviors and formal models of computation define component interactions in these frameworks, and they provide the basis for developing efficient schemes for simulating the hybrid system behavior.

In our work, we adopt the Hybrid Bond Graph (HBG) paradigm [3], an extension of the Bond Graph (BG) modeling language [4], for component-based modeling of embedded systems. HBGs are a domain-independent topological modeling language that capture interactions among the physical and logical processes that constitute a system. The parametric component-based modeling of hybrid systems and the inherent topological structure offer significant advantages for analyzing system behavior and model-based fault diagnosis [5].

In this paper, we address the challenge of translating HBG models to computationally efficient simulation models exploiting causal information that is derived from the topological structure. Mode changes in HBG models, represented as discrete switching events, cause dynamic changes in the topological structure, and, therefore, the computational model during execution. We develop efficient simulation algorithms by converting the HBG models to reconfigurable block diagram structures, using the *Hybrid Sequential Causal Assignment Procedure* to dynamically update the causal information. We demonstrate the technique by deriving the block diagram model of an electrical power system, and running simulation experiments in MATLAB® Simulink® [7].

2 Translating Hybrid Bond Graphs to Block Diagrams

BGs are domain-independent, topological, lumped-parameter models that capture the energy exchange mechanisms in physical processes [4]. The nodes of a bond graph model energy storage, dissipation, transformation, and input-output elements. Connections in the system are idealized, and modeled by two additional nodes: 0- (or parallel) and 1- (or series) junctions. The connecting edges, called *bonds*, define energy pathways between elements. Parameters of nonlinear BG elements are defined by algebraic *modulating functions*, whose parameters are system variables and external input signals [8]. HBGs introduce discrete configuration changes in continuous BG models by allowing junctions to be turned on and off [3]. A two state (*on* and *off*) finite state machine implements the junction *control specification* with the transition guards expressed as boolean functions of system variables and inputs. When a controlled junction is on, it behaves like a conventional junction. When off, all bonds incident on the junction are deactivated. The system mode at any time is determined by composing states of the individual switched junctions. Details of the language are presented in [3].

There are two primary challenges in deriving simulation models from HBGs. First is to *avoid pre-enumeration of model configurations*. A HBG model with m components, each with n_i switching junctions, defines $2^{\sum_{i=1}^{m} n_i}$ different system modes (or model configurations), where $i = 1, 2, \ldots m$. When large, it is infeasible to pre-enumerate all the model configurations. Therefore, model reconfiguration at mode changes must be executed at run-time. Second is to *avoid algebraic loops*. Component-based modeling of hybrid systems produces an underlying mathematical model, which is a set of differential-algebraic equations (DAEs) that may include algebraic loops. Generating fixed-point solutions for DAEs with algebraic loops becomes computationally expensive when the fixed-point method has to iterate to converge to a solution.

Causality Assignment The *Sequential Causal Assignment Procedure* (`SCAP`) [4] applied to well-formed BG models assigns causal directions to all bonds in the model. Causality defines the input-output relations between the associated effort and flow variables. This provides the basis for a graphical block diagram (BD) representation, which captures the complete computational model of the system (This is equivalent to the DAE model of the system). The causally derived BD model will also have the minimum number of algebraic loops [4].

Given causal assignments, there is a one-to-one mapping from the BG to the BD model. For HBGs, however, the causal assignments may change when junctions switch state. To avoid the costly pre-enumeration of system modes, we implement an efficient BD reconfiguration scheme that recomputes the causal assignments incrementally, starting from the junctions that switch state, and propagating causal assignment changes till a new consistent assignment is derived. Corresponding changes are made only to those blocks that have changes in the causal assignments of their incident bonds.

The *Hybrid Sequential Causal Assignment Procedure* (`Hybrid SCAP`) performs the causality assignment dynamically when mode changes occur in the system. We assume that the states of all junctions are available before `Hybrid SCAP` is applied. The algorithm starts with a queue of switched junctions. It picks one junction off the queue, makes all the forced causal assignments, and propagates effects of these assignments, making all the consequent forced changes till none remain. Junctions with incomplete causal assignments to their bonds are added to a second queue. When the first queue is empty, the algorithm picks elements off the second queue, makes a valid causal assignment to an unforced element, and propagates its effects to make any forced changes that result from the chosen assignment. This process continues until all bonds have been assigned causality. The propagation is local, so only a subset of the bonds change causal assignments. Details of the algorithm can be found in [6].

Implementation In MATLAB Simulink implementations, we have explored two approaches. *Implicit switching* uses conditional statements to model the variable input-output relations for block elements whose incident bond(s) can change causality. The switching of the data flow between blocks is, therefore, implicit in the model. The models generated are compact because mode descriptions are expressed concisely as code. However, this approach results in more algebraic loops in the Simulink model, because the input-output directional structure gets buried in the code. During simulation, Simulink invokes fixed point solvers, and the computational overhead affects the simulation efficiency.

Explicit switching uses switching elements to enumerate the data flow paths and the corresponding computational structure for each configuration. At run time, the appropriate switches are triggered to produce the changed block diagram structure. The models created by this approach have many more atomic blocks than the implicit models because multiple BD expansions are enumerated for each element. Since the data flow paths are made explicit for each configuration, no additional algebraic loops are created, however, the switching elements incur overhead associated with zero-crossing detection.

3 Case Study: Electrical Power System

We applied our modeling and simulation framework to the Advanced Diagnostics and Prognostics Testbed (ADAPT) system deployed at NASA Ames. The system consists of *power generation* (solar panel and battery chargers), *power storage* (three sets of lead-acid batteries), and *power distribution* (a number of DC to AC converters and AC and DC loads) subsystems. Relays are used to configure the system in different modes of operation, e.g., charge and/or discharge modes of the batteries, as well as different power supply and load configurations. Because of the large number of possible configurations involving different components, it is infeasible to pre-enumerate all possible modes of operation. We have developed HBG models for all of the components in the ADAPT testbed and used our approach to simulate the system in different configurations [6].

We present the simulation results for a battery supplying power to two DC loads in parallel, with relays that enable the loads to be switched on and off (see Fig. 1). The Simulink model was run for $7,000$ seconds of simulation time with the battery discharging through different load configurations. For this experiment, the explicit switching implementation executed about 20% faster than the implicit switching implementation. This difference in the simulation efficiency

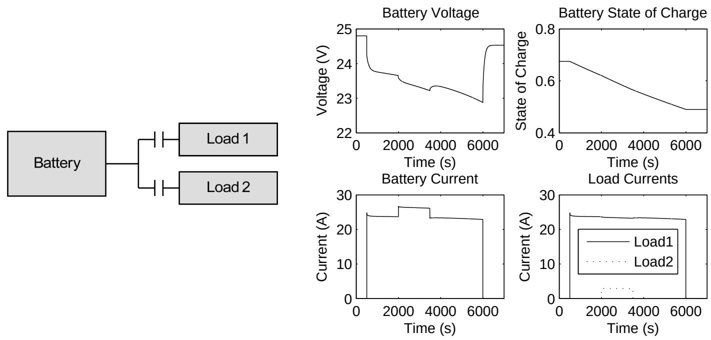

Fig. 1. Example model and simulation results

was consistent with other configurations of the system. In future work, we will formalize the computational modeling framework, and further study the computational efficiency for different systems.

Acknowledgments This work was supported in part by grants NSF CNS-0347440 and NSF CNS-0615214, and a NASA USRA grant from Ames Research Center. We gratefully acknowledge the help from Scott Poll and Ann Patterson-Hine in building the ADAPT system models.

References

1. Liu, J., Lee, E.A.: A component-based approach to modeling and simulating mixed-signal and hybrid systems. ACM Trans. Model. Comput. Simul. **12**(4) (2002) 343–368
2. Lee, E.A., Neuendorffer, S., Wirthlin, M.J.: Actor-oriented design of embedded hardware and software systems. Journal of Circuits, Systems, and Computers **12**(3) (2003) 231–260
3. Mosterman, P.J., Biswas, G.: A theory of discontinuities in physical system models. J Franklin Institute **335B**(3) (1998) 401–439
4. Karnopp, D.C., Margolis, D.L., Rosenberg, R.C.: Systems Dynamics: Modeling and Simulation of Mechatronic Systems. Third edn. John Wiley & Sons, Inc., New York (2000)
5. Narasimhan, S., Biswas, G.: Model-based Diagnosis of Hybrid systems. IEEE Transactions on Systems, Man and Cybernetics, Part A, to appear.
6. Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X.: Efficient simulation of component-based hybrid models represented as hybrid bond graphs. Technical Report ISIS-06-712, Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA (2006).
7. MATLAB/Simulink: (http://www.mathworks.com/products/simulink/)
8. Manders, E.J., Biswas, G., Mahadevan, N., Karsai, G.: Component-oriented modeling of hybrid dynamic systems using the Generic Modeling Environment. In: Proc of the 4th Workshop on Model-Based Development of Computer Based Systems, Potsdam, Germany, (2006).