EFFICIENT SIMULATION OF HYBRID SYSTEMS: AN APPLICATION TO ELECTRICAL POWER DISTRIBUTION SYSTEMS

Indranil Roychoudhury, Matthew Daigle, Gautam Biswas, and Xenofon Koutsoukos
Institute for Software Integrated Systems
Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, TN 37235, USA
Email: {indranil.roychoudhury, matthew.j.daigle, gautam.biswas, xenofon.koutsoukos}@vanderbilt.edu

KEYWORDS

Hybrid system simulation, hybrid bond graphs, component-based modeling, block diagrams

ABSTRACT

This paper presents an efficient simulation scheme for hybrid systems modeled as hybrid bond graphs (HBGs). Considerable computational savings are achieved when mode changes occur during simulation by identifying persistent causal assignments to bonds, and, consequently, fixed causal structures at HBG junctions when the simulation model is derived. Persistent causal assignments also reduce the possible computational structures across all mode changes, and this leads to an overall reduction in the complexity of the simulation models. We demonstrate the benefits of our approach for an electrical power distribution system that includes a fast switching inverter system.

INTRODUCTION

Accurate and efficient modeling and simulation approaches are essential for design, analysis, diagnosis, and prognosis of complex, embedded systems. To address these needs, we have developed component-oriented modeling techniques based on hybrid bond graphs (Manders et al., 2006), and a model-integrated design methodology for efficient simulation that facilitates diagnosis and prognosis experiments (Roychoudhury et al., 2007; Poll et al., 2007). The building of accurate and efficient simulation models for hybrid, nonlinear systems is not trivial, especially since the simulation must deal with the computational issues that arise from nonlinear behaviors, as well as, accommodate system reconfigurations that produce discrete behavior changes. For systems where the reconfigurations occur at high frequencies, such as modern electronics-based electrical power systems (Biel et al., 2004), it is especially important to maintain accuracy in the generated behaviors without sacrificing simulation efficiency.

Bond graphs (BGs) are well-suited for modeling electrical power systems. The BG modeling language allows for multi-domain, topological, lumped-parameter modeling of physical processes by capturing their energy exchange mechanisms (Karnopp et al., 2000). The continuous BG representation has been extended to model hybrid systems by several researchers (Buisson et al., 2002; Magos et al., 2003; van Dijk, 1994; Borutzky, 1995). Our approach, hybrid bond graphs (HBGs), introduces discrete mode changes into continuous BG models through idealized *switching junctions* (Mosterman and Biswas, 1998). A two state (*on* and *off*) finite state machine implements the junction *control specification* (CSPEC). Transitions between states may be functions of system variables and/or system inputs. When a switching junction is on, it behaves like a conventional junction. When off, all bonds incident on a 1-junction (0-junction) are deactivated by enforcing 0 flows (efforts) on all bonds incident on that junction (see Fig. 1). The system mode at any given time is determined by composing the modes of the individual switching junctions.

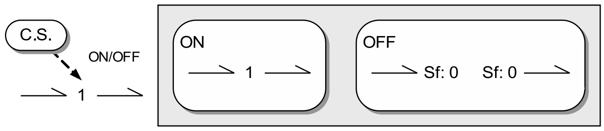

Figure 1: Semantics of a Switching 1-Junction

In earlier work, we produced an efficient simulation method for hybrid systems by constructing reconfigurable block diagram (BD) models from HBGs (Roychoudhury et al., 2007; Daigle et al., 2007). We adopted the BD formalism for simulating HBGs for three main reasons: *(i)* the BD formalism is a widely used computational scheme, *(ii)* the input-output formulation of each block in a BD can be determined using the causality information present in HBGs, and *(iii)* BD models preserve the component structure of the model, which facilitates introduction of faults into components for simulation-based diagnosis and prognosis studies. Our BD-based simulation models include switching elements that enable the online reconfiguration of the BD components to account for different causality assignments in different system modes. Every time a mode change occurs, causalities are incrementally reassigned from the previous mode, and the effort and flow links are rerouted by the switching elements to ensure that the computational model corresponds to the new causality assignment. This approach produces acceptable results when mode changes are infrequent. For fast switching systems

Proceedings 22nd European Conference on Modelling and Simulation ©ECMS Loucas S. Louca, Yiorgos Chrysanthou, Zuzana Oplatková, Khalid Al-Begain (Editors)
ISBN: 978-0-9553018-5-8 / ISBN: 978-0-9553018-6-5 (CD)

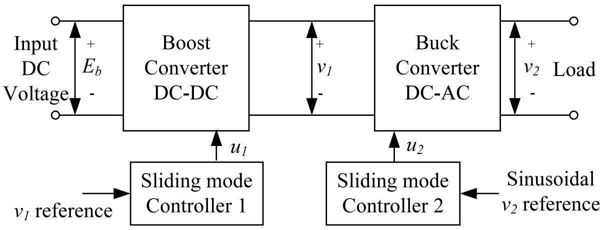

Figure 2: Block Diagram of a Boost-Buck AC Inverter

with frequent mode changes, e.g., electrical power conversion and distribution systems, invoking the procedure for reassigning causality at *every* mode change may produce unnecessary computational overhead, which leads to significant increases in the simulation execution times. Also, the BD models may include extraneous switching elements and signal connections, which account for causality assignments that never occur during the simulation.

This paper extends our earlier work by identifying bonds whose causal assignments persist across mode changes in the HBG model, and restrict possible reconfigurations that can occur in the simulation model when mode changes occur. Confining the effects of mode changes to small parts of the simulation model substantially reduces the computational effort required to execute mode changes during simulation. Further, persistent causality assignments limit the number of switches needed for each BD component, as well as, the number of possible signal connections. Therefore, simulation efficiency is increased, which produces significant gains for fast switching systems. We demonstrate the effectiveness of our approach by applying it to a power conversion and distribution testbed developed for diagnosis and prognosis studies at NASA Ames (Poll et al., 2007).

MOTIVATING EXAMPLE: AC INVERTER

The Advanced Diagnostics and Prognostics Testbed (ADAPT) models aircraft and spacecraft power distribution systems (Poll et al., 2007). It includes battery units for power storage, inverters for DC to AC power conversion, a power distribution network made up of a number of circuit breakers and relays, and DC and AC loads. In this paper, we focus on the AC subsystem, and develop the fast-switching inverter model to motivate our approach.

The inverter, a two-stage boost-buck DC-AC converter (Biel et al., 2004), consists of a cascade connection between a boost DC-DC converter with a full-bridge buck DC-AC converter to achieve a transformerless DC-AC step-up conversion (see Fig. 2). The boost converter boosts the input DC voltage to a higher value (190 V, in our case), and the buck converter stage generates the sinusoidal AC voltage. The fast switching in the boost and buck converters are governed by two sliding mode controllers, one for each stage of the inverter.

The equivalent circuit model of the boost-buck DC-

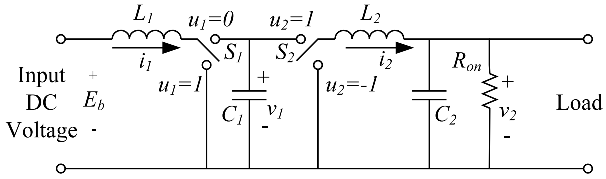

Figure 3: Circuit Model of a Boost-Buck AC Inverter

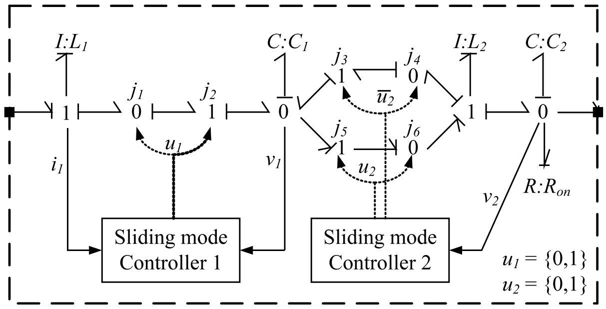

Figure 4: Inverter HBG Component Model

AC inverter is shown in Fig. 3, where S_1 is a conventional power switch, and S_2 corresponds to a full bridge switch. The control signals for S_1 and S_2 are u_1 and u_2, respectively. The differential equations for the system can be found in (Biel et al., 2004), and the model parameters are listed in Table 1. The internal resistance R_{on} accounts for the current that the inverter draws from the battery when it is disconnected from its loads.

The HBG model of the inverter, derived from its circuit model, is shown in Fig. 4. Switch S_1 is represented by the synchronous switching junctions j_1 and j_2, i.e., they share the same CSPEC function. Switch S_2, is represented by the switching junctions $j_3 - j_6$, with j_3 and j_5 having the same CSPEC as junctions j_4 and j_6, respectively. The switching conditions for junctions j_3 and j_4 are logical negations of those for junctions j_5 and j_6.

The sliding mode controllers are also modeled using HBGs. They generate signals that switch the inverter junctions at kilohertz rates. Our previous simulation algorithm invokes the `Hybrid SCAP` causal assignment procedure (Roychoudhury et al., 2007) at every mode change to compute the updated model configurations before the continuous simulation is resumed. However,

Table 1: Inverter Model Parameter Values

Inductances (H)	$L_1 = 0.0022$, $L_2 = 0.075$
Capacitances (F)	$C_1 = 0.0069$, $C_2 = 6 \times 10^{-6}$
Resitances (Ω)	$R_{on} = 489.49$
Sliding mode controller 1 parameters	$\alpha = 0.8$, $\beta = 4.3649$ $\delta = 111.375$, $K = 829.3347$
Sliding mode controller 2 parameters	$a_1 = 15.915$, $a_2 = 0.0048$
Reference Voltages (V)	$v_{1Ref} = 190$, $v_{2Ref} = 120\sqrt{2}\sin(120\pi)$

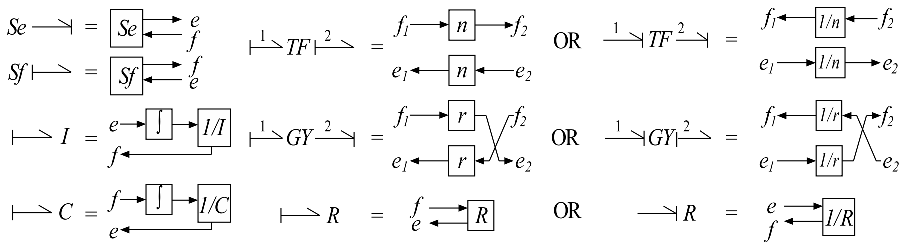

Figure 5: Computational Structures for 1- and 2-Port Bond Graph Elements

careful inspection of the inverter HBG model shows that the causality assignments at the switching junctions remain the same when the junctions are on. When the junctions are off, the causal changes do not propagate to adjacent junctions (see the next section for details). Therefore, the external calls to `Hybrid SCAP` at every mode change are not necessary, and considerably slows the simulation. If we can identify the causality assignments that do not change when reconfigurations occur, the number of calls to `Hybrid SCAP` can be reduced, and the model reconfiguration task can be simplified. This approach is likely to produce more efficient simulation models than our previous approach.

EFFICIENT SIMULATION OF HYBRID BOND GRAPH MODELS

Efficient simulation models for hybrid systems should meet two primary requirements: *(i)* avoid pre-enumeration of all system modes, especially for systems with a large number of modes, and *(ii)* minimize the amount of computations performed when mode changes occur. Reassignment of causality produces changes in the computational model. But, we can minimize the number of changes during reconfiguration by *(i)* recognizing causal assignments that persist across all modes, and *(ii)* not accounting for configurations that we can predetermine will never occur. As a result, we can reduce the extent of causal propagation changes, and simplify the simulation models.

Converting Bond Graphs to Block Diagrams

In our work, we assume that energy storage elements (i.e., C and I) are in *integral causality*. Fig. 5 shows the BD structure for 1- and 2-port BG elements (Karnopp et al., 2000). The Sf, Se, and 1-port C and I elements, and the n-port I and C fields, each have a unique BD representation because their incident bonds have only one possible causal assignment. The BDs for the n-port I and C fields are simple functional and structural extensions to the 1-port I and C elements, respectively, and hence not shown in Fig. 5. The R, TF and GY elements allow two causal assignments each, and each assignment produces a different BD model.

Mapping a junction structure to its BD model is facilitated by the commonly used notion of the *determining bond*, which captures the causal structure for the junction.

Definition 1 (Determining Bond) *The* determining bond *of a* 0- (1-) *junction is the bond that establishes the effort (flow) value of all other bonds incident on the junction.*

Fig. 6 shows the BD expansion for a 1- junction. For a 1-junction, all other bonds' flow values are equal to the determining bond's flow value, and the effort value of the determining bond is the algebraic sum of the effort values of the other bonds connected to this 1- junction. A junction with m incident bonds that does not switch can have m possible BD configurations.

There is a well-defined procedure for converting a causal BG structure to a BD model (Karnopp et al., 2000). First, each bond is replaced by two signals, i.e., the effort and flow variables for the bond. Next, each bond graph element is replaced by the computational structure corresponding to the assigned causality, and the signals connections are established to complete the model.

Converting Hybrid Bond Graphs to Block Diagrams

As is shown in Fig. 6, the reassignment of causality due to mode changes can alter a junction's determining bond, and, therefore, its underlying computational BD model. However, instead of rebuilding the entire BD model every time mode changes occur, we include switching elements in the BD components to reconfigure the computational model during simulation. For example, a switching junction with m incident bonds can switch between $m + 1$ possible computational configurations, m corresponding to each incident bond being a determining bond, and one corresponding to the junction being off, in which each outgoing signal is set to zero. Therefore, the physical connections between blocks are fixed, but the interpretation of the signal on the connection (effort or flow) changes depending on the causal assignment to the bonds.

In some cases, however, the causal assignment for a bond is invariant across *all* possible modes of system behavior, i.e., the causal assignment is *persistent*. For example, a C-element will always impose effort on a 1-junction through the connecting bond. In this case, the BD for the 1-junction does not need to include any

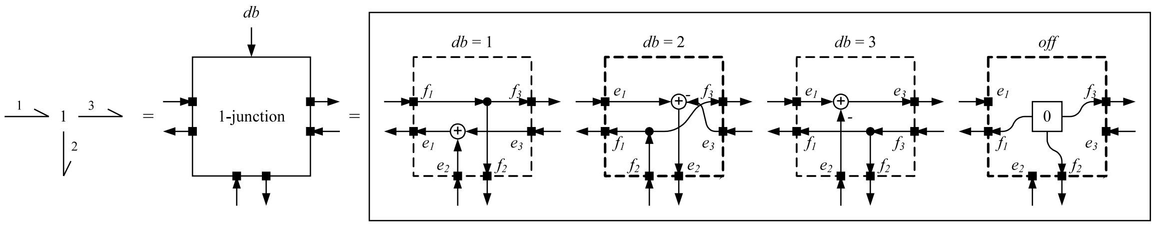

Figure 6: Block Diagram Expansion of a 1-Junction

switching mechanism to accommodate the possibility of this bond being its determining bond. If all bonds of a junction have persistent causal assignments, then its determining bond is fixed for all modes of the system in which it is on. In this case, the BD for the 0-junction does not need to include any switching mechanisms because it can assume a fixed structure when the junction is on. In previous work, we have termed a nonswitching junction with this property to have *fixed causality* (Roychoudhury et al., 2007). Switching junctions, by definition, change causality when they turn off, but this change may not affect adjacent junctions. Therefore, we extend our previous definition of fixed causality to also include switching junctions.

Definition 2 (Fixed Causality) *A junction that does not switch is in* fixed causality *if, for* all *modes of system operation, its determining bond does not change. A switching junction is in* fixed causality *if, for* all *modes in which the junction is on, its determining bond does not change, and for all modes where it is off, the inactivation of its incident bonds does not affect the determining bond of any of its adjacent junctions.*

Persistent causality of bonds, and fixed causality of junctions can be identified efficiently using a `SCAP`-like algorithm before we construct the BD model for simulation from the HBG. In this algorithm, the causality assignment is first performed at junctions connected to sources and energy storage elements, because the bonds connecting them to these junctions have persistent causality. A 0-(1-) junction is in fixed causality if it is connected to a Se (Sf) or a C (I) element. Otherwise, a junction is in fixed causality if *(i)* its determining bond connects it to a fixed causality junction (either directly, or through a TF or a GY element), or *(ii)* all incident bonds other than its determining bond are connected to fixed causality junctions. Once a junction is determined to be in fixed causality, we propagate this information to its adjacent junctions to check if they too are in fixed causality, or any of their bonds are in persistent causality.

Some additional analysis is required to determine whether a switching junction is in fixed causality. When the junction is on, all its bonds must have persistent causality. When the junction turns off the determining bond for its adjacent junctions should not change, i.e., none of its incident bonds can be a determining bond for its adjacent junctions. A special case occurs if two switching junctions are adjacent and share the same CSPEC. The knowledge that they switch together can help determine if causal changes will propagate when they switch. When we visit a junction for the first time, all its adjacent junctions may not have been checked for fixed causality yet. Hence, the causality may need to be propagated from all adjacent junctions before it can be determined that the junction is in fixed causality. The computational complexity of our approach to identify persistent causality of bonds and fixed causality of junctions, is polynomial in the size of the HBG, as it is similar to the `SCAP` algorithm.

If a junction has incident bonds with persistent causality, or if the junction is in fixed causality, the computational model of the junction block can be reduced by eliminating switching elements and signal connections which account for causality assignments that will never occur during simulation. This is illustrated for a 3-port switching 1-junction in Fig. 6. If the junction is not in fixed causality, its implementation can, in general, switch between four possible configurations as mode changes occur. However, if the junction is in fixed causality, the BD representation for this junction has only one valid on configuration, in addition to the off configuration. If the junction is not in fixed causality, for example, if its bond 1 connected to a Se, its BD representation need not include a configuration with bond 1 as its determining bond. Switched junctions in fixed causality help minimize causality reassignment computations when mode changes occur. Therefore, when this 1-junction switches, we know exactly what the causality assignment at this junction is without having to call any external causality reassignment procedure.

When switching junctions not in fixed causality change modes, we have to make external calls to a causality reassignment procedure. In previous work, we have developed the `Hybrid SCAP` algorithm that reassigned causality incrementally, starting from the junction directly affected by the switching, and then propagating the changes only to those junctions whose causal assignments were affected by changes in the adjacent junctions (Daigle et al., 2007; Roychoudhury et al., 2007). In this work, we use the knowledge of junctions in fixed causality, and bonds with persistent causality, to reduce the search and propagation for the `Hybrid SCAP` algorithm. Causal changes are not propagated along bonds with persistent causality, or to junction in fixed causality.

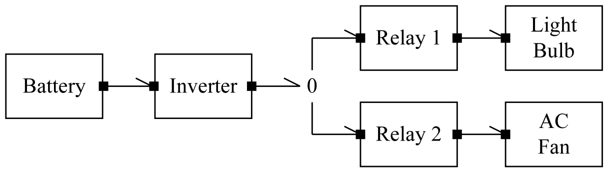

Figure 7: ADAPT Subsystem for Case Study

Consider the inverter HBG model (Fig. 4), where all nonswitching junctions are in fixed causality. The incident energy storage elements specify a unique determining bond for these junctions. All switching junctions are also fixed. Consider switching junctions j_1 and j_2. Since they always change modes simultaneously (because they share the same CSPEC), when on, j_1 always imposes flow on its adjacent junction j_2 which is also on. When they are off, the causality assignment of other active junctions are not affected. The case is similar for pairs j_3 and j_4, and j_5 and j_6. Since all junctions are in fixed causality, the mode switchings in the inverter do not require reassignment of causality, because the changes never propagate. Therefore, `Hybrid SCAP` is not invoked, and minimal changes have to be made to the computational model when mode changes occur, thus considerably speeding up the inverter simulation, as we illustrate later.

Simulating the Block Diagrams

Once a reconfigurable BD is generated for a new mode, using the extensions described above for a system model, the execution engine executes the BD in a continuous manner until the next mode change occurs. If the mode change is attributed to the switching of junctions that are not in fixed causality, the simulation is paused, the modified `Hybrid SCAP` algorithm is invoked to reassign causality, and the BD is reconfigured accordingly before the continuous simulation resumes. On the other hand, if a mode change is attributed to a switching junction with fixed causality, the BD reconfiguration is performed without invoking `Hybrid SCAP`.

CASE STUDY

We demonstrate our modeling and simulation framework using the ADAPT system at NASA Ames (Poll et al., 2007). In this paper, we focus on a subsystem of ADAPT shown in Fig. 7, i.e., a battery driving an inverter that is connected to two loads through two relays. One of these loads is a light bulb, while the other load is a large fan.

We build the system model using a component-based modeling approach (Manders et al., 2006; Poll et al., 2007). We use the HBG model of the inverter shown in Fig. 4 for our experiments. The HBG model of the lead-acid battery is based on an electrical equivalent circuit model, which captures the nonlinear battery behavior (Daigle et al., 2007). We omit a detailed description of the battery model due to space constraints. The resistive light bulb load consists of a single power port, a 1-junction, and a R element. The R element parameter value is a constant $234.6966\ \Omega$.

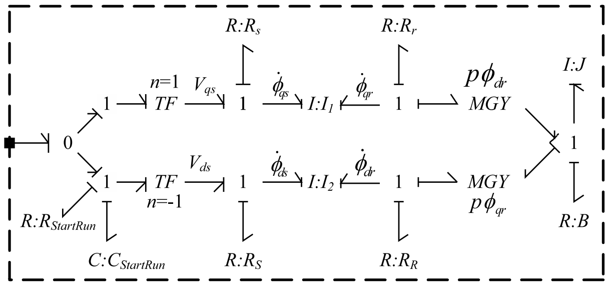

Figure 8: AC Fan Hybrid Bond Graph

Table 2: AC Fan Model Parameter Values

Inductances (in H)	$L_{ss} = 0.275$, $L_{SS} = 0.274$, $L'_{rr} = 0.272$, $L'_{RR} = 0.271$, $L_{ms} = 0.1772$, $L_{mS} = 0.2467$
Inertia (in kg m^2)	$J = 6.5 \times 10^{-4}$
Resistances (in Ω)	$R_s = 163.02$, $R_S = 168.14$, $R'_r = 145.12$, $R'_R = 145.12$, $R_{startRun} = 26$,
Friction (in kg m^2/s)	$B = 4.734 \times 10^{-4}$
Capacitances (in μF)	$C_{startRun} = 21.1$
Other Parameters	$N_S/N_s = 1.18, p = 2$

The AC fan is modeled as a single phase, fixed capacitor induction motor (Thaler and Wilcox, 1966). We represent the system using the standard $d - q$ model, described in (Krause, 1986). We do not present the fan model equations due to space constraints. The model parameters are presented in Table 2. The HBG model of the fan, shown in Fig. 8, is adapted from that described in (Karnopp, 2003). Note that the AC fan has two I-fields with parameters,

$$I_1 = \begin{bmatrix} L_{ss} & L_{ms} \\ L_{ms} & L'_{rr} \end{bmatrix}, \text{ and } I_2 = \begin{bmatrix} L_{SS} & L_{ms} \\ L_{ms} & L'_{RR} \end{bmatrix}.$$

Experimental Results

For the experimental results, we automatically derived a Matlab® Simulink® model of the subsystem shown in Fig. 7, from the HBG model of the system using the implementation described in (Roychoudhury et al., 2007) along with the extensions presented in this paper. All experiments were performed on a 2.4 GHz Intel® Pentium Core™2 Duo CPU desktop, with 2 GB RAM. The model was simulated using a fixed-step simulation with a sample period of 7.5 μs.

Fig. 9 shows the results of the simulation. We plot the voltages and currents at the output of the battery and the inverter, as well as the rotational speed of the AC fan. The simulation model was run for 20 seconds of simulation time. In the first configuration, the light bulb is connected to the inverter from $2 - 5$ seconds, followed

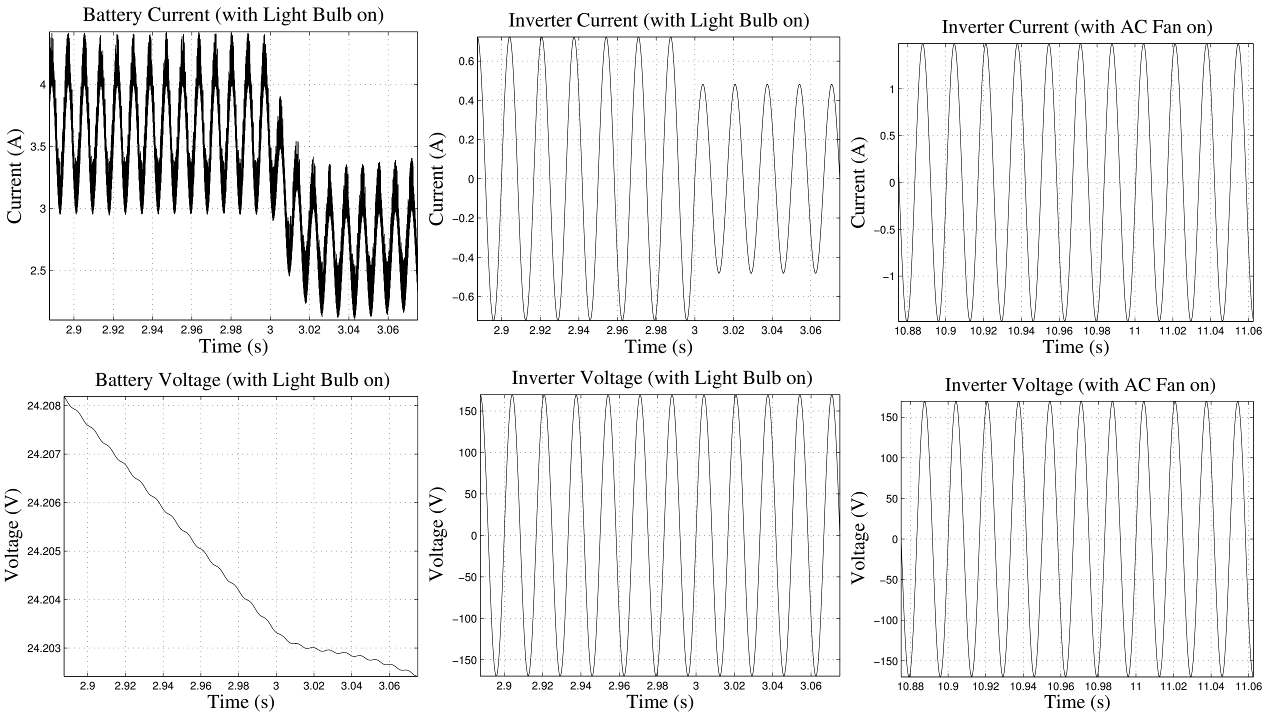

Figure 9: Simulation Results

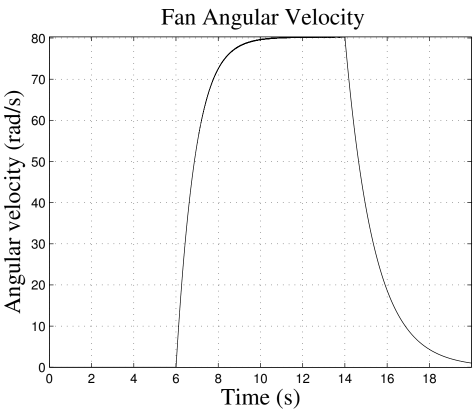

Figure 10: Angular Velocity of AC Fan

Table 3: Duration (in s) for Simulating Model for 1s

`SCAP` called at every mode change	`Hybrid SCAP` called at every mode change	No causality reassignment procedure called
6054.3	6025.2	58.3

by a second configuration where the AC fan is turned on between $7 - 15$ seconds. An abrupt fault is injected in the light bulb resistance at 3 seconds to demonstrate the usefulness of the simulation approach for diagnosis applications. As we can see, the sliding mode controllers are robust to load changes, and generates true 120 V rms voltage for both the loads. However, the light bulb fault affects the inverter current, and, therefore, the battery current and voltage. The AC fan current shows a phase difference of 0.1346 rad. As can be seen in Fig. 10, when the AC fan is switched on, its speed of rotation increases until it reaches a steady state of about 78.5 rad/s. On turning off, the speed falls to zero.

Table 3 presents the result of an experiment to illustrate the efficiency gained by simplifying a reconfigurable BD model by identifying bonds with persistent causal assignments and junctions in fixed causality, and avoiding the need for causal reassignment for these modes. For this experiment, we assume that the AC fan is the only load and is on for the duration of the experiment. Each column in Table 3 reports the real time taken to simulate 1 second of simulation time for different HBG simulation runs. In all runs, all junctions of the HBG model are in fixed causality. In the first run, we call `SCAP` every time an inverter mode change occurs. Next, we repeat the previous run, using `Hybrid SCAP`. Finally, in the third run, we simulate the HBG without requiring any external calls to `Hybrid SCAP`, since all switching junctions are fixed. As can be seen from Table 3, our enhanced simulation approach, implemented in the third run, is 103.85 times faster than the first run, and 103.35 times faster than the second run. Our simulation approach also resulted in considerable improvements in the efficiency of simulation of a number of other configurations, especially for large systems like the VIRTUAL ADAPT simulation testbed (Poll et al., 2007). Further increase in simulation efficiency can be obtained by running our simulation models in the Rapid Accelerator mode of Simulink.

CONCLUSIONS

In this paper, we have presented an improved framework for simulating hybrid systems. The crux of these improvements is the identification of persistent causality of bonds, which not only avoids unnecessary invocations of the external `Hybrid SCAP` algorithm, thereby gaining increase in simulation efficiency, but also improves the efficiency of the `Hybrid SCAP` algorithm, as well as, enables the simplification of the simulation models by removing parts that correspond to configurations that never occur during the simulation. In the future, we will extend our modeling approach and computational model generation schemes to handle situations of derivative causality, and systematically evaluate how our approach performs when applied to other real-world large hybrid systems.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Grants CNS-0615214 and CNS-0347440, and NASA NRA NNX07AD12A.

REFERENCES

Biel, D., Guinjoan, F., Fossas, E., and Chavarria, J. (2004). Sliding-mode control design of a boost-buck switching converter for ac signal generation. *IEEE Transaction on Circuits and Systems - I*, 51(8):1539–1551.

Borutzky, W. (1995). Discontinuities in a bond graph framework. *Journal of the Franklin Institute*, 332(2):141–154.

Buisson, J., Cormerais, H., and Richard, P.-Y. (2002). Analysis of the bond graph model of hybrid physical systems with ideal switches. *Proc Instn Mech Engrs Vol 216 Part I: J Systems and Control Engineering*, pages 47–63.

Daigle, M., Roychoudhury, I., Biswas, G., and Koutsoukos, X. (2007). Efficient simulation of component-based hybrid models represented as hybrid bond graphs. In Bemporad, A., Bicchi, A., and Butazzo, G., editors, *HSCC 2007*, volume 4416 of *LNCS*, pages 680–683. Springer-Verlag.

Karnopp, D. (2003). Understanding induction motor state equations using bond graphs. In *Proceedings of the International Conference on Bond Graph Modeling and Simulation*, volume 35, pages 269 – 273.

Karnopp, D. C., Margolis, D. L., and Rosenberg, R. C. (2000). *Systems Dynamics: Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons, Inc., New York, third edition.

Krause, P. C. (1986). *Analysis of Electric Machinery*. John Wiley & Sons, Inc.

Magos, M., Valentin, C., and Maschke, B. (2003). Physical switching systems: From a network graph to a hybrid port hamiltonian formulation. In *Proc IFAC conf Analysis and Design of Hybrid Systems*, Saint Malo, France.

Manders, E.-J., Biswas, G., Mahadevan, N., and Karsai, G. (2006). 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. IEEE CS Press.

Mosterman, P. J. and Biswas, G. (1998). A theory of discontinuities in physical system models. *Journal of the Franklin Institute*, 335B(3):401–439.

Poll, S., Patterson-Hine, A., Camisa, J., Nishikawa, D., Spirkovska, L., Garcia, D., Hall, D., Neukom, C., Sweet, A., Yentus, S., Lee, C., Ossenfort, J., Roychoudhury, I., Daigle, M., Biswas, G., Koutsoukos, X., and Lutz, R. (2007). Evaluation, selection, and application of model-based diagnosis tools and approaches. In *AIAA Infotech@Aerospace 2007 Conference and Exhibit*.

Roychoudhury, I., Daigle, M., Biswas, G., Koutsoukos, X., and Mosterman, P. J. (2007). A method for efficient simulation of hybrid bond graphs. In *Proceedings of the International Conference of Bond Graph Modeling*, pages 177 – 184, San Diego, California.

Thaler, G. J. and Wilcox, M. L. (1966). *Electric Machines: Dynamics and Steady State*. John Wiley & Sons, Inc.

van Dijk, J. (1994). *On the role of bond graph causality in modeling mechatronics systems*. PhD thesis, University of Twente, Enschede, The Netherlands.

AUTHOR BIOGRAPHIES

INDRANIL ROYCHOUDHURY received the M.S. degree in computer science from Vanderbilt University, Nashville, TN, in 2006, where he is currently a Ph.D. Candidate. His web page is http://people.vanderbilt.edu/˜indranil.roychoudhury/.

MATTHEW DAIGLE received the Ph.D. degree in computer science from Vanderbilt University, Nashville, TN, in 2008. His web page can be found at http://people.vanderbilt.edu/˜matthew.j.daigle/.

GAUTAM BISWAS received the Ph.D. degree in computer science from Michigan State University, East Lansing. He is a Professor of Computer Science and Computer Engineering in the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. His web page can be found at http://www.vuse.vanderbilt.edu/˜biswas/.

XENOFON KOUTSOUKOS received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 2000. He is an Assistant Professor in the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. His web page is http://www.vuse.vanderbilt.edu/˜koutsoxd/.