ISSN: 2249-7196
IJMRR/ Oct 2020/ Volume 10/Issue 4/-/1-06
James Nutaro / International Journal of Management Research & Review

Hybrid Simulation of Electrical and Telecommunications Networks1

James Nutaro ∗, Phani Teja Kuruganti ∗, Laurie Miller †, Sara Mullen †, and Mallikarjun Shankar ∗
∗Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831
Email: {nutarojj,kurugantipv,shankarm}@ornl.gov
†Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455 Email: {mill0660,mull0197}@umn.edu

***Abstract*—** The modern power grid is strongly integrated with its communication network. While a power system primarily consists of elements that are modeled by continuous equations, a communication system has discrete event dynamics. We model the integrated operation of these two systems with a hybrid modeling and simulation technique. Systematically combining continuous and discrete event system models is necessary for correctly simulating critical system behaviors. This paper discusses an approach based on the Discrete Event System Specification (DEVS) that characterizes the interaction of the two systems formally to preserve simulation correctness. We demonstrate the implementation of our integrated hybrid simulation technique with detailed generator and network models in a wide-area cooperative automatic load-control scenario.

INTRODUCTION

Modeling and simulation of hybrid systems is a key enabler for next generation systems that will integrate packet-based communication networks and advanced control technologies [1–6]. Modern communication technology has a prominent role in current efforts to improve the reliability and efficiency of the North American electric power grid. Because communi- cation networks have a central place in the future power grid, accurate predictions of power grid behavior will be inseparablefrom accurate models of the communication infrastructure. Modeling and simulation tools for future power systems need to represent, in a holistic fashion, electric power networks, digital communication networks, and their interactions.

A model that integrates generators, loads, transmission lines,control processes, and packet switching data networks is an example of a hybrid system. Physical laws that dictate the behavior of electromechanical subsystems are described by differential-algebraic equations, with some discrete dynamics that result from circuit breakers, relays, and other types of local response mechanisms. Packet switching networks are discrete

[1]Research sponsored by the Laboratory Directed Research and Develop- ment Program of Oak Ridge National Laboratory (ORNL), managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. The submitted manuscript has been authored by a contractor of the U.S. Government under Contract DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free licenseto publish or reproduce the published form of this contribution, or allow othersto do so, for U.S. Government purposes.

event systems whose dynamic behavior is described by chains of significant events. Significant events in these networks include the expiration of timers, sending and receiving packets,packet buffer overflows, and packet collisions.

There is little published work that describes simulation tech- niques for integrating models of power systems and network- based wide area control schemes. A recently developed toolfor integrated simulation of power and communications isdescribed in [7]. The approach taken there is directed by the need to integrate closed software packages. Though a workingsimulation environment was constructed and demonstrated, the interaction between communications and power system models is restricted to fixed synchronization intervals. This could leadto simulation induced behaviors that do not originate from the actual model description.

This paper describes a technique for integrating continuous models of power system dynamics and discrete event models of digital communication networks. The integration technique is given formal semantics in terms of the Discrete Event Sys- tem Specification [8] (equivalent semantics in terms of Hybrid Input/Output Automata are presented in [9]). These formal semantics provide a rigorous mathematical basis for building and simulating hybrid system models. We have implemented the technique using the NS2 [10] and *adevs* [11] simulation tools. We demonstrate the integrated simulation capability in a wide area, cooperative, automatic load control scenario.

The outline for the rest of this paper is as follows. SectionII presents a mathematical structure for modeling continuous systems with discrete event external behaviors. In section III, we describe an NS2/*adevs* implementation of the hybrid simulation scheme. Section IV gives a detailed description of the power system model and discrete event control scheme. We describe experiments with this model in section V.

HYBRID PROCESSES

We specify a hybrid system with discrete external dynamicsusing the following structure;

X and Y, the input and output value sets

S, the internal state set

$F : S \times \mathbb{R} \rightarrow S$, the evolution function (1)

$G : S \rightarrow \mathbb{R}$, the event scheduling function

$A : S \times X_\Phi \rightarrow S$, the discrete action function, where $X_\Phi = X \cup \{\Phi\}$ and Φ is the non-event, and$L : S \rightarrow Y$, the discrete output function.

The set X is the range of values that can be injected intothe system, Y is the range of values that can be produced by the system, and S is the range of the system's internal state variables. The function F describes how the continuous, internal variables evolve between discrete events. The function$G$ indicates how much time will elapse before the next discrete internal event. The function A describes discrete state change in response to internal (state) and external (input) events. The function L determines how the system output changes in conjunction with discrete internal events.

Specifically, the evolution function $F(q, h)$, with $q \in S$ and$h \in \mathbb{R}$, takes the system from a state q at time t to a laterstate q' at time $t + h$. This function describes continuous, autonomous evolution of the model's internal state. The system evolves continuously until $G(q) = 0$ or a change occurs in the input trajectory. At these points, the discrete action func- tion dictates an immediate and, possibly, discontinuous state change.

The discrete action function $A(q, u)$, with $u \in X_\Phi$, deter- mines the response of the model to discrete events. These events can be inputs to the system or be triggered by its internal dynamics. In either case, the system state changes instantaneously from q to $q' = A(q, u)$. Changes due to inter- nal dynamics occur when $G(q) = 0$, and the subsequent state of the system is determined by $A(q, \Phi)$. External events are due to a change in the input trajectory. In this case, the state immediately following the event is given by $A(q, x)$, where$x \in X$ is the value of the input trajectory immediately after the change occurs.

The discrete output function $L(q)$ defines the model's outputtrajectory. The initial output value is given by $L(q_0)$, where q_0 is the initial state of the system. Discrete changes in the output trajectory are concurrent with internal events, i.e., a change in the system output takes place when $G(q) = 0$. Theoutput trajectory takes the value $L(q)$, and it keeps this valueuntil the system again enters a state in which G evaluates to zero.

A DEVS based formalization of the dynamics associated with the structure (1) can be constructed as follows. The internally continuous, externally discrete model is described as an atomic model with state set S. Its input and outputsets are X and Y. The state transition, time advance, and output function of the atomic model are defined in terms of the evolution function F, event scheduling function G, discrete output function L, and discrete action function A. These definitions are

$$\begin{aligned} \delta_{int}(q) &= A(F(q, ta(q)), \Phi) \\ \delta_{ext}(q, e, x) &= A(F(q, e), x) \\ \delta_{con}(q, x) &= A(F(q, ta(q)), x) \qquad (2) \\ ta(q) &= G(q) \\ \lambda(q) &= L(F(q, ta(q))) \end{aligned}$$

In this representation, output and internal events coincide with the expiration of the time advance. The discrete output is computed using the system state just prior to the discrete event(i.e., prior to applying the discrete action function).

An implementation of this system will, in general, require events that do not result in discrete actions (i.e., an evaluation of A) or discrete output (i.e., an evaluation of L). These types of events are needed, for instance, when the evolution and event scheduling functions are implemented with numerical integration and state event detection algorithms (see, e.g., [12]). A DEVS model that is functionally equivalent to (2) can be had by defining a function

$$IntegStep : S \rightarrow \mathbb{R}$$

that picks the next integration step size. The system dynamics are then defined by

$$\delta_{int}(q) = \begin{cases} A(F(q, ta(q)), \Phi) & \text{if } G(q) \leq IntegStep(q) \\ F(q, ta(q)) & \text{otherwise} \end{cases}$$

$$\delta_{ext}(q, e, x) = A(F(q, e), x)$$

$$\delta_{con}(q, x) = A(F(q, ta(q)), x) \qquad (3)$$

$$ta(q) = \min\{G(q), IntegStep(q)\} \quad \lambda(q) = \begin{cases} L(F(q, ta(q))) & \text{if } G(q) \leq IntegStep(q) \\ \Phi & \text{otherwise} \end{cases}$$

In the integrated model, the evolution function F cap- tures the continuous electromechanical behavior of the power system. The discrete action function A describes over-speed breakers and discrete load changes. This model interfaces with discrete event processes that describe the control scheme and communication system. We describe the implementation of this interface in section III.

I. HYBRID SYSTEM SIMULATION WITH *adevs* AND NS2 A hybrid simulation tool aimed at network-based systems was developed using the NS2 [10] and *adevs* [11] simulation packages. Discrete event and continuous sub-processes that do not model communications are implemented using *adevs*. Communications specific processes are modeled using NS2. Models developed with *adevs* are integrated into the NS2simulation model using the simulation control API that ispart of the *adevs* simulation software. This API provides a

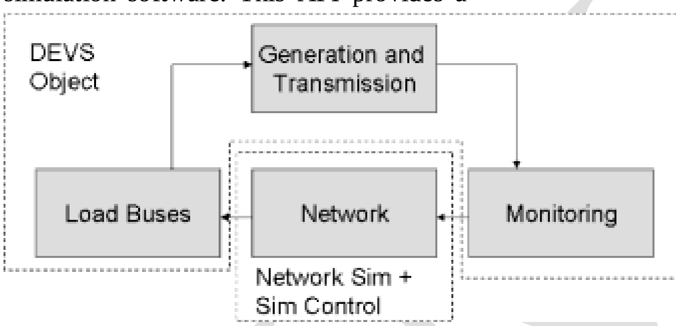

Fig. 1. Simulation Architecture

collection of functions that allow the underlying model to be manipulated as a single discrete event process (specifically, the resultant of the DEVS model [8]).

The *adevs* simulation software is encapsulated in an NS2 TclObject, and this TclObject is used directly by the NS2 simulation. The separation of the model into *adevs* and NS2 components is shown in Fig. 1. The NS2 module invokes the *adevs* components when two types of events occur;

1) an internal *adevs* event
2) receipt of a message at a process modeled within *adevs*.

Type 1 events are scheduled by NS2 on behalf of the *adevs* simulator. One event of this type is scheduled at the beginning of the simulation. When a Type 1 event occurs, the NS2simulator performs the following actions:

1) Query the *adevs* module for any network messages thatneed to be sent
2) Schedule NS2 events to send messages returned by theabove query
3) Tell the *adevs* component to update its internal state
4) Query the *adevs* component for its next event time andschedule a corresponding *adevs* event in NS2

Type 2 events cause the NS2 simulator to inject input into the *adevs* simulator in three steps;

1) Inject input events into the *adevs* component
2) Tell the *adevs* component to update its internal stateusing the injected events

3) Query the *adevs* component for its next event time andschedule a corresponding *adevs* event in NS2

The DEVS structure (3) is used to simulate continuousprocesses. A fourth order Runge-Kutta integration scheme isused to simulate an ODE set whose solution is the evolutionfunction F. Zero crossing functions are used to describe thelocation of state events. Time events are explicitly scheduled.State events are detected using discontinuity locking in conjunction with interval bisection. With this approach, thecontinuous variables are integrated over a single integrationstep. If the sign of a zero crossing function is unchanged atthe end of the step, then it is assumed that no state eventoccurred in the interval. Otherwise, the integration step size ishalved, and the test is repeated. If this procedure causes the step size to reach some small threshold value, then the eventis assumed to occur at the end of that minimal step size.

II. Automatic Under-frequency Load Shedding

We consider a system for automatic load control to demon- strate the capabilities of the hybrid simulation framework. The continuous component in this system is a power generation and transmission model of a 17 bus system that is based on the IEEE 14 bus system. To accommodate the power flow used in this model, three additional buses were added to the IEEE 14 bus model to separate buses with both loadand generation. The model consists of twelve loads and five generators that are interconnected as shown in Fig. 2. Since frequency disturbances rather than voltage disturbances are the focus of the current investigation, only real power flow is calculated in the transmission network [13].

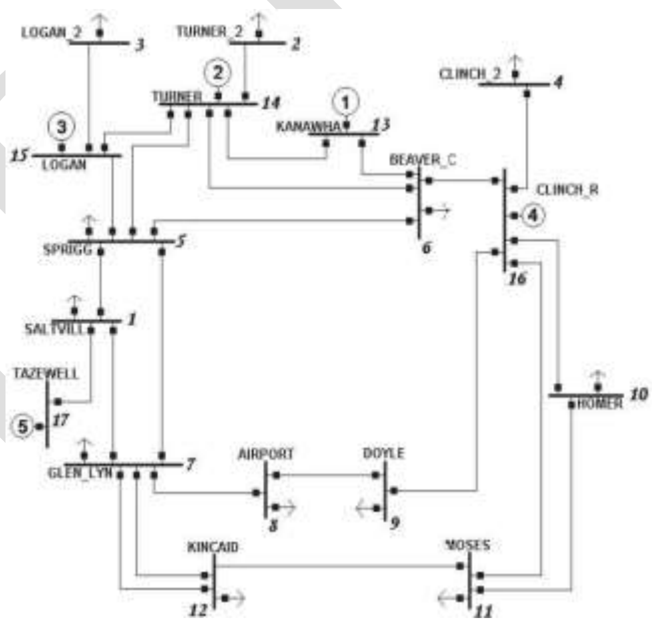

Fig. 2. The 5 generator and 12 load bus power system model.

The generators are modeled as synchronous machines using the swing equation plus additional equations that model a governor, non-reheat turbine, and over speed breaker. One of the five generators also includes a basic Automatic Generation Control (AGC) unit that eliminates steady-state frequency error throughout the system so long as no islanding is taking place. The three equations that describe the generator dynam- ics are

$$\Delta\dot{\delta}_g = \Delta\omega$$

$$\Delta\dot{\omega} = (\Delta P_m - \Delta P_e)/M$$

$$\Delta\ddot{P}_m = -100(k_{agc} b \Delta\delta_g + \Delta\omega/R + 0.25\Delta\dot{P}_m + P_m)$$

where ΔP_m and ΔP_e are the deviations of mechanical power output and electrical demand from the initial steady-state operating point P_0, $\Delta\delta_g$ is the change in relative generator shaft angle, $\Delta\omega$ is the deviation of the shaft angular velocity

from 60Hz, and b is the island status. If the system has split into islands, the island status is 0, and the AGC is effectively turned off [13]. The values R, M , and k_{agc} are the generator's speed droop constant, rotational inertia, and the AGC gain. If the speed deviation of a machine exceeds ± 0.1 Hz, then it is disconnected from the transmission network.

A DC power flow is used to calculate real power flow usingknown generator shaft angles and electrical power demand at the load buses. Disconnected generators are treated as load buses with zero power demand [13]. To facilitate the calculation of electrical demand P_e on the generators, the network admittance matrix Y is broken into the four sub- blocks terminal reduced to a fraction α of the full power demand. Changes to α occur in discrete increments $\Delta\alpha$ at a maximum rate of $1/T$.

The operation of the monitor is depicted graphically in Fig. 3. Circles denote discrete phases, with the appropriate state variable modification shown below the phase name. Arrows denote transition conditions and the associated output in the form *condition* / *output*. A new monitor state and output is calculated every T seconds using the sampled generator variable values. indicating that electrical power demand can be increased.

Load change requests are summarized by the load service fraction α, with $\alpha \in [0, 1]$. When $\alpha = 1$, the generator can tolerate the full electrical demand seen at its terminal. When $\alpha < 1$, the generator would like to see the demand on its
If the α_i are 1, then all of the demand at the load bus is met (i.e., $L_s = L_d$). Otherwise, each load bus sheds a fraction of its electrical demand. However, no individual load bus is willing to reduce its power demand by more than 9.4%, corresponding to all the α_i having a value of zero.

Generator	$1/R$	k_{agc}
1	300	0
2	225	200
3	300	0
4	300	0
5	225	0

TABLE I

GENERATOR PARAMETERS

Load bus	$t = 0.0$	$t = 1.0$	$t = 10.0$
1	0.0		
2	-0.217		
3	-0.942		
4	-0.112		
5	-0.478		
6	-0.076		
7	-0.295	0.0	-0.4
8	-0.09	0.0	-0.09
9	-0.035		
10	-0.061	0.0	-0.4
11	-0.135	0.0	-0.135
12	-0.149	0.0	-0.149

The monitors and load buses communicate through a packet switching network. Communications lines follow the network transmission lines, and packets are routed from origin to destination through this shared communication medium. The communication lines are modeled as queues with a fixed throughput, measured in bits per second (bps), and base delay. The time required for a packet to traverse a single line is given by Each line has a buffer for queuing packets, and only one packet can traverse the communication line at any time. No packets are dropped. In general, a message will need to travel through several lines before reaching its destination. Network flooding is used to implement the broadcast function, with unique packet identifiers used to prevent rebroadcasting of packets that have already been processed [14].

EXPERIMENTS

In the first set of experiments, the network throughput was fixed at 1Mbps, and the base line latency was varied between 20ms and 135ms. The five generators fail shortly after $t = 10$ in the 135ms scenario. The generator frequency, load service fractions, and electrical supply and demand totals for the 130ms scenario are shown in Fig. 5 and 6. This is the upper end of the survivable latency range, and there is a noticeable ripple in the system frequency, caused by an oscillating load service fraction

power demand dt

Figure 9 shows how the controller performance varies as a function of base line latency. The controller performance
where t_{end} is the end of the experiment observation time (20 seconds in this case).

The electrical demand schedule that was used in these experiments is shown in Table II. The $t = 0$ column shows initial power demand at each bus. Subsequent columns contain an entry only for buses at which the power demand changes. Electrical demand is described by 'per unit' power injected at the load bus. Without any load control, this schedule causes all five generators trip offline following the load spike at $t = 10$ seconds. The failure scenario is shown in Figure 4.

decreases gradually as the base latency increases. Catastrophic failure occurs when the base line latency reaches 135ms (not shown in the figure).

In the second set of experiments, the base line latency was fixed at 20ms and the throughput varied between 115Kbps and 10Gbps. The system fails at 115Kbps with all of the generators tripping offline. The generator frequency, load service frac- tions, and electrical supply and demand totals for the 256Kpbs scenario are shown in Fig. 10 and 11. The system behaves nicely in this case, with the serviced load meeting demand

The network performance as a function of throughput is shown in Fig. 12. The performance actually decreases as additional throughput becomes available, reaching minimum performance at 30Mbps, and then stabilizing near 99.65% at 100Mbps. Fig. 13 shows that the number of load-shedding events changes as the line throughput changes. The number of load shedding events stabilizes as the performance metric stabilizes. The surprising drop in performance as throughput increases is due to the complex interaction of the control scheme, electrical power flow, and communication network. As the link bandwidth changes, the queuing behavior in the network changes, and this effects the order in which the loads are shed from the system. Because individual loads vary in size, the order in which they are reduced substantially impacts the behavior of the electrical network and load control system.Moreover, the impact of a particular load shedding event is dependent on when it occurs and where the bus it is located in the network. The combined effect on overall sytem behavior would be difficult to anticipate without an integrated, dynamic model of the power, control, and communication systems.

CONCLUSION

We have described and implemented an approach for the simulation of integrated power and communication systems. Modeling the continuous and discrete systems together in the power systems scenario in a manner that preserves the correctness of the system is a central contribution of this paper. We leveraged DEVS methodology to create the hybrid model and implemented the system using adevs and NS2.

Our experiments with a load-shedding scenario exposed behaviors that can only be observed with an integrated hybrid model. In particular, we find that network flows affect load- shed order and available bandwidth and network latency affects the control behavior in subtle ways. Identifying these behaviors requires running detailed simulations. Because it is intractable to create analytical models of these integrated systems, simulation is a necessary. Analysis of complex,

REFERENCES

C. Hauser, D. Bakken, and A. Bose, "A failure to communicate: Next- generation communication requirements, technologies, and architecture for the electric power grid," *IEEE Power & Energy Magazine*, vol. 3, no. 2, pp. 47–55, Mar/Apr 2005.

M. Shahidehpour and Y. Wang, *Communication and Control in Electric Power Systems*. New Jersey: Wiley Interscience, 2003.

S. Bhowmik, K. Tomsovic, and A. Bose, "Communication models for third party load frequency control," *IEEE Transactions in Power Systems*, vol. 19, no. 1, pp. 543–548, February 2004.

M. Amin and B. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE Power and Energy Magazine*, vol. 3, no. 5, pp. 34–41, 2005.

P. J. Antsaklis and X. D. Koutsoukos, *Software-Enabled Control: Infor- mation for Dynamical Systems*. IEEE Press, 2003, ch. Hybrid Systems Review and Recent Progress.

P. J. Antsaklis and J. Baillieul, "Guest editorial: Special issue on networked control systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1421–1423, September 2003.

K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury, "EPOCHS: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf compo- nents," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 548– 558, May 2006.

B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation, 2nd Edition*. Academic Press, 2000.

J. Nutaro, T. Kuruganti, and M. Shankar, "A technique for integrating hybrid system simulation models and discrete event simulation software packages," 2006, submitted to the 2007 Annual Simulation Symposium, preprint available at http://www.ornl.gov/1̃qn/pubs.html.

"The Network Simulator - ns2," 2006, online at http://nsnam.isi.edu/nsnam/index.php/User Information.

"*adevs*: A Discrete Event system Simulator," 2006, online at http://www.ornl.gov/1̃qn/adevs.

F. E. Cellier and E. Kofman, *Continuous system simulation*. Springer, 2006.

S. Mullen, "Power system simulator for smart grid development," Master's thesis, University of Minnesota, Minneapolis, MN, May 2006.

A. S. Tanenbaum, *Computer Networks, Third Edition*. Upper Saddle River, New Jersey: Prentice Hall PTR, 1996.