Simulation Framework for

Demand Response Analysis and Control System

Prepared For:
California Institute for Energy and Environment
University of California Office of the President

In Response To:
Research Opportunity Notice DR ETD-02-01

Lead authored by:
Oak Ridge National Laboratory
1 Bethel Valley Rd.
Oak Ridge, TN 37931
Phone: +1-865-574-2704
Fax: +1-865-576-0003

July, 2009

Document History

Revision History

Revision Number	Revision Date	Revision / Reviewed By	Summary of Changes	Changes marked
1.0	051009	Arjun Shankar	Original Document	
1.1	052009	Arjun Shankar and Contributors	Added sections and initial case study draft	
1.2	061509	Jim Nutaro/Arjun Shankar	Case Study Details, Simulation components	
1.3	063009	Arjun Shankar	Second Case Study Details, Smoothing Text	
1.4	070409	Jim Nutaro	Updated Case Study	
1.5	071109	Arjun Shankar	Revisions with case study and write-ups overall	
2.0	071509	Arjun Shankar	Final revision	

Contents

1. Introduction and Background 5
1.1 Document Context 5
1.2 Background 5
1.3 Assumptions 6
2. Simulator Framework Components 7
2.1 Power Flow Computation 7
2.2 Communication Computation 7
2.3 Scenario Computation 8
2.4 Integrating Module 9
3. Simulator Functional Components 10
3.1 Conceptual Framework 10
3.2 Available Simulator Frameworks 12
3.2.1 OpenDSS 12
3.2.2 THYME – Toolkit for Hybrid Systems Modeling and Evaluation 13
3.2.3 GridLAB-D 15
3.2.4 PSAT 16
4. Case Studies 18
4.1 Case Study A: Simulation Applied to Automatic Load Management for Frequency Regulation 18
4.1.1 Fundamental dynamics 18
4.2 Case Study B: Simulation Applied to Automatic Under-frequency Demand-Responsive Load Shedding 24
4.2.1 Experiments 28
5. Conclusions 36
5.1 Technology Aspects 36
5.2 Policy/Rules Implementation 36

6. References 38

1. Introduction and Background

1.1 Document Context

This document meets the deliverable requirement for Task 4, Develop Simulation Framework for a Demand Response Analysis and Control System (DRACS) for the California Institute for Energy and Environment's (CIEE) grant for Research Opportunity DR ETD-02-01. The grant was provided to the EnerNex-led team to address two research topics and questions identified in the opportunity notice:

- *"Using a military (or another, such as air traffic control) C3I system as a model, adapt it to conceptually deal with C2I electricity applications such as dispatching DER to keep the lights on. Compare and contrast the chosen C3I model with the requirements for implementing a C^2I strategy for integrating utility information and communications systems. Are there analogies that indicate utilities can operate in a similar fashion? If not, what are the gaps that need to be filled and is it feasible to fill the gaps?*

- *Given current utility systems and assuming the systems are integrated, how would the CAISO or an UDC operate its control centers in a military (or another) C3I style? With up-to-date real-time information and the ability to control all of its available assets, given a particular operational scenario, how would a plan to address the scenario be executed using strategies based on military (or another) C3I?"*

The requirements identified in this document are based on existing use cases (e.g., [1-3]), interaction with San Diego Gas & Electric stakeholders, and coordinated efforts between EnerNex Demand Response experts, the Open Smart Grid AMI Enterprise team, and C3I system engineers at Oak Ridge National Laboratories (ORNL).

1.2 Background

DRACS is a component within the Demand Response Management System (DRMS) responsible for the "situational awareness" of the real time network and Demand Resource topology. The DRMS is the overall management system for the demand response environment. In addition to the situational understanding functionality supported by DRACS, the functionality of the DRMS system includes making business decisions based on business rules on how, who, what, when, and where demand events occur. These business rules can be used for both real time DR events or in planning for day-ahead events. Based on these business rules, the DRMS is responsible for developing the optimal scenarios/combinations of demand resources for addressing each demand response event, and communicating with these demand resources through systems such as AMI, legacy load control systems, and contracted DR providers. The reference architecture document [6] discusses the relationship between DRACS and DRMS in greater detail.

Different sub-components within DRACS manage the database of Demand Resources, provide analysis of DR scenarios against DR events, and provide real time electrical network and Demand Resource visualization capabilities of the overall DR topology. A simulator module within DRACS is responsible for analyzing requests for blocks of energy, capacity, reserves, or ancillary services and providing a predictive confidence level of success for meeting a requested Demand Resource event. DRACS uses measured responses to load demand requests over time to refine its

internal model. We use the term simulator in the context of DRACS as a scenario analysis capability as well as a design simulation capability. The scenario analysis function in DRACS is responsible for evaluating DR scenarios against DR events. It is responsible for analyzing requests for blocks of energy, capacity, reserves, or ancillary services and providing a predictive confidence level for success in meeting the requested DR event with the set of available mechanisms and real-time state. In the operations of DRACS it will be a run-time execution module (i.e., an operations and/or contingency evaluation capability). Simulating any DR system's behavior, though, aims to evaluate expected DR penetration, scope of dispatch strategies, range of behaviors, etc. in varied scenarios (including worst-case) is complementary in role. Here the goal is to determine what strategies need to be designed and made available within the system prior to “going live”. The simulator functionality overlaps with the scenario analysis run-time module. The run-time module must handle a specific case, and thus will need more accurate information of the situation at hand. The simulation module, on the other hand, must capture a broad set of cases and hence some of the inputs may be generalized into a class of cases to consider. For purposes of this document we allow a slight abuse of terminology in referring to these dual capabilities with the term *DRACS Simulator*. When clarification is required we refer to the run-time sub-component as the *DRACS Scenario Analyzer*.

This document gives details of the simulation framework. Specifically, we consider existing tools and approaches to performing the simulation and scenario analysis required in DRACS, and we illustrate the concrete results with simplified case studies using the THYME [7] simulator. These case studies outline the workings of the DRACS simulation component and offer guidelines for simulator designers.

1.3 Assumptions

Assumptions

1. The simulator operates within the DRACS system which itself resides within the overall DRMS system. Although we list as complete a set of the requirements as possible, the availability and prioritization of the inputs during simulator construction is a design-time decision.
2. When appropriate, in addition to meeting the requirements described within this document, the system designer/implementer must also be compliant with ALL appropriate standards organizations’ terminology, interoperability, and messaging. This includes:
    a. IEC 61968 Application integration at electric utilities – System interfaces for distribution management.
    b. IEC 61970 Application integration at electric utilities – Energy management system application program interface (EMS-API).
    c. North America Energy Standards Board (NAESB) Demand Side Management and Energy Efficiency (DSM-EE) standards
3. Simulator requirements will be informed by California requirements for Demand-Response. In most cases, however, these requirements will apply to other regions and areas in the power grid.
4. Input requirements such as consumer behavior representations to the simulator will be informed by prior studies. These are likely to evolve and change over time – consequently the system design will allow evolution and growth in simulator requirements.
5. Readers are broadly aware of basic simulation approaches including discrete event simulation and process-based simulations.

2. Simulator Framework Components

The simulation framework builds on the requirements discussed in DRACS Simulator Requirements Document [4]. We discuss the framework by giving the primary components that are required in the simulator and approaches to their implementation. For concreteness and clarifying what the simulator offers we also give specific case studies in Section 4.

2.1 Power Flow Computation

The core of the simulator is a power flow computation module that takes into account a steady-state (DC) power flow generation, transmission, and distribution configuration. Several industry power flow solvers take a topology and compute load-flows for DC configurations of the network. For the simulator component in DRACS, the DC configuration is adequate because the scenarios do not include transient analyses. This module is commonly available and will need to be modified for the available input parameters for the particular practitioner's environment.

2.2 Communication Computation

The power grid infrastructure continues to evolve as a large complex system supported by a growing and complex network of sensors and controls that operate through wide area communication networks. The communications and control system for the power grid has been designed primarily to regulate power flow via generator scheduling, regulating power demand, and long haul power transmissions through transmission corridors and inter-ties. In the future, however, more rapid and adaptive control systems will be required to respond in real-time to power system events and pricing.

The advent of demand side management and, more broadly, distributed energy resources demands a careful characterization of the communication system that is needed to operate a large-scale power system in real-time. The efficiency of the power system is increasingly dependent on the networked SCADA and distributed control system loops that use heterogeneous communication technologies. Recent proposals for Distributed Energy Resources (DER), transactive processing, and Demand Response Enabling Technology Development (DRETB) on the power grid require millisecond responses among tens of thousands of intelligent agents and smart devices to negotiate a price, respond to rate fluctuations, and report status.

There is a need to develop a characterized and bounded behavior for the evolving communication system standards for the future grid. This need translates to including the communication emulation within the simulator framework because different monitoring approaches and their communication modalities will influence the delays encountered during peak shaving accomplished by AMI-based or load-aggregation-point driven demand side resource management.

A survey of the literature shows that there are no software modules that offer a verifiable formal approach to integrating the communication aspects of a simulator simultaneously with the power flow analysis. (The EPOCH tool [13], e.g., takes an approach that integrates disparate tools

over synchronization intervals instead of a provably correct joint simulation methodology.) This is a key need because discrete event simulation may produce inaccurate simulation results if the events from the power system space and the communication space are not coordinated. The important requirement here is that the simulation be provably correct in coordinating the time domains of the power system analog world and the communication network's digital world. To our knowledge, THYME [7] is one of the only simulators that takes this approach.

2.3 Scenario Computation

The DRMS system [4] provides scenarios for the simulator to evaluate in the run-time operations mode. For the offline simulator, scenarios are created as time-stamped profiles of system behaviors and operator responses (described by system rules). The simulator "walks through" the time-stamped evolution of the system behavior checking for instability. A module in the simulator systematically explores the state space of the potential system behaviors.

The steps in the scenario computation algorithm are as follows:

For Run-Time Operations:

1. Input current system operating points, generation levels, and loads. Including projected generation and load profiles.
2. Input range of target demand reduction.
3. For the database of demand reduction, take rule-base for rotating demand reduction to generate target areas.
4. For candidate target areas divide the demand reduction according to pre-defined policies (e.g., lumpsum, equally divided, divided by a preset ratio, etc.)
5. Calculate expected load reduction based on initial estimate (later, based on historical data and current system state estimation algorithm)
6. Compute load-flow to estimate degree of instability or dispatch.
7. Compute confidence levels for dispatch. If appropriate confidence level has been reached for the amount of dispatch, then provide feedback to operator.
8. Compute next alternative target areas for demand reduction and repeat steps 3-8 until the confidence level has reached, or the number of pre-defined iterations has been exceeded.

For Off-Line Analysis:

The steps are similar to those for the run-time operations with the exception that the input status of loads, generation, and demand is varied across a wide variety of inputs to generate the average-case.

For the worst case, historical information and expert utility (e.g., the California ISO) input will have to provided a candidate set of generation, load, and environment conditions that are likely to exercise the system extremes. The expectation of dispatch will also be skewed to the lower end of the probability range to simulate an inelastic system. To allow for high variability, the expectation of dispatch is skewed to the higher end of the

probability range – i.e, most of what is demanded becomes available at rates corresponding to the fastest one may anticipate at any one point in the system.

2.4 Integrating Module

An integrating module component gives the overarching structure for the simulator and supports these major components. Subcomponents within these major components are threaded together with a separate flow harness (which consists of the top-level simulation algorithms discussed in Section 2.3) for each of the simulator behaviors in offline and online mode.

3. Simulator Functional Components

In this section, we give the functional components that derive from the requirements outlined in Simulator Requirements Document [4] and discuss existing simulator frameworks that are candidates for addressing the DRACS simulation need.

3.1 Conceptual Framework

Figure 1: DRACS Architecture Components [6] describes the block diagram of the DRACS system with the simulator and scenario analyzer module called out.

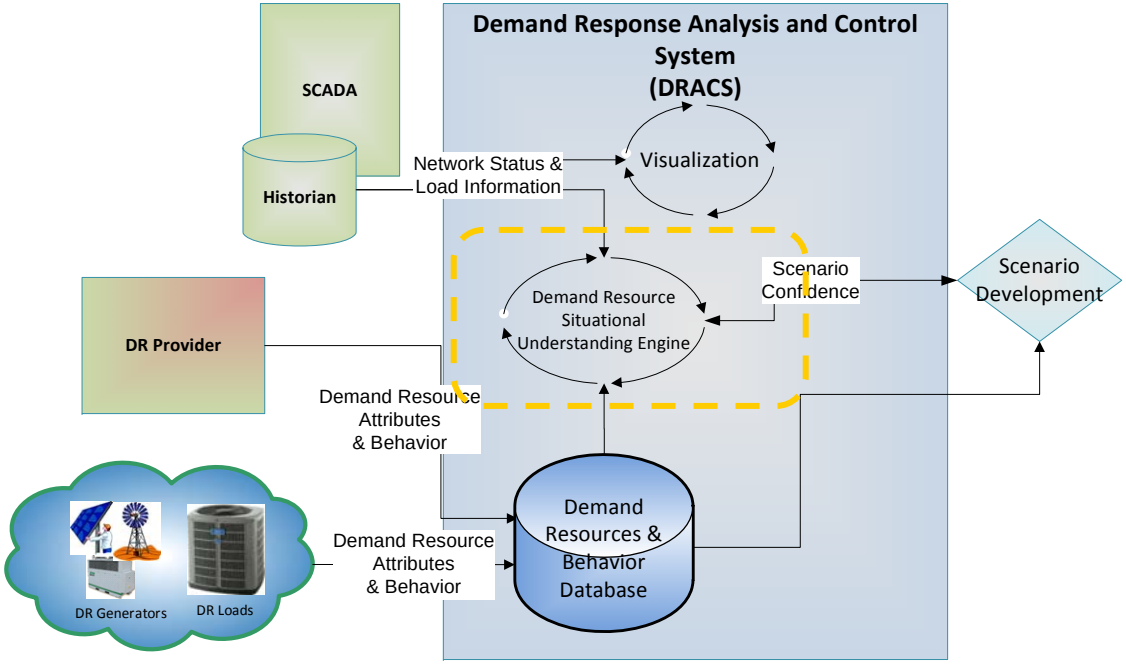

Figure 1: DRACS Architecture Components

The conceptual framework within the simulator is illustrated in Figure 4 below:

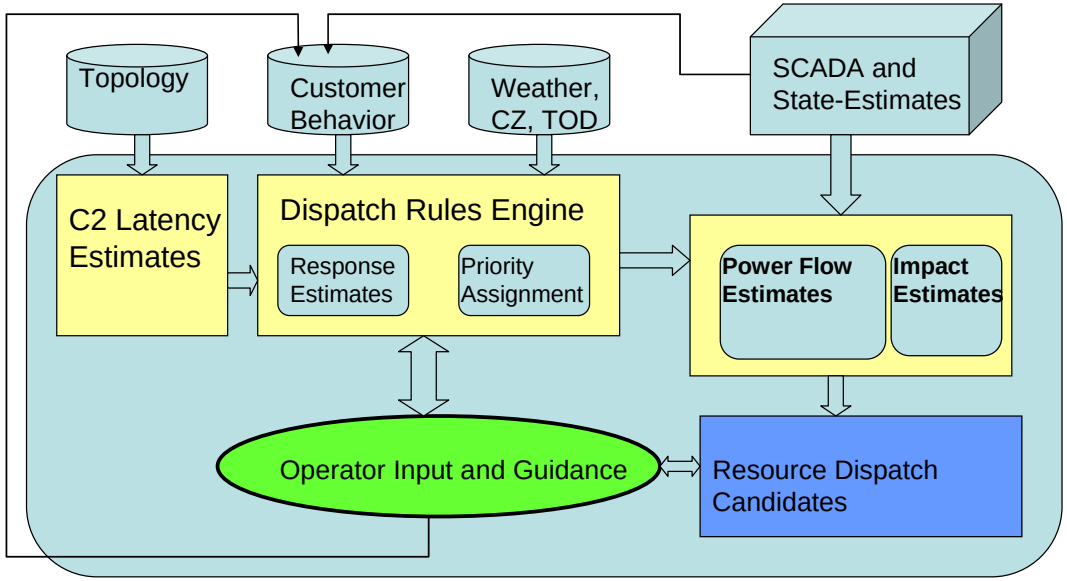

Figure 2: Operational Flow in Simulator Modules

The operational flow of the run-time simulator function is illustrated by the flow of data from the databases through the solver and into an operator feedback and control tool.

The DRMS generates scenarios for managing each DR event. These scenarios are built based on business rules and from a collection of Demand Resources in the Demand Resource Attribute and Behavior database that meets both the business rules and DR event requirements. DRACS receives a scenario from the DRMS. The simulator's on-line analysis function evaluates the scenario against the collection of Demand Resources Baseline estimates and previous behavior, current electrical network events, and topology communication and social response algorithms and calculates the likelihood of success in meeting the DR event objectives. DRACS compares against "Baseline" values which are estimates of the electricity that would have been consumed by a Demand Resource in the absence of a Demand Response Event as discussed in [6]. Depending on the type of Demand Response product or service (one of energy, capacity, reserve, or regulation service), baseline calculations may be performed in real-time or after-the-fact. The baseline is compared to the desired metered electricity consumption during the Demand Response Event to determine the Demand Reduction Value. There are three Types of Baseline Models

1. Baseline Type 1 (Interval Metered) - A Baseline model based on a Demand Resource's historical interval meter data which may also include but is not limited to other variables such as weather and calendar data.
2. Baseline Type 2 (Non-interval Metered) - A Baseline model that uses statistical sampling to estimate the electricity consumption of an Aggregated Demand Resource where interval metering is not available on the entire population.

3. Behind-The-Meter Generation - A performance evaluation methodology, used when a generation asset is located behind the Demand Resource's revenue meter, in which the Demand Reduction Value is based on the output of the generation asset. Distributed generation resources are considered "behind-the-meter" generators, such as combined heat and power (CHP) systems, wind turbines, and photovoltaic generators that generate electricity on site.

The DRACS simulator uses baseline values and estimates of the dispatched load or available ancillary service to provide statistical confidence levels for the DRMS system to go forward with the implementation of the control decisions.

3.2 Available Simulator Frameworks

We next consider existing and available alternatives that are candidate tools to form an operational simulator consisting of offline and online simulation capabilities.

3.2.1 OpenDSS

OpenDSS is an open source distribution system simulator. The simulator is mainly constructed to support steady-state frequency domain analysis of the distribution system. The figure shows the structure of DSS, it can be executed as a standalone simulator or through another software platform (VB, Python, Matlab). Figure 1(a) shows the software structure of the OpenDSS with a COM interface for external access and scripts interface for setting up stand-alone simulation tools. Figure 1(b) shows the object structure of the OpenDSS tool with built in circuit elements for composition and an iterative solver for power flow simulation. The tool is primarily targeted towards distribution planning and analysis with an ability to analyze and recommend possible distribution generation locations (time and location value of a particular DG), annual generation and load simulations, harmonics analysis along with power flow (allowing to study non-linear behavior of certain elements using "compensation currents") , fault analysis, and general AC circuit analysis.

The tool has post-processing elements to compute currents and power at branches and nodes and metering tools like the EnergyMeter. The tool comes with a suite of plotting options either using the internal plotting tool or by exporting data to be readable by Excel or Matlab for enhanced plotting requirements. Although the tool does not have a high-performance iterative solver, the fact that it is open source and available for a relatively large subset of power system elements (distribution-level) makes it a powerful and flexible research and analysis tool. The tool is currently windows-based with a graphical programming interface and a simple to use COM interface for driving the simulator from an external program.

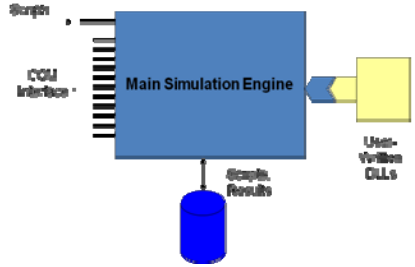

Figure 3(a)

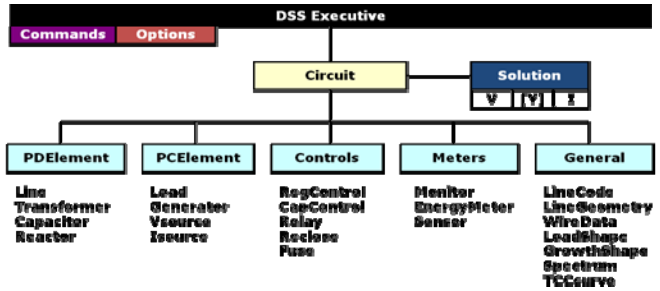

Figure 1(b)

3.2.2 THYME – Toolkit for Hybrid Systems Modeling and Evaluation

THYME is a hybrid simulation tool aimed at network-based systems was developed using the *NS2[14]* and *adevs[15]* 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 NS2 simulation model using the simulation control API that is part of the adevs simulation software.

The simulation package includes a power system model that is suitable for examining control and communication problems in the context of dynamic stability. The power system component consists of a synchronous machine and transmission network model built from the swing equation, a simple generator excitation and frequency control model, area generation control, and a sinusoidal steady state model of the transmission network. These are implemented with the Adevs discrete event simulation software, which uses an Runge-Kutta integration scheme to simulate the power system's continuous dynamics. Discrete control logic and simple communication models can be implemented using Adevs, and these can be combined with the power system model. The adevs-ns2 package allows NS2 to be used when more detailed models of the communication network are required.

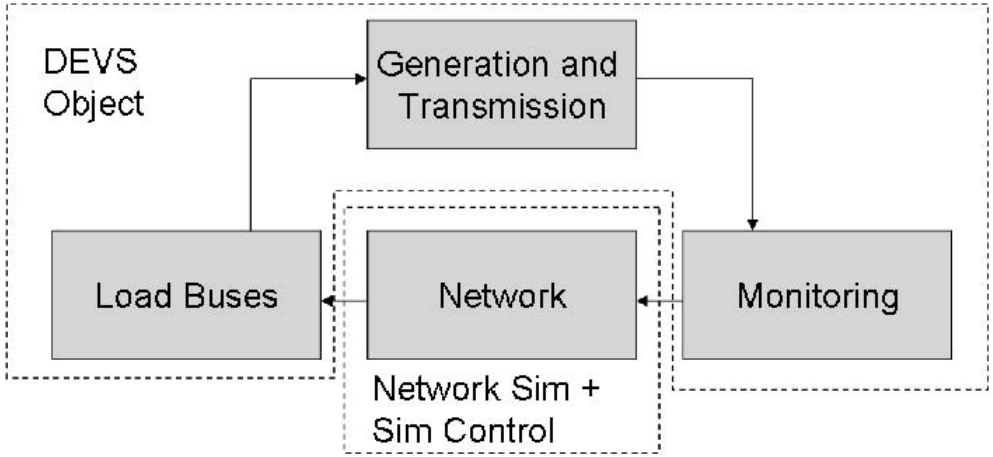

Figure 4: The components of THYME power system model.

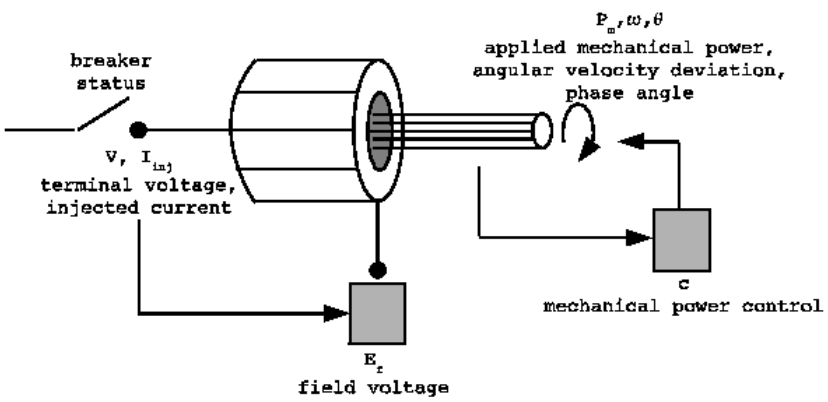

Figure 5: Elements of THYME's synchronous machine model.

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. NS2 invokes the *adevs* components when two types of events occur:

1. an internal *adevs* event and
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 NS2 simulator performs the following actions:

1. it queries the *adevs* module for any network messages that need to be sent,
2. it schedules NS2 events to send messages returned by the above query,
3. it tells the *adevs* component to update its internal state, and
4. it queries the *adevs* component for its next event time and schedules a corresponding *adevs* NS2 event.

Type 1 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 state using the injected events, and
3. query the *adevs* component for its next event time and schedule a corresponding *adevs* event in NS2.

The DEVS structure (4) is used to simulate continuous processes. A fourth order Runge-Kutta integration scheme is used to solve the ODE set. Zero crossing functions are used to describe the location of state events. Time events are scheduled explicitly.

State events are detected using discontinuity locking in conjunction with interval bisection. With this approach, the continuous variables are integrated over a single integration step. If the sign of a zero crossing function is unchanged at the end of the step, then it is assumed that no state event occurred in the interval. Otherwise, the integration step size is halved, and the test is repeated. If this procedure causes the step size to reach some small threshold value, then the event is assumed to occur at the end of that minimal step size. THYME is open source and we use it in the case studies discussed later in this document.

3.2.3 GridLAB-D

GridLAB-D is a new power distribution system simulation and analysis tool that provides valuable information to users who design and operate distribution systems. It incorporates the modeling techniques, with high-performance algorithms to deliver end-use modeling. GridLAB-D is coupled with distribution automation models and software integration tools for users of power system analysis tools. GridLAB-D is being developed by the U.S. Department of Energy (DOE) at Pacific Northwest National Laboratory (PNNL), in collaboration with industry and academia. GridLAB-D Version 1.2 was released in June 2008. Version 2.0 is planned for release in the fall 2009 [8].

At its core, GridLAB-D has an advanced algorithm that can determine the simultaneous state of millions of independent devices, each of which is described by multiple differential equations solved only locally for both state and time. GridLAB-D can simulate distribution systems to a high level of detail, even the behavior of specific appliances such as refrigerators and air conditioners. It models transmission systems, connecting with distribution-level models to form a full grid-modeling tool. The advantages of this algorithm over traditional finite difference-based simulators are: it is much more accurate, it can handle widely disparate time scales ranging from sub-seconds to many years, and can integrate with new modules and third-party systems. GridLAB-D does not require the use of reduced-order models, so the danger of erroneous assumptions is averted.

The GridLAB-D system currently implements modules to perform the following functions:

- Power flow and controls, including distributed generation and storage
- End-use appliance technologies, equipment and controls
- Data collection on every property of every object in the system
- Boundary condition management including weather and electrical boundaries.

In the current version of GridLAB-D the AC power flow solution method used for the transmission system is the Gauss-Seidel (GS) method [9], chosen for its inherent ability to solve for poor initial conditions, and to operate well in multiprocessor environments. The distribution module utilizes the traditional forward and backward sweep method [10] for solving the unbalanced 3-phase AC power flow problem. The following power distribution system components are implemented and available for use: overhead and underground lines, transformers, voltage regulators, fuses, switches, and shunt capacitor banks. Commercial and residential end-uses are implemented using the Equivalent Thermal Parameters model [11]. Currently implemented residential end-uses are: electric water heaters, refrigerators, dishwashers, cloth washers and dryers, ranges and microwaves, electric plugs and lights, internal gains, and house loads (including air conditioning, heat pumps, and solar loads).

GridLAB-D aims to model a variety of concepts not easily modeled before such as: the impact of demand response programs and grid-friendly appliances, the effect of distributed energy resource (DER) and energy storage, the business case for Smart Grid technologies, the impact of new control schemes, such as conservation voltage reduction, and the introduction of plug-in hybrids and the use of Vehicle to Grid technologies at varying penetrations. GridLAB-D aims to provide the ability to model consumer choice behavior in response to multiple rate offerings (including fixed rates, TOU, CPP, and RTP) to determine whether a suite of rate offerings is likely to succeed. GridLAB-D can also be calibrated to observed consumer behavior to understand its interaction with various peak shaving strategies. The impact of consumer satisfaction on the available of peak-shaving resources can be evaluated and a more accurate forecast of the true available resources can be determined. GridLAB-D will even be able to evaluate the consumer rebound effect following one or more curtailment or load-shed events in a single day.

GridLAB-D does not consider the communication network explicitly to include different communication modalities and control latencies in a provably correct integrated framework. This may open the simulation to errors in time-accuracy. While this may not be a shortcoming for the environments simulated, it remains an area of uncertainty.

3.2.4 PSAT

Power System Analysis Toolbox (PSAT) [12] is an open source Matlab and GNU/Octave-based software package for analysis and design of small to medium size electric power systems. PSAT includes power flow, continuation power flow, optimal power flow, small-signal stability analysis, and time-domain simulation, as well as several static and dynamic models, including nonconventional loads, synchronous and asynchronous machines, regulators, and FACTS. PSAT is also provided with a complete set of user-friendly graphical interfaces and a Simulink-based editor of one-line network diagrams.

PSAT is not aimed at a communication overlaid infrastructure that controls a demand-responsive system such as DRACS. It is however a useful component of the load-flow solution component.

4. Case Studies

We present concrete examples of using a simulation framework to study demand-side management problems using different approaches and how a simulation provides insights to the user (operator, in this case). The first case study considers dispatch for load management to accomplish frequency regulation. The second case study considers the effect of latencies in performing load-reduction (agnostic to the specific technique used). Different latencies are received by the system differently.

The goal of the discussion of the two case studies below is not the specific case results per se as much as the application of the simulation framework to a system in which load management and dispatch interacts with communication and control which is the central simulation concern in the DRACS simulator module.

4.1 Case Study A: Simulation Applied to Automatic Load Management for Frequency Regulation

Controllable load can in part replace generation for frequency regulation and spinning reserve. How this is done depends on the time scales available for control. Control of frequency in an emergency, to prevent under frequency failures for example, requires action in seconds to minutes. Load acting as spinning reserve has minutes to act. The potential for dynamic control can be understood with a very simple model, which will be further explored by simulation of a detailed model of a centralized control process operating on the IEEE 118 bus test system.

4.1.1 Fundamental dynamics

Proportional control of electrical load, which ties change in load to change in frequency, has been suggested by several authors as a technique for dynamic frequency regulation. Both are natural extensions of the Use Cases[1-3] considered in this analysis, which takes advantage of the automatic load regulation that could be accomplished with AMI. The potentials benefit of load modulation can be understood with a simple model. In a power system, frequency changes at a rate proportional to the difference between electrical supply and electrical demand. At a large scale, changes in frequency away from nominal are resisted by losses due to inefficiencies at off nominal operations and by active control of the generator output to dampen frequency excursions. Load can contribute to the latter, augmenting or replacing the actions of generators to bring the system's frequency back to its nominal value.

To be precise, the model is formulated as follows. It has four parameters: the net inertia m of the generating units, the resistance r to operating away from nominal frequency, and the gains g of the generators' response and l of the loads' response to a frequency excursion. The gains g and l lump together the gains of speed governors, rate of response of spinning reserves, reaction time of the loads to a request for curtailment, and other, similar factors. The frequency deviation f from nominal and power imbalance p are related to each other by the pair of differential equations

$$(d/dt)\, f = (p - r f) / m$$

$$(d/dt)\, p = -(l + g)\, f$$

and the role of load modulation is immediately apparent: it augments the generation control that is already in place. A power system at equilibrium has *f* and *p* equal to zero. If there is a mismatch between generation and supply, the second equation describes the response of generation and load controllers to correct the imbalance. Significantly, if the system is stable without active control of the load (i.e., $l = 0$), then it is stable with active load control (i.e., $l > 0$) *and* the frequency excursions are extinguished more rapidly; introducing this technology always yields a benefit. Local, automatic controls at the generators permit an approximately continuous response to frequency deviations. If, however, the loads are managed from a utility's control center, then communication effects will modify the performance of the system as a whole.

1.1.1. Experiments

For this simple analysis, two aspects of the control and communication system are considered: the communication threshold Δf at which a generator sends its frequency measurement to the control center, the data rate of the network which carries sensor and actuation signals, and the quantity of background traffic that is being carried on that network. A simple model of a capacity limited communication network is a first come, first served queue with fixed service time. With this embellishment, the model captures the capacity limits of the communication network and the delays introduced by competition for network resources. We can now examine the network capacity that must be committed to obtain the most benefit from the load control system.

This scenario uses the IEEE 118 bus test case. There are 34 generators, each modeled as shown in Figure 5, and having uniform dynamic parameters. To emphasize the role of the load control in damping frequency oscillations, the control parameters of the generators are set to give the system an under-damped response to a step change in load. For this experiment, there is a 30% reduction in the electrical load at $t=1$ second. Figure 6 shows the response of the 34 generating units when active load control is not used; the under-damped response is obvious in the large, initial swings between 61 and 58.5 Hertz. Figure 7 shows the improved response for the same experiment but with active control of the loads; superimposed on the frequency plot is the load reduction effected by the control center. In this case, the control center is notified of every 0.2 Hz change in frequency at the terminals of the generators. These measurements arrive via the communication network, and, upon receipt, the control center uses the average value of its frequency measurements to determine the quantity of load which should be shed. The adjusted load level is then transmitted to the load busses via the communication network, each bus acting on the request when it is received.

For this demonstration, the data rate of the network is fixed at 10^4 packets per second and the sampling threshold Δf and quantity of background traffic are varied. This reveals the sensitivity of the system to i) the amount of non-control data on the network (e.g., as produced by other Smart Grid applications); ii) the precision of the frequency measurement received by the control center; and iii) the electro-mechanical dynamics during the contingency. If the control system performs best if the frequency threshold is small and the network is dedicated entirely to transporting sensor and control signals: this situation is shown in Figure 7, where $\Delta f=0.02$ Hz and there is no background traffic.

To compare scenarios with different network parameters, the performance of the controller is measured by the integral of the square of the frequency excursion over the duration of the experiment; specifically, by calculating

$$J = \sum_{g \in \text{generators}} \int_0^{20} [f_g(t)]^2 \, dt$$

so that if J is small, the control worked well. Figure 8 shows the performance of the system as function of Δf and the background utilization of the network. Two facts are immediately apparent. First, the performance generally improves as the sensor threshold becomes smaller, but there are diminishing returns past Δf =0.05 Hertz. Second, a greater portion of the network's capacity must be held in reserve to ensure that the contingency, a 30% loss of load in this case, does not overwhelm the network. If this occurs, the generators may stabilize the system before the

load adjustments are delivered to the load actuators. This delayed control signal then becomes a contingency itself, generating more control data, and further aggravating the problem.

This simple study clearly shows that there is a tradeoff to be made between the efficacy of the load control scheme (as determined by its sensor transmit threshold) and amount of network capacity require to support it. Moreover, using this simple model of the communication network, there appears to be a minimum capacity required to ensure the control actions do not themselves instigate a destabilizing dynamic. Simulations of this kind are invaluable for examining these kinds of communication induced effects in distributed control systems, and this case study illustrates the calculation that the DRACS simulator performs. The results suggest that the computation of the dispatch impacts on the overall system stability are closely linked with the delay itself and the throughput for the monitoring signals. Unless this is simulated carefully, the operator's dispatch decisions may have dramatic consequences when the DR penetrations assume significant percentages. In the following section, a more sophisticated system for load control, with a more realistic model of its supporting communication network, are considered.

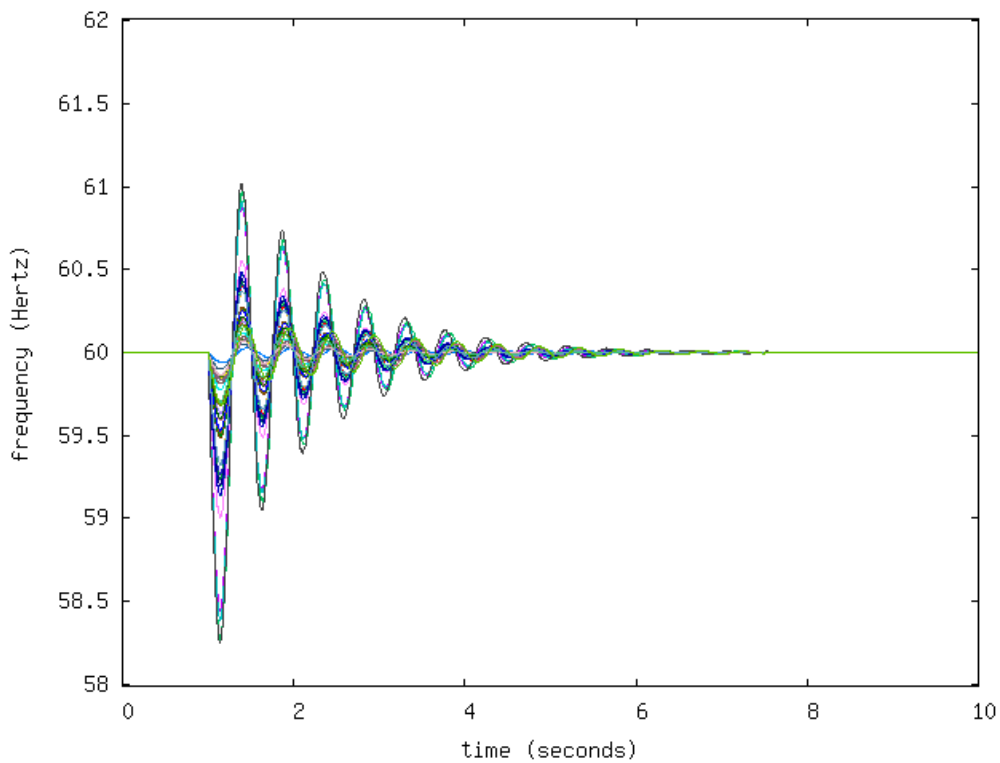

Figure 6: Frequency response without load control.

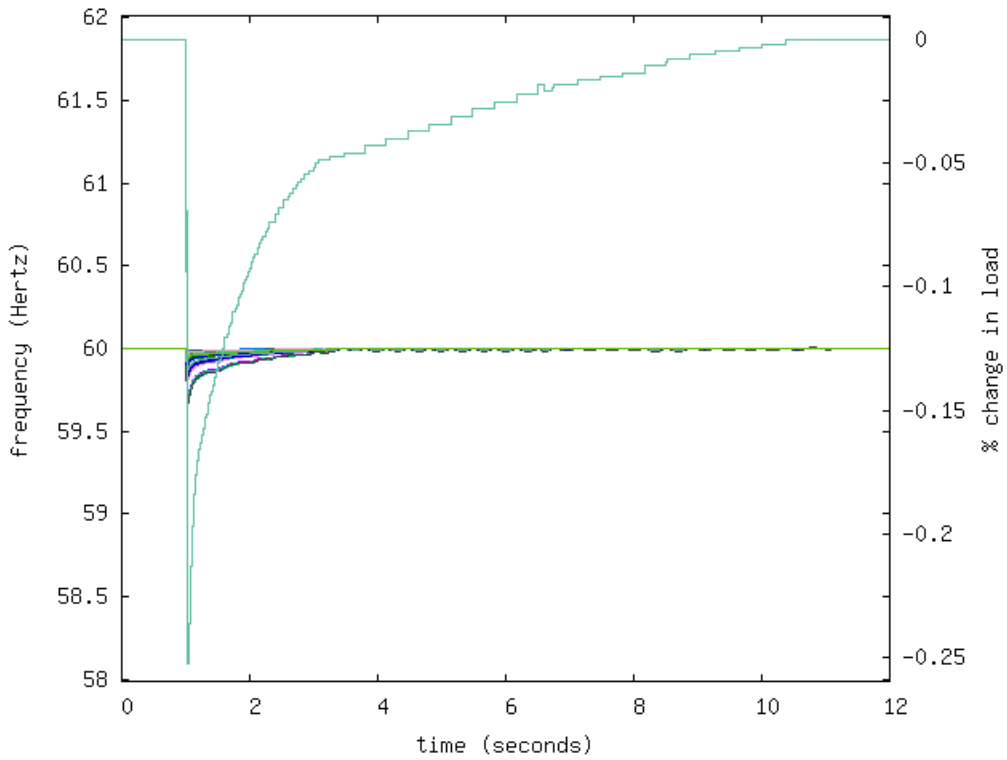

Figure 7: Frequency response with load control.

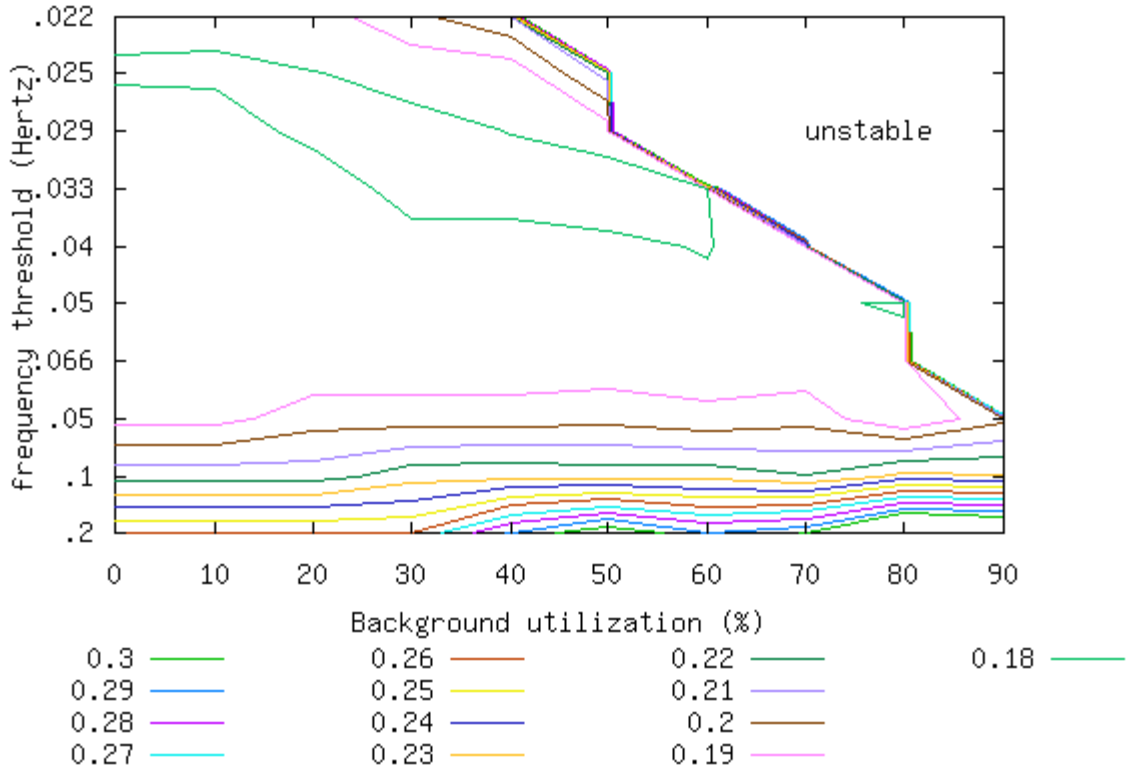

Figure 8: Response surface of the network as a function of its communication parameters. Contour levels are the value of the performance metric J. There is a gradual improvement in performance as the frequency threshold shrinks, but a corresponding requirement for more network capacity.

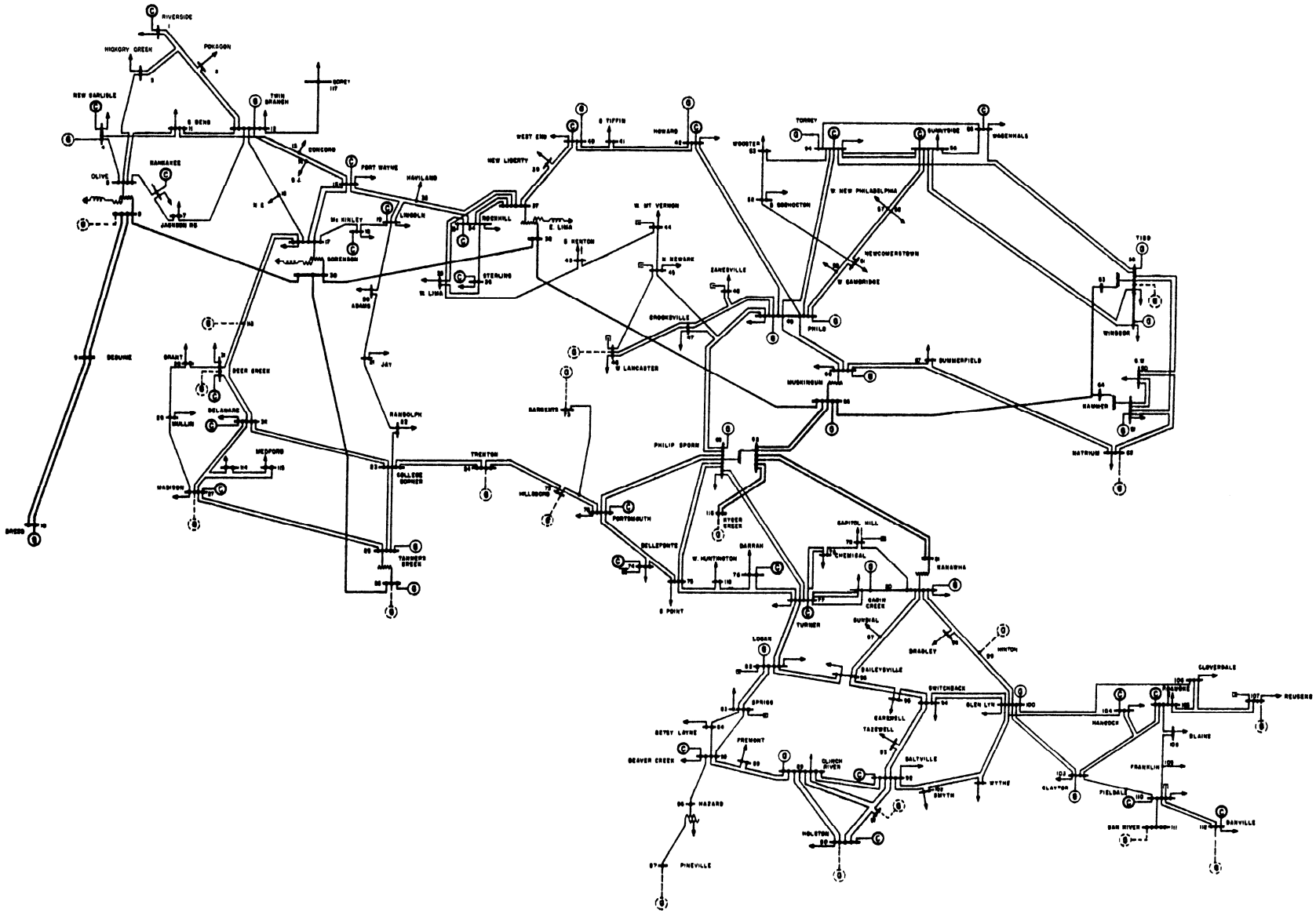

Figure 9: IEEE 118 bus test case.

4.2 Case Study B: Simulation Applied to Automatic Under-frequency Demand-Responsive Load Shedding

Adapted and built upon James Nutaro, Teja Kuruganti, Laurie Miller, Sara Mullen, and Mallikarjun Shankar, "Integrated Modeling of the Electric Grid, Communications, and Control," IEEE Power Engineering Society General Meeting, June, 2007 [7]

In this section, we use a notional system for automatic load control to demonstrate the capabilities of THYME's 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 calculations for this model, three additional buses were added to the IEEE 14 bus model to separate buses with both load and generation. The model consists of twelve loads and five generators that are interconnected as shown in Fig. 2.

The power flow and load shedding aspects of this model have been simplified in two ways. First, the power system model was developed to study the mechanical response of generators to significant load changes and generator or transmission line losses REF. The bus voltages are assumed to remain constant, line resistance and shunt reactance are neglected, and a linearized DC power flow is used to calculate the real power flows in the network. This simplification is justified because the load control mechanism studied here is triggered by frequency variations; consequently voltage deviation and reactive power flow are neglected. Second, all loads are assumed to have an interruptible component, and every load responds immediately to shedding requests. This simplifies both the controller and load model by assuming ideal load behaviors. We expect that a model which includes both reactive power flow and market driven load behaviors may predict different outcomes from what is shown here. Nonetheless, these two simplifications give us a manageable starting point and allow for a clearer presentation of the model and its hybrid dynamics.

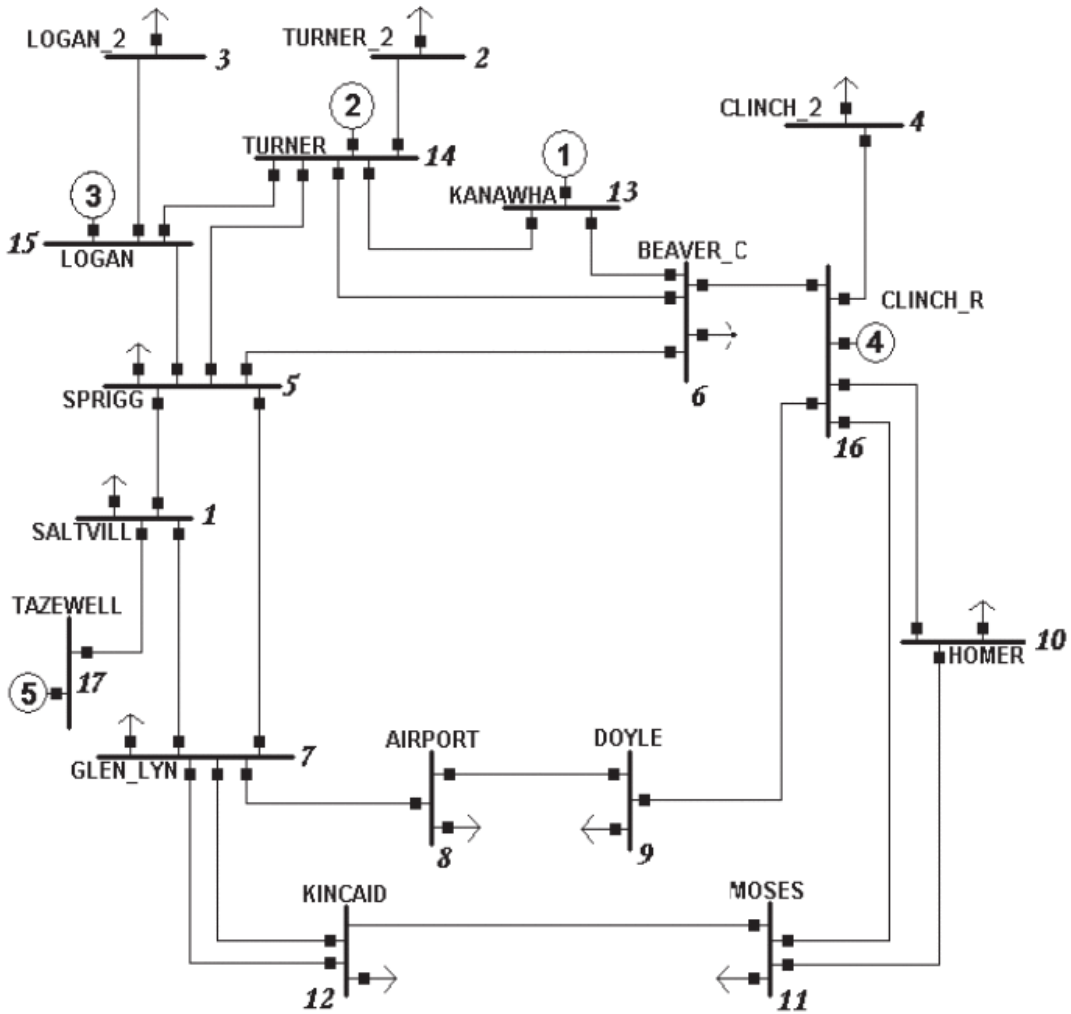

Figure 2. The 5 generator and 12 load bus IEEE power system model.

The generators are modeled as synchronous machines using the swing equation plus additional equations that model a governor, non-reheat turbine, and overspeed breaker. One of the five generators also includes a basic Automatic Generation Control (AGC) unit that eliminates steady-state frequency error throughout the system. The three equations that describe the generator dynamics are

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

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 the relative generator shaft angle, and $\Delta\omega$ is the deviation of the shaft angular velocity from 60Hz. The values R, M, and k_{agc} are the generator's speed droop constant, rotational inertia, and AGC gain. If the speed deviation of a machine exceeds ± 0.1 Hz, then it is disconnected from the transmission network.

Real power flow is calculated from generator shaft angles and electrical power demand at the load buses. Disconnected generators are treated as load buses with zero power demand. To facilitate the calculation of electrical demand P_e on the generators, the network admittance matrix Y is broken into the four sub-blocks shown in Eqn. 6. The Y_{ll} block describes load to load connections, Y_{lg} and Y_{gl} describe the symmetric generator-load/load-generator connections, and Y_{gg} describes generator to generator connections. Similarly, the bus angle and electric power vectors are split into upper and lower blocks. The vectors $\bar{\Theta}_l$ and $\bar{P}_l$ denote the load bus angles and injected power. The vectors $\bar{P}_e$ and $\bar{\Theta}_g$ are the electrical demand on the generators and the generator shaft angles. The power flow equations are

$$\begin{bmatrix} \bar{P}_l \\ \bar{P}_e \end{bmatrix} = \begin{bmatrix} Y_{ll} & Y_{lg} \\ Y_{gl} & Y_{gg} \end{bmatrix} \begin{bmatrix} \bar{\Theta}_l \\ \bar{\Theta}_g \end{bmatrix} \qquad (6)$$

and the electrical demand on the generators is given by

$$\bar{\Theta}_l = Y_{ll}^{-1}(\bar{P}_l - Y_{lg}\bar{\Theta}_g)$$
$$\bar{P}_e = Y_{gl}\bar{\Theta}_l + Y_{gg}\bar{\Theta}_g \,.$$

Attached to each generator is a monitor that samples the generator state at a fixed rate of $1/T$. The sampled variables are the generator mechanical power ΔP_m, rate of change in mechanical power $\Delta\dot{P}_m$, electrical load ΔP_e, shaft velocity $\Delta\omega$, and shaft acceleration $\Delta\dot{\omega}$. At each sampling instant, the monitor estimates time to an under-frequency failure by

$$t_f = \begin{cases} \dfrac{60.0(\Delta\omega + 1) - 59.9}{|\Delta\dot{\omega}|} & \text{if } \Delta\dot{\omega} < 0 \\ \infty & \text{if } \Delta\dot{\omega} \geq 0 \end{cases}$$

and time to meet demand by

$$t_s = \frac{\left|\Delta P_m - \Delta P_e\right|}{\left|\Delta \dot{P}_m\right|}.$$

The generator is in danger of being disconnected if

$$t_f \le t_s .$$

In this case, the monitor will broadcast a request asking all load buses to reduce their power demand. If, on the other hand,

$$t_f > kt_s$$

where $k >> 1$ is a safety factor, the system is operating under capacity. In this case, the monitor will generate a message 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 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. The monitor state and output are computed at each sampling time. Circles denote discrete phases, and the action performed in each phase is denoted by *state change / output*. Arrows denote phase change conditions. At each sampling instant, the phase change conditions are evaluated and the phase is changed accordingly. Then the output value is produced and, subsequently, the state variable change is applied. A new monitor state and output is calculated every T seconds using the sampled generator variable values.

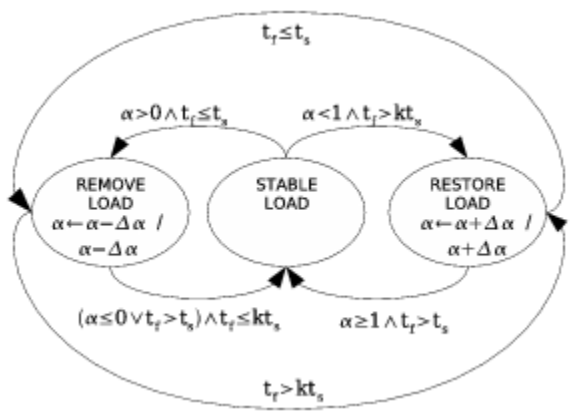

Figure 3. State transition diagram for the generator monitor.

Load buses remember the last load service fraction received from each monitor. The remembered requests are denoted α_i, with $i \in [1,5]$ indicating the monitor that produced the request. Each load bus is also aware of its electrical demand. On receiving a message, the load bus removes or restores some of its power demand, resulting in an overall reduction in demand on the generators. Each bus assumes that it produces $1/12$ of the total electrical demand, and that each generator carries $1/5$ of the total electrical demand. Therefore, whenever a single generator requests a reduction of its load from 1 to α, each load bus disconnects only $(1-\alpha)/60$ of its total electrical demand from the transmission network.

This reasoning leads to the load control rule that is implemented by each load bus. Letting L_d be the full power demand at a load bus, the amount of load served, denoted L_s, is equal to

$$L_s = L_d\left(1 - \frac{1}{12} + \frac{1}{60}\sum_{i=1}^{5}\alpha_i\right).$$

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 8.3%, corresponding to all the α_i having a value of zero.

The monitors and load buses communicate through a packet switching network. Communication 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

$$\frac{\text{bits}}{\text{throughput}} + \text{base delay}.$$

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 re-broadcasting of packets that have already been processed.

4.2.1 Experiments

Two sets of experiments were conducted to study how base line latency and throughput impact the effectiveness of the control scheme. A fixed parameter set was used for the generators, electrical transmission network, and monitors. The initial power P_0 at each generator is calculated to ensure steady state at a selected bus angle. Other generator parameter selections are listed in Table 1. The controller parameters were $T = 0.01$, $k = 10^4$, and $\Delta\alpha = 0.1$. The size of a control message was fixed at 900 bytes. Base link latency and throughput were varied independently. Network performance is summarized by the time average percentage of load served, i.e.,

$$\text{Performance} = \frac{1}{t_{end}} \int_0^{t_{end}} \frac{\text{power supply}}{\text{power demand}} dt$$

where t_{end} is the end of the experiment observation time (20 seconds in this case).

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

Table 1. 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

Table 2. Electrical demand schedule

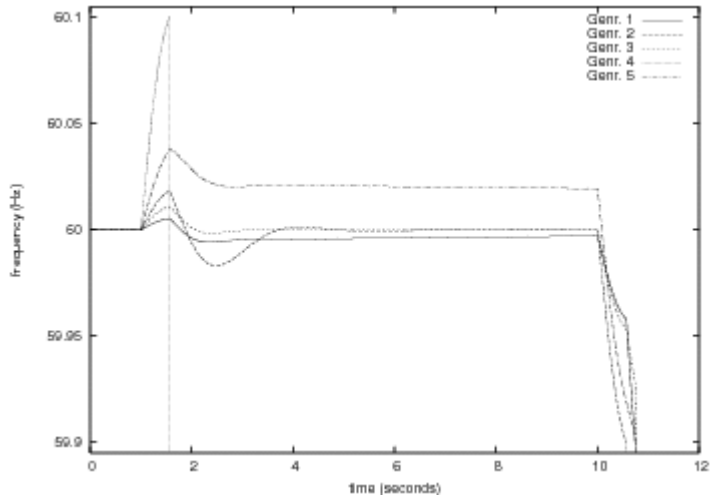

Figure 4. System failure in the absence of any load control scheme.

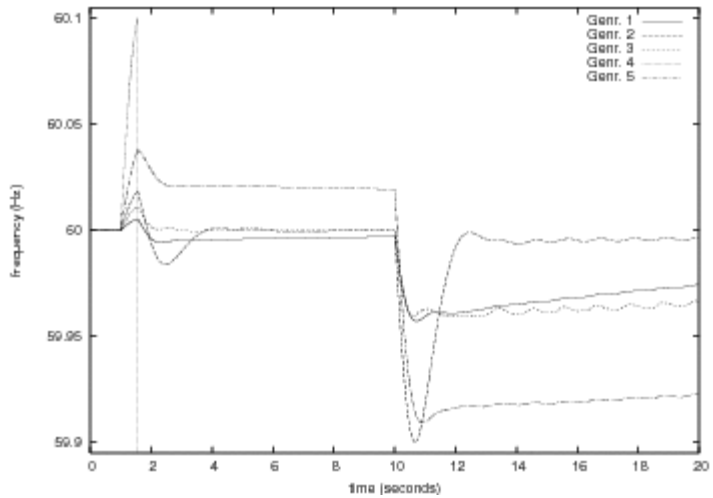

Figure 5. Generator frequencies when base line delay is 130ms and throughput is 1Mbps.

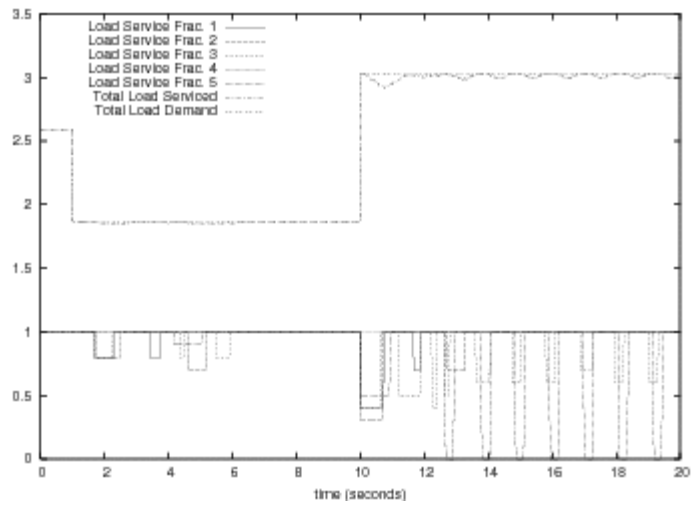

Figure 6. Total demand, served demand, and service fractions when base line delay is 130ms and throughput is 1Mbps.

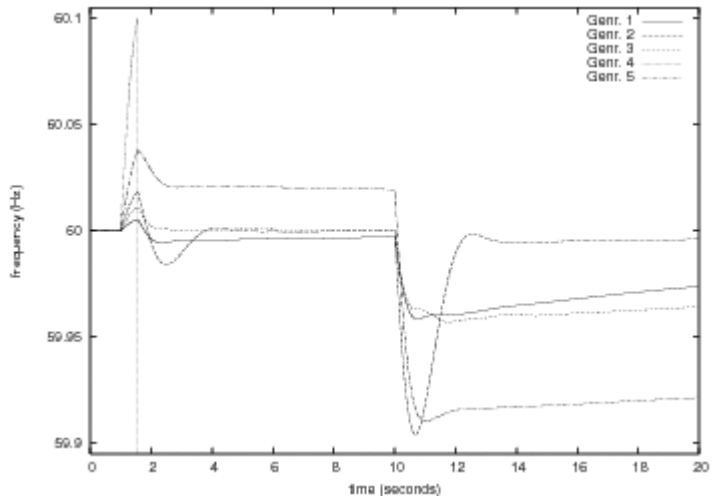

Figure 7. Generator frequencies when base line delay is 50ms and throughput is 1Mbps.

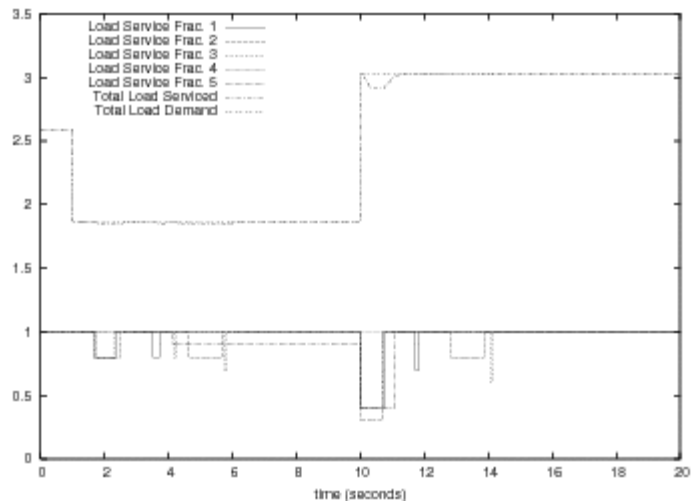

Figure 8. Total demand, served demand, and service fractions when base line delay is 50ms and throughput is 1Mbps.

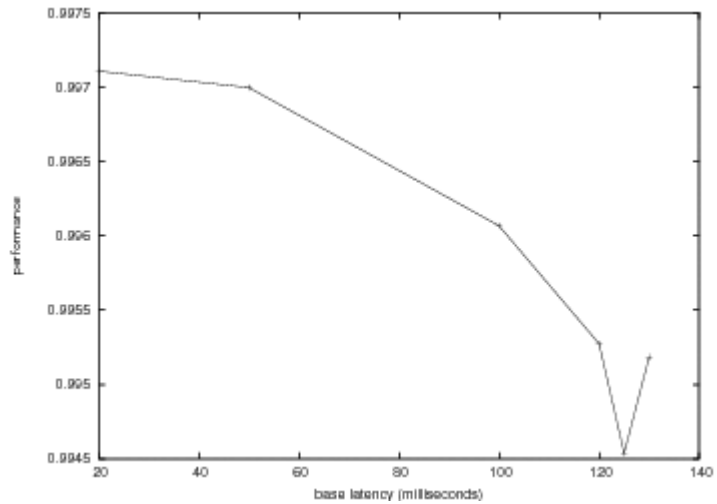

Figure 9. Performance as a function of latency with throughput fixed at 1Mbps.

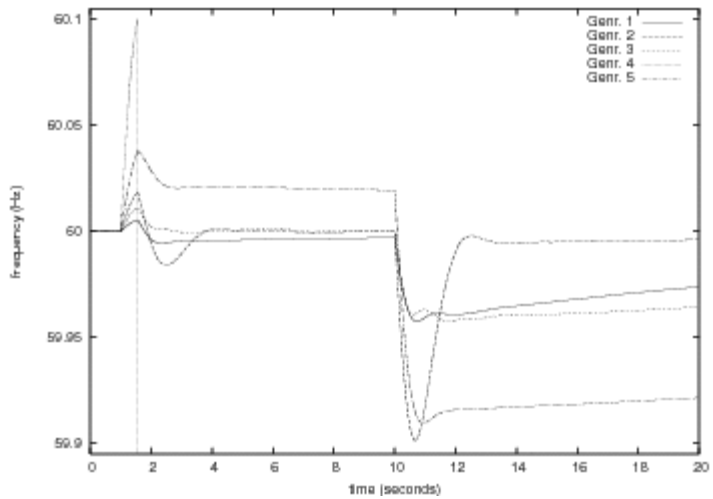

Figure 10. Generator frequencies when line throughput is 256Kbps and base latency is 20ms.

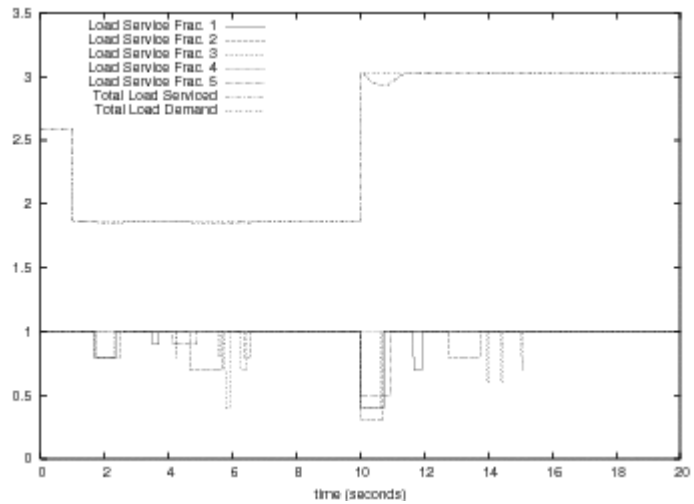

Figure 11. Total demand, served demand, and service fractions when line throughput is 256Kbps and base line delay is 20ms.

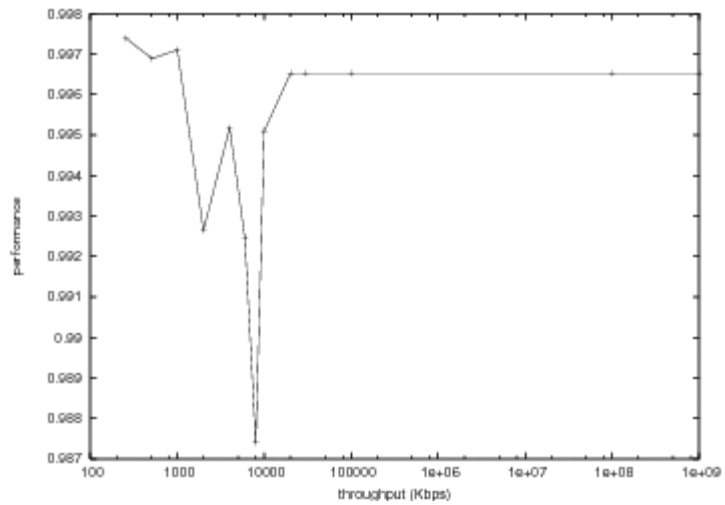

Figure 12. Performance as a function of throughput with base latency fixed at 20ms.

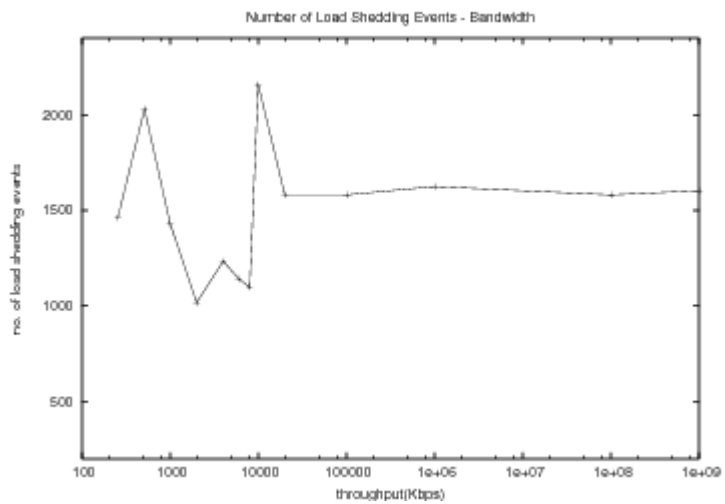

Figure 13. Number of load shedding events as a function of throughput with base latency fixed at 20ms.

The electrical demand schedule that was used in these experiments is shown in Table 2. 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.

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 that is caused by oscillations of the load service fraction at generators three and five. These ripples disappear and the system reaches a stable state at 50ms of latency, as shown in Figs. 7 and 8.

Figure 9 shows how the controller performance varies as a function of base line latency. The controller performance 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 fractions, and electrical supply and demand totals for the 256Kbps scenario are shown in Fig. 10 and 11. The system behaves nicely in this case, with the served load meeting demand approximately 5 seconds after the load spike at $t = 10$. 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 affects 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 it is located in the network. The combined effect on overall system behavior would be difficult to anticipate without an integrated, dynamic model of the power, control, and communication systems.

5. Conclusions

Our simulation framework description and case-studies offer several lessons learned which we outline below:

5.1 Technology Aspects

- Significant gaps exists in the simulation technology space that shed light on the communications impacts on power system behavior. Most simulation environments either focus only on the power system or only on the communication network. We find that that the DRACS functionality needs an integrated simulation of the communication network(digital) and the power system network (analog). The integrated approach alone can correctly relate the delays in the message with the control timing latencies.

- Statistical aggregation approach are effective but estimates (for operators) will need to be validated with monitoring signals during operation
  There are emerging recommendations to monitor the system for Demand Resources status. With statistical estimation, although the appropriate levels of confidence limits may be achieved, the reliability of the power system will require sampling of the set of demand resources. Thus there is an aggregation of the probabilistic estimate of dispatch, and an estimate of the actual system status by sampling. More pervasive sampling, possibly integrated with emerging metering infrastructures will fill any confidence gaps. The confidence gaps in the interim must be maintained within the margin of tolerance for reliability.
- Delay vs. Control tradeoff differs for different technology mix for dispatch. Integrating technology alternatives in one bag for dispatch will need a lot of simulation to get confidence.
  Current simulations abstract load behavior. These will need to be refined according to the technologies dispatched. The delays in obtaining the response (to, e.g., Time-Of-Day pricing related thermostat settings vs. critical peak price signals) affect the stability of the system. The dispatch strategy must simulate the response time based on the technology dispatched.

5.2 Policy/Rules Implementation

- Dispatch request hierarchy from ISO to IOU needs clarification as a module in DRMS.
  The location of operation of a DRACS system affects where in the distribution hierarchy dispatch decisions must be made. The CAISO may obtain high-level needs from the simulation to request dispatch (e.g., for peak shaving). These will need to be translated to requests to IOU's who can make their individual decisions on the precise dispatch approach. DRACS operating in the CAISO context simulates the state space at a different granularity than if it were operating within a utility. Emerging proposals for proxy and dynamic demand response resources further diversify the mode of obtaining system relief through demand response.

- Human behavior aggregation will need to be specified (at ISO and at Load Aggregation Point level).
  The simulation for expected dispatch requires statistical means to gain confidence in the outcomes. Aggregation granularity is a key requirement to be able to allow for variability in human behavior factors discussed below. For instance, a sub-division or metropolitan area aggregation has a focused dispatch strategy. On the other hand a dispersed strategy needs a hierarchical aggregation structure with confidence limits computed at each layer of the hierarchy.

- Rule-bases for dispatch decisions will need to be created by a team of experts reviewing the individual territory and IOU policies. These rule-bases will vary from area to area and region to region. Although the offline and online simulation framework provides a harness for the DRACS operation, population of candidate rule-sets is a precursor step that the IOU operations will have to experiment with (using the offline analysis capability) before any deployment tests are performed

6. References

[1] Edison SmartConnect – Open Innovation Use-Case Descriptions. http://www.sce.com/PowerandEnvironment/smartconnect/open-innovation/usecasechart.htm, D1 Use case.

[2] Edison SmartConnect – Open Innovation Use-Case Descriptions. http://www.sce.com/PowerandEnvironment/smartconnect/open-innovation/usecasechart.htm, C1 Use case.

[3] Edison SmartConnect – Open Innovation Use-Case Descriptions. http://www.sce.com/PowerandEnvironment/smartconnect/open-innovation/usecasechart.htm, E1 Use case.

[4] DRACS Requirements Use Cases and Requirements, *EnerNex Corporation,* January 2009

[5] DRACS Simulator Requirements, *EnerNex Corporation,* May 2009

[6] Demand Response Architecture and Control System Reference Architecture and Design, *EnerNex Corporation*, January 2009

[7] J. Nutaro, T. Kuruganti, L. Miller, S. Mullen, and M. Shankar, Integrated Hybrid-Simulation of Electric Power and Communications Systems, Nutaro, J.; Kuruganti, P.T.; Miller, L.; Mullen, S.; Shankar, M., IEEE Power Engineering Society General Meeting, 2007, Tampa FL, 24-28 June 2007, Digital Object Identifier 10.1109/PES.2007.386202

[8] E. Lightner and D.P. Chassin, GridLAB-D: The next revolution in power systems simulation, Pacific Northwest National Laboratory.

[9] J. Grainger and W. Stevenson, Jr., 1994. Power system analysis. McGraw, Hill Inc., New York, New York.

[10] W.H. Kersting, 2007. Distribution system modeling and analysis. Second Edition. CRC Press, Boca Raton, Florida.

[11] P. Mellor, D. Roberts, and D. Turner, Lumped parameter thermal model for electrical machines of TEFC design, *Proc. IEE Electric Power Applications*, vol. 138, no. 5, pp. 205–218, 1991.

[12] Milano, F. Power Systems Analysis Toolbox (PSAT), IEEE Transactions on Volume 20, Issue 3, Aug. 2005 Page(s): 1199 – 1206, Digital Object Identifier 10.1109/TPWRS.2005.851911

[13] 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 components," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 548–558, May 2006.

[14] NS-2, The network simulator. http://www.isi.edu/nsnam/ns

[15] *Adevs,* Nutaro, J. ADEVS website. Available via http://www.ece.arizona.edu/~nutaro/. Accessed on May 27, 2003.