Real-Time Dynamic Security Assessment

Fast Simulation and Modeling Applied to Emergency Outage Security of the Electric Grid

by Robert Schainker, Peter Miller, Wadad Dubbelday, Peter Hirsch, and Guorui Zhang

THE ELECTRICAL GRID CHANGES constantly with generation plants coming online or off-line as required to meet diurnal electrical demand, and with transmission lines coming online or off-line due to transmission outage events or according to maintenance schedules. In state-of-the-art electric utility control centers (illustrative example shown in Figure 1), grid operators use energy management systems (EMSs) to perform network and load monitoring, perform necessary grid control actions, and to manage grid power flows within its terroritory or region of responsibility.

Limits to flows and voltages on the transmission system are assigned on the basis of transmission line thermal limits and/or off-line studies of voltage and transient dynamic stability. Power flow limits for each transmission line determined in these off-line studies are, by design, conservative, since system operators must always maintain the security and economic operation of their

1540-7977/06/$20.00©2006 IEEE

power system over a wide range of operating conditions. Also, the assumption that the grid power flows settle down to steady-state condition is reexamined in real time as the transmission grid conditions change in real time.

Dynamic security assessment (DSA) software analyses allow for the study of the transient and dynamic responses to a large number of potential system disturbances (contingencies) in a transient time frame, which is normally up to about 10 s after a disturbance/outage. Currently, these analyses are performed off line since the simulation process takes hours of computer time to complete for a typically large grid network, which must be simulated for each condition of a large set of all possible outage conditions that could occur. The current, long simulation time makes DSA calculations impractical for use in a real-time application, wherein an operator would need to perform real-world control actions within tens of minutes after a real-world outage to be sure that the grid will not go into an unstable voltage instability and/or a cascading blackout condition.

figure 1. Illustrative example of a state-of-the-art electric grid energy management system.

Therefore, if the DSA calculation could be completed in less than about 10 min, operators who control the grid during emergency conditions (terrorist induced or "nature" induced) can indeed have sufficient time to take appropriate corrective or preventive control actions to handle the identified critial events, which may cause grid instability, or cause cascading outages that would severely impact their utility grid region or their neighboring utility regions, which would potentially avert billion dollar expenses associated with regional blackouts.

The work that lead up to this article was motivated by the attempt to dramatically reduce the time for DSA calculations so that DSA analyses can be converted from offline studies to routine, online use in order to aid grid operators in their real-time controller analyses. The large amount of time for DSA calculations occurs because grid transients for a large grid network must be calculated over about a 10 s time interval and properly represent a large interconnected power system network system, which must properly represent detailed static and/or dynamic models of power system components, such as transmission network solid-state flexible ac transmission system devices, all types of generators, power system stabilizers, various types of relays/protection systems, load models, and various types of faults or disturbances.

This article describes the methods and successful results obtained in developing a real-time version of the DSA tool. The material below is organized by first providing a description of the DSA software package generally used by the U.S. electric utility industry. Then discussed is a way to dramatically reduce the computation time to perform DSA calculations, which, among other useful techniques, uses a new distributed computational architecture. The results from applying this new version of DSA are then presented using a large utility system as an example. The results clearly show that, indeed, using the new DSA approach, calculations for a large power system can be performed fast enough for the real-time application to EMSs that operate today's grid systems. The article then ends with some insights and concluding remarks.

Dynamic Security Assessment (DSA)

DSA software performs simulations of the impact of potential electric grid fault conditions for a preset time frame after a potential grid disturbance, usually over a time interval of 5–10 s after an outage contingency condition occurs. Contingency conditions studied include "normal" transmission line and/or power plant outages caused by acts of nature or equipment (e.g., outages due to lightning and/or generator "trips"), wear and tear (for example, equipment age failures), and outage conditions caused by human error and/or potential terrorist-induced equipment failures.

Recent efforts by the authors of this article have focused on improving the performance of the DSA calculation process with the eventual goal of implementing the DSA evaluation process in an online utility energy management system (EMS). Past DSA research projects have resulted in significant achievements in determining which outage contingency conditions are significant and not significant by rapidly separating the outage contingencies into "definitely safe" and "potentially harmful" groups. The "potentially harmful" group must be studied in more detail to accurately determine whether a "potentially harmful" contingency is in fact harmful.

Dynamic Security Assessment Models

The DSA program uses a complete representation of all the generators (for example, fossil, nuclear, gas, oil, hydro, and wind generators) including their exciters, governors and stabilizers, transmission lines and many other linear and nonlinear components. For example, nonlinear devices embedded into DSA software include such items as:

1) synchronous machines
2) induction motors
3) static VAR compensators

DSA models each transmission line as a network impedance model with capacitance, inductance, and resistance.

4) thyristor-controlled series compensations
5) thyristor-controlled tap changers and/or phase regulators
6) thyristor-controlled braking resistors or braking capacitors
7) static load models (nonlinear loads)
8) high-voltage dc link
9) user-defined models, as appropriate.

In addition, DSA software models different types of electric grid protection relays:

1) load shedding relay
2) underfrequency load shedding relay
3) voltage difference load dropping relay
4) underfrequency generation rejection relay
5) underfrequency line tripping relay
6) impedance/default distance relay
7) series capacitor gap relay
8) rate of change of power relay.

Also modeled within the DSA software are static nonlinear load models, which are different from constant impedance load models.

DSA also models the following four types of static nonlinear loads:

1) constant current load
2) constant mega-voltage-ampere load
3) general exponential voltage and frequency-dependent load
4) thermostatically controlled load.

Additionally, DSA models each transmission line as a network impedance model with capacitance, inductance, and resistance. Each line also has thermal line rating limits. In addition, tap- and phase-shifting transformers are modeled.

Contingencies for DSA are defined in terms of the fault type, location, duration, and sequence of events making up a contingency scenario. Typical short-circuit faults are three-phase faults, single-line-to-ground and/or double-line-to-ground short-circuit faults. Automatic switching actions taken into account in the computation simulation are line removal or line closure into the grid network. The location of the short-circuit fault can be at the electrical bus, line end, or line section.

DSA Algorithm

The solution to all these devices operating in an electric grid requires solving a large set of differential equations. For a 5,000-node network with 300 generators, over 14,000 nonlinear differential equations must be simultaneously solved. DSA uses a numerical analysis method to solve these nonlinear differential equations. The numerical method uses a

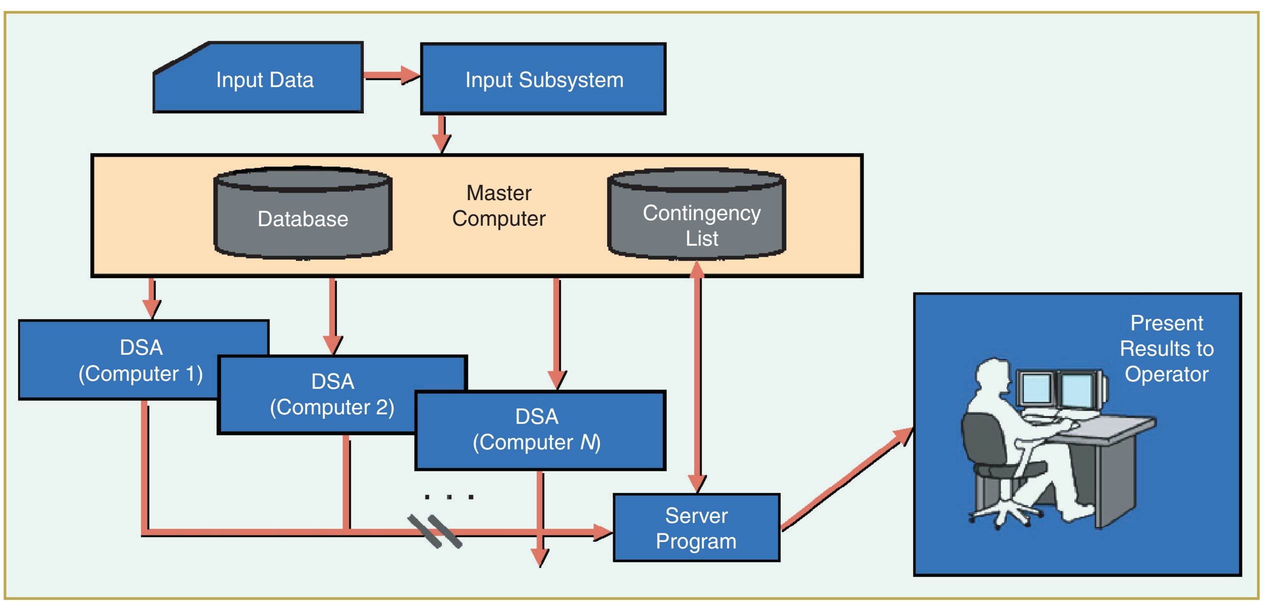

figure 2. A schematic of the distributed computer architecture used to improve the DSA computation time.

small time step of about 0.01 s, and at each time step, the method linearizes the equations to calculate the future time response. A classic Newton-Rahpson iteration approach is incorporated into the numerical method, and for a 10-s simulation, 1,000 such time steps are used.

The solution for a conventional transient stability program can take considerable time to solve for one contingency and even longer for multiple contingencies. Typically, for a 5,000-node network in which 300 contingencies are investigated, a 30-s transient stability simulation may take over two hours of calculation time, dependent on the type of computer used in the calculation.

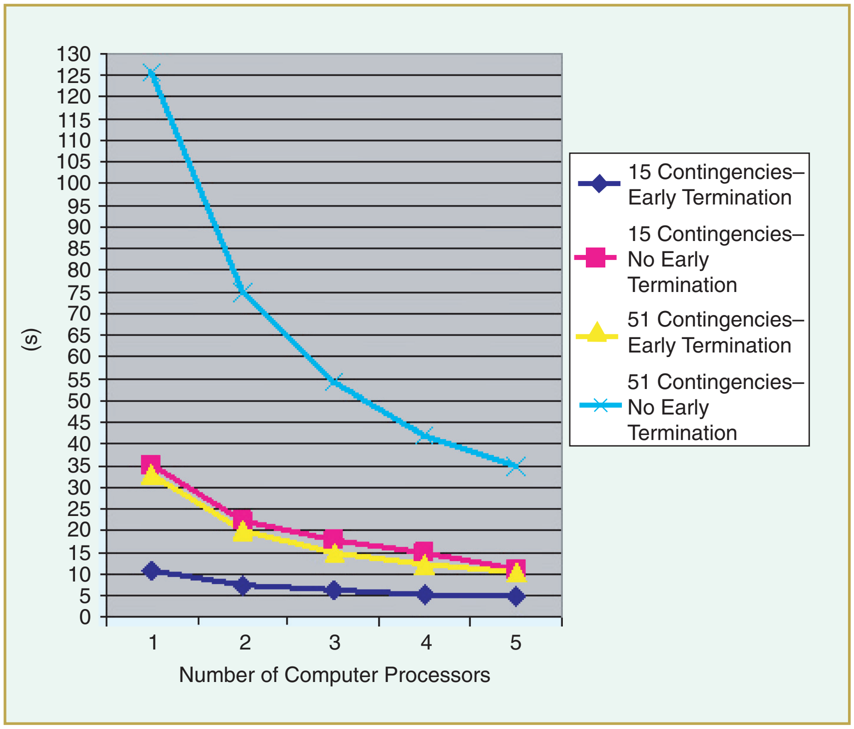

figure 3. The DSA computation time performance, with and without early termination method.

One would need over 100-fold improvement in DSA simulation time performance to be able to do this calculation in about 10 min or less.

Based on these requirements, some of the significant ways for improving the DSA performance deployed by the authors herein are described below.

1) An improved stopping (called *early termination*) criteria when evaluating each contingency is used to reduce the overall time each contingency is simulated. That is, if the program simulation is for 10 s and after a short time duration, say less than 2 s of simulation time, it can be determined that the contingency case being investigated is unstable or stable, then the DSA program evaluating that contingency is stopped, and a flag is set to unstable or stable for the contingency case being investigated. If no stable/unstable determination can be made, then the DSA program for that contingency case runs the full 10-s simulation time period specified. Using this technique, the DSA program does not have to be run to completion for every contingency. It is run to completion only for those contingencies that are moderately stable or moderately unstable.
2) A novel distributed computing architecture (see Figure 2) was also used to improve the time it takes to perform the numerous contingency cases investigated. In general, there are two ways of performing distributed computation, and both were investigated. One is to parallelize the DSA algorithm and its calculation approach, using central processing units (CPUs) in parallel to perform the calculations. This will improve the performance somewhat, but due to the sparse nature of the differential equation matrices involved, this improvement has been found to be not very useful. A better technique is to run the full DSA software application on each of n computers (set up to communicate with each other) and distribute the contingencies (so each computer runs a different set of contingencies). The master computer distributes the contingencies to each of the slave/server computers as needed. Of course, this will work as long as the number of contingencies is equal, or exceeds the number of computers, which is certainly the case. Full distributed computation is thus achieved and the only slow down is due to the use of one master computer to orchestrate/distribute the contingencies to the other computers and receive/catalogue the solution results from the other computers as the results become available.

Using the above methods, and others, the authors developed a new DSA computation architecture and approach, which did improve the computation time by a factor of about 100+, based on the following improvement components: an improvement factor of about 2, due to not having to move data among computers and hard disk storage locations, an improvement factor of about 3 by using the "stopping" criteria discussed above, an improvement factor of about 4 by using five computers in the distributed computer architecture discussed above, and an improvement factor of about 6 due to faster CPUs used to perform the calculations, as compared to those used circa 2000.

DSA Input Data

The DSA input data consist of three sets of data:

- ✔ the power flow data, which contain all the transmission line configurations, tap-changing transformers, phase-shifting transformers, load representation, electric breaker information, relay information, and the type and location of the generation plants

- ✔ The dynamic data, which contain various types of generator models, including the generator exciter models, governor models, power system stabilizers, the exciter models, governor models, power system stabilizers, basic generator parameters (along with their limits and time constants), load models, and protection relay models
- ✔ The contingency data, which include the various types of faults, including the type, location, duration, and clearance of the faults and the switching actions after faults are cleared.

DSA Output Data

The DSA program produces output results for each contingency and for each generator. The results are data and information on items such as the relative generator angle, the speed of the generator, and the voltage at each generator. This output is temporarily saved on the computer running the contingency and is then transferred to the master computer at the end of the contingency run. On the master computer, time-dependent plots for each contingency of the top three worst grid node response cases are made available to the user.

Figure 3 was produced using the DSA improvement methods described above. On the vertical axis of this figure is the computer run time needed to perform a DSA calculation for a utility grid system that has 5,839 electric buses, 11,680 transmission lines, and 779 generators. The computation time needed to run this large, representative utility test case with only the master computer and then, sequentially, with one, two, three, four, and the available portion of the master computer used as a "fifth" slave computer. Each point on the plots in the figure show the time required to do all the DSA calculations. Comparison data were plotted for cases where the number of contingencies was 15 and 51. Also, for comparison purposes, data were plotted for cases where the early termination logic was used for each contingency case computed and for when no early termination logic was used for each contingency case computed. The results were impressive showing a significant improvement in computing time. For the test case with 51 contingencies, the computing time ranged from 125 s (using only one computer) down to 35 s using all five computers (i.e., the master and the four slave computers). This set of runs showed an improvement factor of about 3.6 in computing time. For the test case with 15 contingencies, the computer run time ranged from 35 s (using only one computer) down to 10.3 s using all five computers (i.e., the master and the four slave computers). This set of runs showed an improvement factor

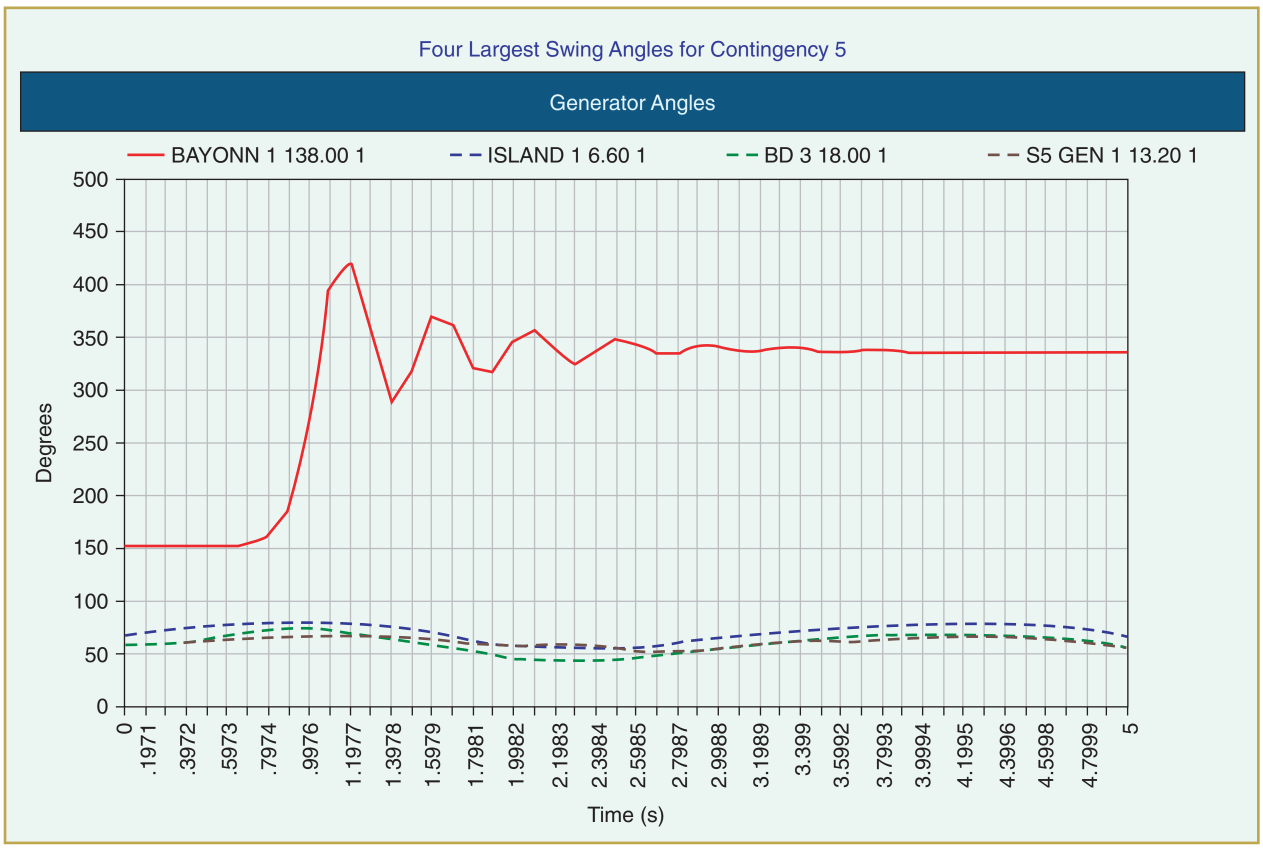

figure 4. The DSA output plot for the largest generator swing angle for a stable contingency case.

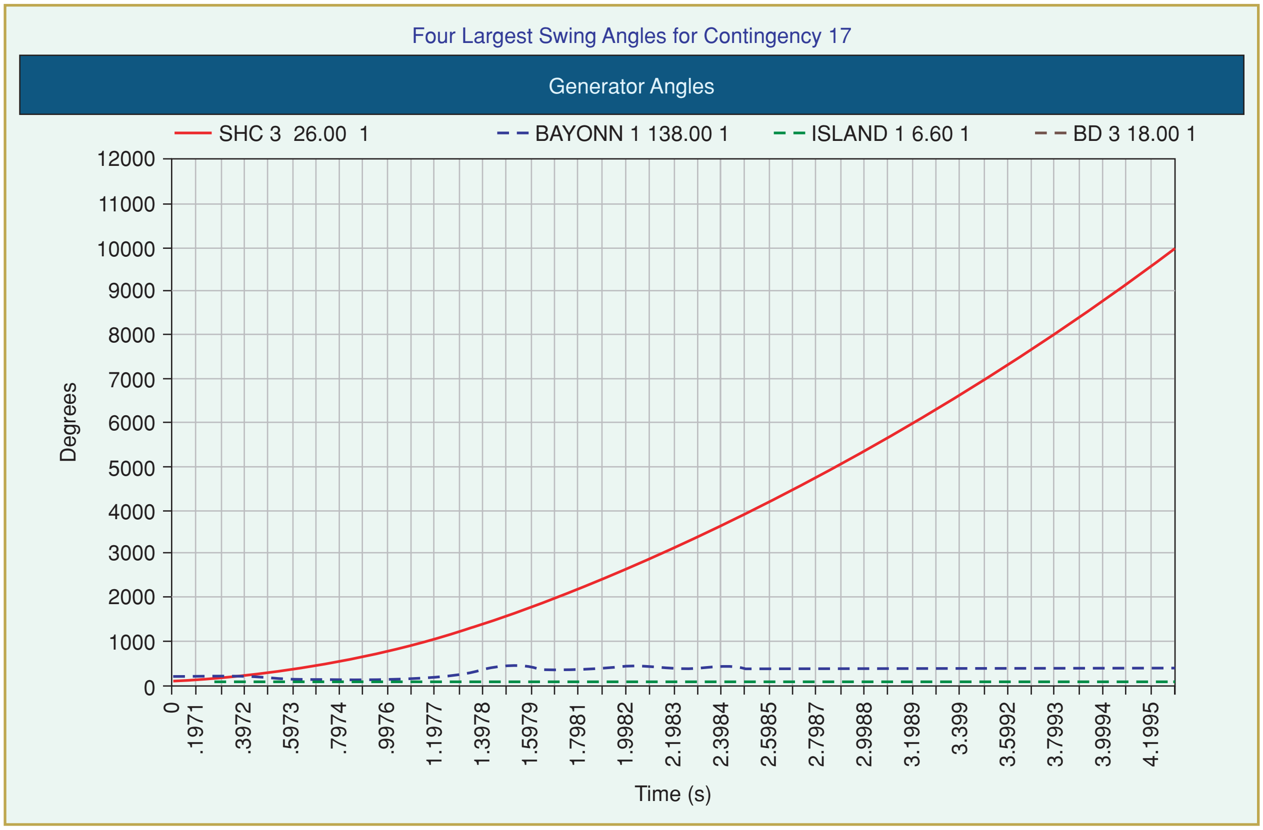

figure 5. The DSA output plot for the largest generator swing angle for an unstable contingency case.

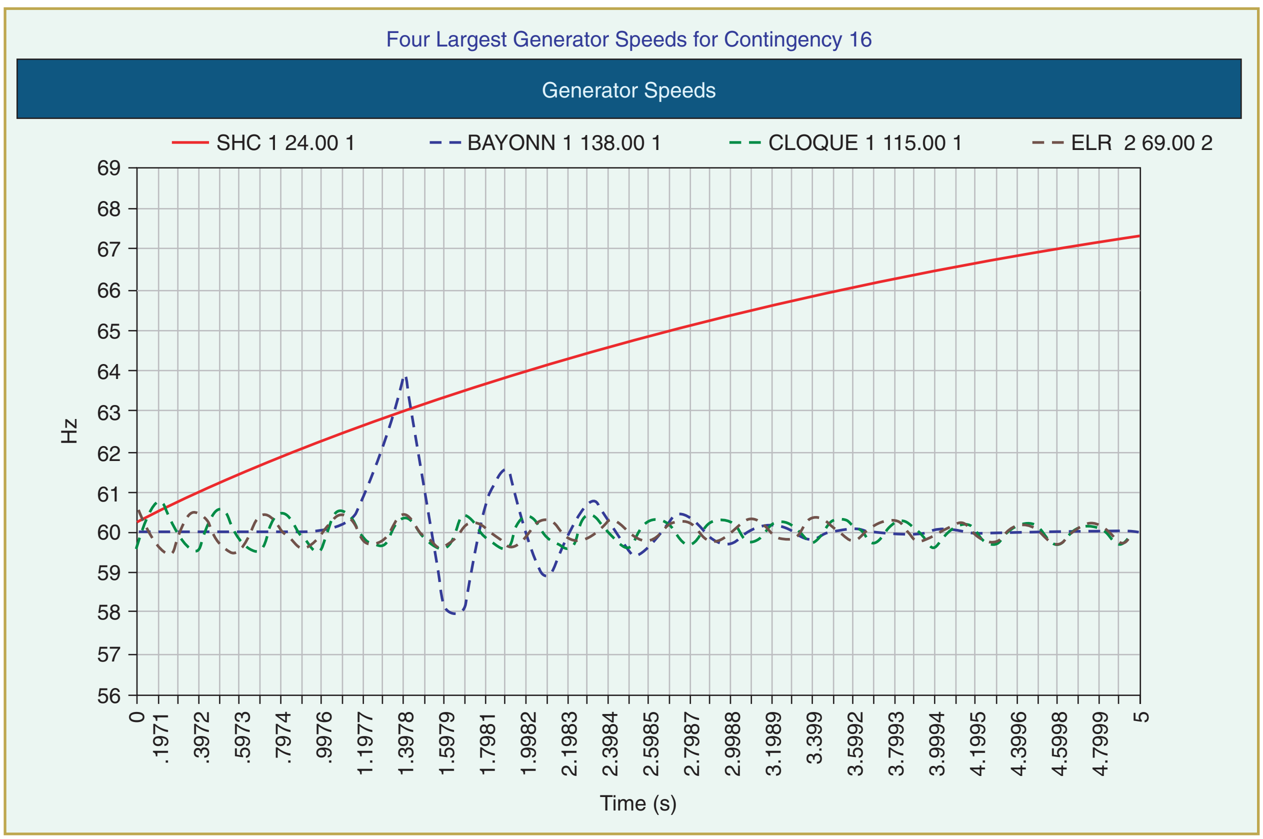

figure 6. DSA output plot for largest generator speeds for an unstable case.

of about 3.4 in computer run time. These results clearly show the power of the master-slave computer architecture developed and successfully investigated and tested.

A number of DSA algorithm improvements were also investigated. The most effective one investigated was the "early termination" method. Sample results are also shown in Figure 3. Using the early termination method and comparing it to the "no early termination" method, for the test case with 51 contingencies, the computer run time improved from 125 s to 33 s (using only one computer) and from 35 s to 10 s (using all five computers, i.e., the master and the four slave computers). This set of runs showed about an improvement factor of 3.8 to 3.5 in computer run time. For the test case with 15 contingencies, the computer run time improved from 35 s to 11 s (using only one computer) and from 12 s to 5 s (using all five computers—i.e., the master and the four slave computers). This set of runs showed an improvement factor of about 3.2 to 2.4 in computer run time. These results also clearly show the power of the master-slave computer architecture system developed and successfully tested.

DSA Graphical Output Displays

The DSA program provides several graphical output displays to show the following types of output results (some of which are illustrated in Figures 4–7):

- ✔ largest generator speed angles, for both stable and unstable contingency cases
- ✔ highest frequencies, for both stable and unstable contingency cases.

Conclusions

Using the distributed computer architecture for DSA calculations, grid operators can now quickly analyze a large number of system contingency outage events. Thus, they can evaluate the appropriate preventive or corrective control actions to effectively handle various severe system disturbances or even mitigate costly cascading blackouts, events that are either initiated by nature or terrorist induced.

Online dynamic security analysis (DSA) requires extensive computer resources, particularly for large electric power systems. With the recent advances in computer technology and the intra- and interenterprise communication networking, it now becomes cost-effective and possible to apply distributed computing to online DSA in order to meet real-time performance requirements needed in the electric utility industry.

Thus, the major conclusions of the work presented herein are:

- ✔ The distributed computing architecture to perform the dynamic security assessment (DSA) analysis of a large interconnected power system with a large

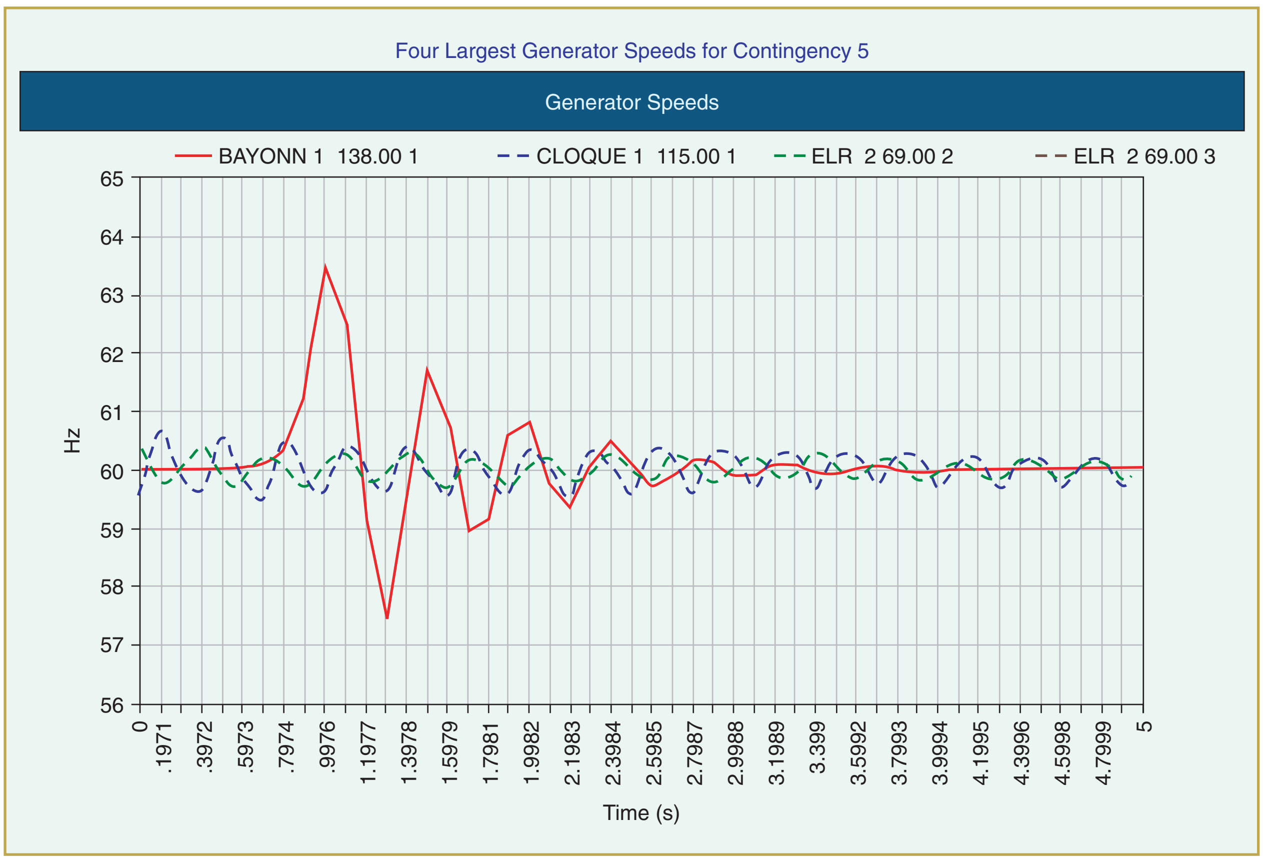

figure 7. DSA output plot for largest generator speed display for stable case.

Contingencies for DSA are defined in terms of the fault type, location, duration, and sequence of events making up a contingency scenario.

number of contingencies has been demonstrated to be extremely fast. As such, this computerized approach should be implemented for real-time decision-making conditions, which are faced by utility and grid operators when any unplanned outage condition occurs that might lead to system instability or even cascading blackout conditions.

- ✔ The distributed computer approach developed was tested successfully with five computers in a master-slave arrangement that is scalable to any number of extra slave computers.
- ✔ The dynamic security analysis (DSA) using distributed computing can be fully integrated with utility operator EMSs using real-time operating conditions and grid State Estimation estimators.
- ✔ The dynamic security analysis (DSA) using distributed computing can also be used for performing operational planning studies for large power systems.
- ✔ The dynamic security analysis (DSA) using the distributed computing technology presented herein used the Oracle 9i relational database and its related software. This enables flexible software integration with a wide variety of IT infrastructure systems currently used by many electric utilities and/or grid operators.
- ✔ The proposed approach can be used to better utilize existing computer resources and communication networks of electric utilities. This will significantly improve the performance of DSA computations electric utilities perform routinely.
- ✔ The performance of the DSA approach presented herein is also fast enough for the real-time calculation of the interface transfer limits using real-time operating conditions for large interconnected power systems.

Acknowledgments
The material presented in this paper was sponsored by the Department of Homeland through a Space and Naval Warfare Systems Center, San Diego, Contract N66001-04-C-0076.

For Further Reading
"Analytical methods for contingency selection and ranking for dynamic security analysis," EPRI, Palo Alto, CA, TR-104352, Project 3103-03 Final Rep., Sep. 1994.

"Simulation program for on-line dynamic security Assessment," EPRI, Palo Alto, CA, TR-109751, Jan. 1998.

"Standard test cases for dynamic security assessment," EPRI, Palo Alto, CA, TR-105885, Dec. 1995.

A.A. Fouad and V. Vittal, *Power System Transient Stability Analysis Using the Transient Energy Function Method*. Englewood Cliffs, NJ: Prentice Hall 1992.

C.K. Tang, C.E. Graham, M. El-Kady, and R.T.H. Alden, "Transient stability index from conventional time domain simulation," *IEEE Trans. Power Syst.*, vol. 9, no. 3, Aug. 1993.

G.D. Irisarri, G.C. Ejebe, W.F. Tinney, and J.G. Waight, "Efficient computation of equilibrium points for transient energy analysis," *IEEE Trans. Power Syst.*, vol. 9, no. 2, May 1994.

Biographies
Robert Schainker is a technical executive and manager of the Security Program Department at the Electric Power Research Institute. He received his D.Sc. in applied mathematics and control systems, his M.S. in electrical engineering, and his B.S. in mechanical engineering at Washington University in St. Louis, Missouri.

Peter Miller is program manager, Homeland Security Advance Research Project Agency, Mission Support Office, Science and Technology, U.S. Department of Homeland Security. He holds a B.S. in mathematics from the City University of New York, where he received the Borden Prize, and he holds an S.M. in computer science and electrical engineering from the Massachusetts Institute of Technology.

Wadad Dubbelday is an engineer at the Navy at the Space and Naval Warfare Systems Command (SPAWAR) office in San Diego, California. She holds a B.S. in physics from the Florida Institute of Technology and holds M.S. and Ph.D. degrees in electrical engineering-applied physics from the University of California at San Diego.

Peter Hirsch is a project manager in the Power Systems Assets Planning and Operations Department of the Electric Power Research Institute. He also is the manager of the software quality group within EPRI's Power Delivery and Markets Sector. He holds a B.S. in applied mathematics and engineering physics and M.S. and Ph.D. degrees in mathematics from the University of Wisconsin. He is a Senior Member of the IEEE.

Guorui Zhang is principle engineer at EPRI-Solutions, a subsidiary of the Electric Power Research Institute. He received his B.S. in computer software engineering at Singh University, China; he received his Ph.D. in electrical engineering at the University of Manchester Institute of Science and Technology, Manchester, England. He is a Senior Member of the IEEE. p&e