A User-Friendly Simulation Program for Teaching Power System Operations

Thomas J. Overbye, Member   Peter W. Sauer, Fellow   Caroline M. Marzinzik, Student Member   George Gross, Fellow

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

Abstract

This paper describes a user-friendly power simulation program for teaching power system operations and control. The motivation for the development of the program is to provide students with a simple and useable tool for gaining an intuitive feel for power system operations. This Microsoft Windows™ based program simulates power system operation over a specified time period (of typically several hours to a day). Students dynamically interact with the simulated system through various windows, including a one-line display. The use of data files and option windows allows the program to simulate a wide variety of different power systems and operating problems. Efficient programming results in a modest program size of just 450k, allowing for easy distribution to students.

Introduction

As power systems become increasingly complex, there is a critical need to make available improved tools for training power system engineers and analysts. Traditionally this training has been provided primarily to student engineers, system operators, and practicing engineers. However, a worldwide trend towards increasingly more open transmission systems, the growing competition in electricity markets, the changing role of regulation, and the entry of new working professionals into the electricity business has created an unprecedented need to provide personnel without an extensive technical background with at least a basic understanding of power system operations. This is required in order to ensure that the interactions between business decisions and the technical/physical constraints of the power system are appropriately considered and understood. In particular, these people need an understanding of the effects of various power and energy transactions on system operation, and the constraints often imposed by the transmission system.

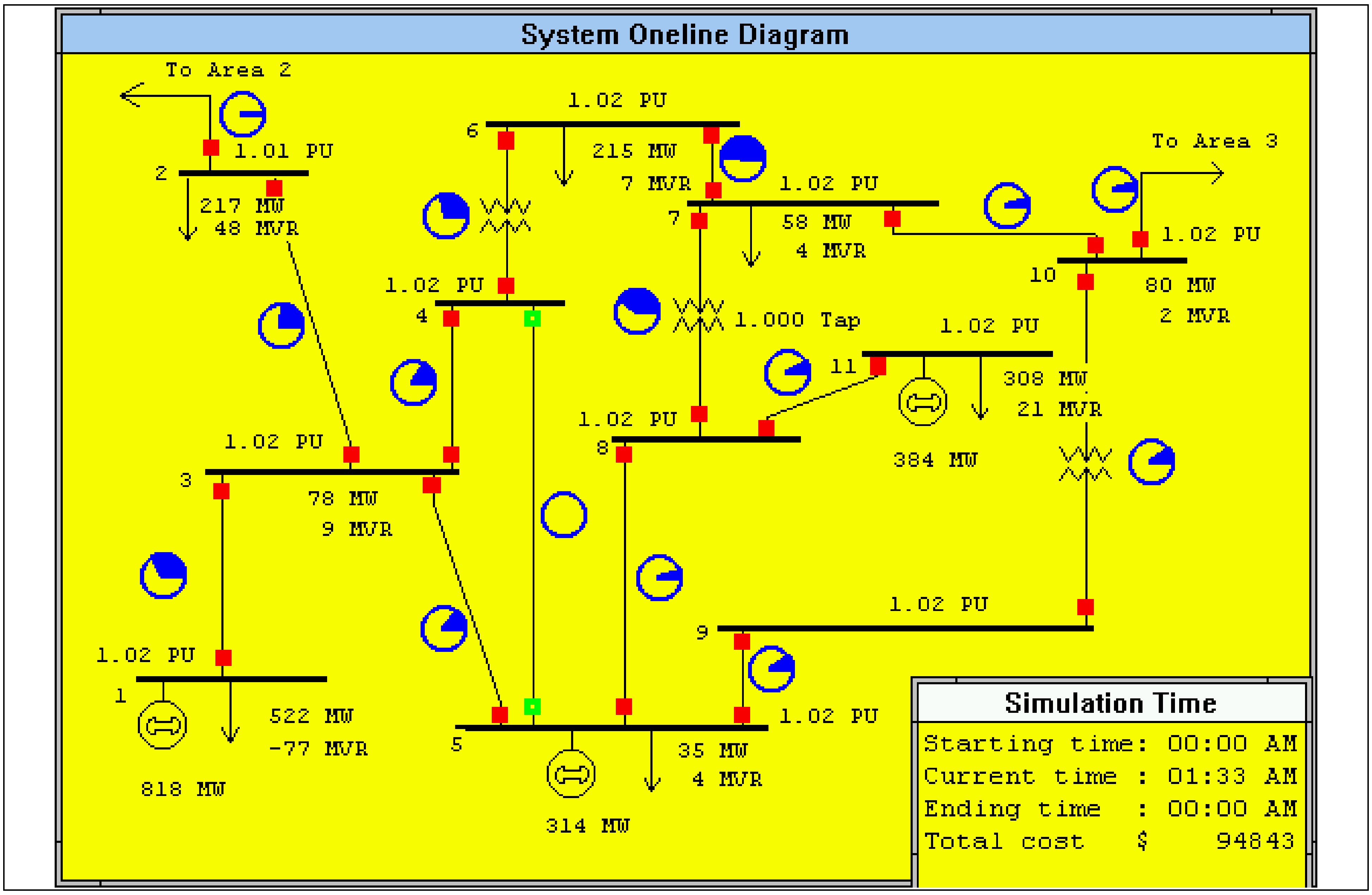
Over the years computer simulation programs have played an important role in providing students with a better understanding of power system operation. As a result of the rapid advances in computer hardware and software, computer-based power system educational tools have grown from very simple implementations, providing the user with little more than a stream of numerical output, to very detailed representations of the power system with an extensive graphical user interface (GUI). Examples of such programs include dispatcher training simulators (DTS) for operator instruction [1-4], along with a wide variety of different simulation programs targeted for university use [5- 10].

The motivation for the development of the program described in this paper was to have a self-contained power system simulation program for instruction both of engineering undergraduates and of personnel with a nontechnical background involved in the electricity business. This required the use of a user-friendly GUI so the students could spend their time gaining an intuitive feel for operating an electrical system, rather than just learning how to use the program. Basic concepts needed to be presented simply, yet the program required sufficient detail to maintain the interest of and provide challenge to the advanced engineering student. The principal thrusts of the program include basic power flow in a network, how system controls (such as generator MW/voltage setpoint, LTC transformers, line outages, and power transactions) affect power flow, and area control concepts such as area control error (ACE), automatic generation control (AGC) and economic dispatch (ED).

Effective application of object-oriented programming (OOP) techniques using Borland Pascal [11] in a Microsoft Windows™ 3.1 environment produced a package which has modest system requirements. The program executable and associated data files are small enough to allow distribution on a single floppy disk. The size of the program executable is just 450k; the size of the data files is, of course, dependent upon the size of the simulated system. As an example, a 13 bus system requires about 30k. To run the program, the minimum recommended system requirements are a PC with a 486 or better processor, a VGA or better display adapter, a Windows-compatible mouse, and Windows 3.1. The program was developed for use in the Edison Electric Institute (EEI) Power Systems Planning and Operations School created and run by the University of Illinois at Urbana-Champaign (UIUC) and the University of California at Berkeley. The program is also used in the power system classes at UIUC.

Program Features

In general, the program simulates the operation of an operating area in an interconnected power system over a given period of time (usually from several hours to a day). During



Simulation Time

```
Starting time: 00:00 AM
Current time : 01:33 AM
Ending time  : 00:00 AM
Total cost   $    94843
```

Figure 1: One-Line Diagram of a Sample Thirteen Bus System

the course of the simulation the loads will usually change automatically, and, optionally, different types of random events can occur (such as equipment outages). The student operates the system primarily by using a dynamic one-line diagram to change system controls including generator MW outputs, LTC transformer tap positions or circuit breaker status. As an example, Figure 1 shows the one-line diagram for a thirteen bus system. Other windows allow the student to interchange power from other (external) areas, to further monitor the system, and to change various options. The student is responsible for operating the system at minimum cost while maintaining correct area interchange, and ensuring that system voltages and flows are within their limits. Different data cases can be used to simulate different systems. A student can start operating just a two bus system and then work up to increasingly larger systems.

Simulation Solution Cycle

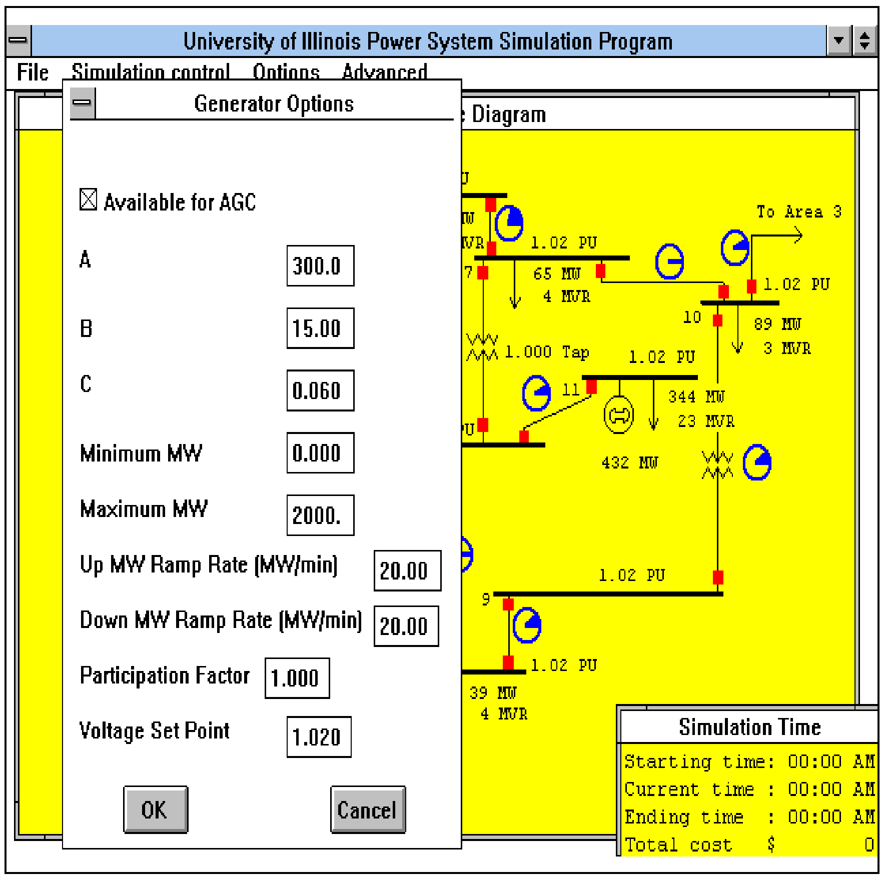
Because of the emphasis on longer time frame transmission system concepts, the key component of the simulation is the power flow solution. The basic steps of the simulation are the following:

1. Read simulation case and one-line diagram information from disk files, set simulation time to start time and perform initial power flow solution.
2. While simulation time <= end time Do
    a. Change load, perform any scheduled events and, on an optional basis, introduce stochastic events.
    b. Check for limit violations and outage any lines that have reached their maximum thermal limits.
    c. Perform network topology processing.
    d. Set scheduled transactions and do area control calculations, including (optionally) ED.
    e. Perform power flow solution.
    f. Update simulation displays, clock, and total cost.
3. Stop the simulation.

To begin a simulation the mouse is used to select a case from disk with a file dialog box. The one-line diagram and simulation time windows are then displayed along with a menu bar. At any time during the simulation, the student can interact with the case (such as by changing controls or setting up MW transactions) through the various windows.

System One-Line Diagram

The one-line diagram is the most important part of the graphical user interface (GUI). The one-line representation, like all of the other windows in the program, has been designed using object-oriented techniques. The advantages of this approach include flexibility to allow the user to interact with all objects on the screen, small source and executable files, extensibility and reusability of existing code, and a user-friendly GUI. When the program is running the one-line is



Generator Options

☒ Available for AGC

A 300.0

B 15.00

C 0.060

Minimum MW 0.000

Maximum MW 2000.

Up MW Ramp Rate (MW/min) 20.00

Down MW Ramp Rate (MW/min) 20.00

Participation Factor 1.000

Voltage Set Point 1.020

Simulation Time

```
Starting time: 00:00 AM
Current time : 00:00 AM
Ending time  : 00:00 AM
Total cost   $       0
```

Figure 2: Sample Generator Dialog Box

continually updated with bitmap copies used to provide a "smooth" animation. The display update rate depends upon the computer's speed and size of the simulation, with several times per second typical with a relatively fast PC, a small system (less than 20 buses) and 1024 by 768 display resolution.

The one-line uses graphical symbols to represent system equipment such as buses, generators, loads, transmission lines and transformers. A combination of numeric values and graphical symbols are used to show system values. For example in Figure 1 numeric values are used to show bus voltage magnitudes, and load/generation real and reactive power values. However, graphical pie charts are used to show percentage line loading. As the line loading increases, the pie becomes increasingly full; for flows above the transmission line's limit the pie changes color from blue to red, with the percentage value then shown in the interior of the pie. Additionally, the rotor position changes to indicate the voltage angle at generators. Options are provided to show the voltage magnitudes in either per unit or actual kV, and to "declutter" the display by removing different classes of values (such as all line flows).

The user can control the simulation by clicking the mouse on different components on the one-line. For example circuit breakers can be opened by positioning the cursor on the circuit breaker symbol and clicking the right mouse button, while they are closed by clicking with the left button. Because of the fast display update rate, the new flows appear almost instantaneously. Similarly, the MW output of the generators is increased by positioning the cursor on the generator MW value and clicking with the left button (or with the right button to decrease the value). Transformer taps, generator set point voltages, and capacitance values can also be modified in a similar manner. By double clicking with the left button on the actual device symbol the user can view and sometimes modify different parameters associated with the device. For example, Figure 2 shows the dialog box obtained by double clicking on the generator symbol. All values in this dialog box can then be changed by the user. If desired, the user can also use the double border of the one-line window to change its size or position on the screen.

The OOP techniques of inheritance and polymorphism are effective in succinctly providing this display functionality. In OOP the object hierarchy defines the relationships between various objects, with descendant objects inheriting access to all its ancestor's code and data. Polymorphism is giving an action one name that is shared up and down an object hierarchy, with each object in the hierarchy implementing the action in a way appropriate to itself [11]. For example, polymorphism permits clicking the left mouse button on the breaker to open the device, while the same action on the generator MW value increases the generator's MW output. Figure 3 shows the program's object hierarchy. The object hierarchy can be easily modified to permit the incorporation of additional objects.

System Data Files

Each power system case has two associated data files. The first file contains the one-line diagram information. The one-line displays are currently built off-line using a separate Windows-based program. With this program the one-line diagram designer uses the mouse to position objects (such as a line pie chart) on the new representation and then uses a dialog box to enter the values of various parameters associated with the object (such as for the line pie chart the desired size, and the "from" bus, the "to" bus and the circuit number of the line flow associated with the pie chart). The program then stores this information in a non-ASCII format in a file. The time required to build the one-line diagram depends upon the system size and complexity. For reference purposes, the one-line diagram shown in Figure 1 required about an hour to build.

The second file is a text file that contains all the remaining information necessary to define the case. The actual power system information is defined in the beginning of this file using the IEEE common data format [12]. The remainder of the file then contains various simulation options and more detailed device modeling information. For example, to simulate the daily load variation, the real and reactive load at each bus is modeled as a piecewise linear function of time. Additionally, specific events such as removing/inserting lines can

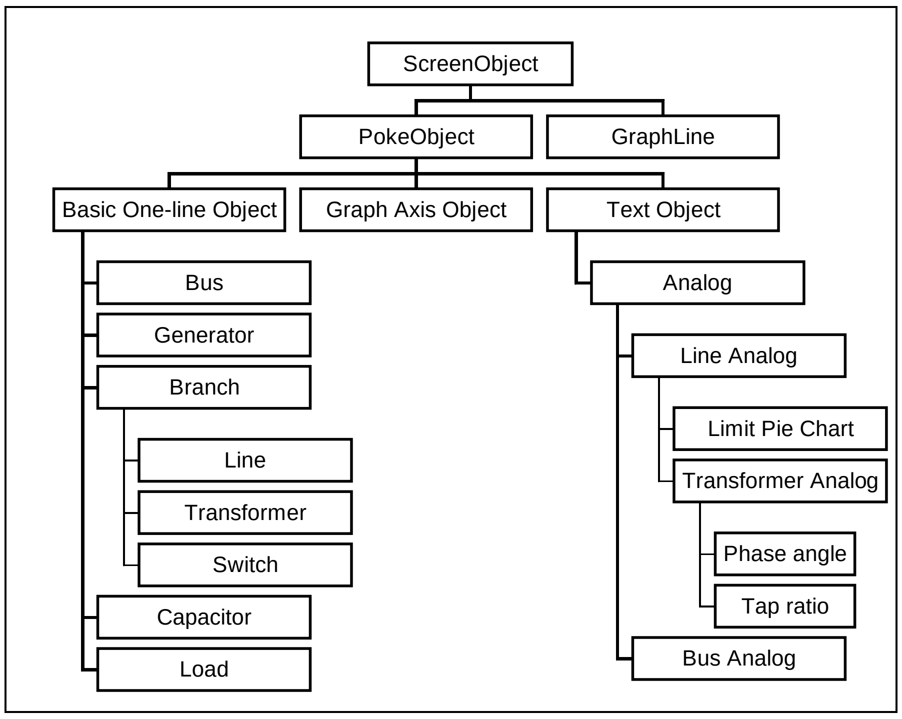


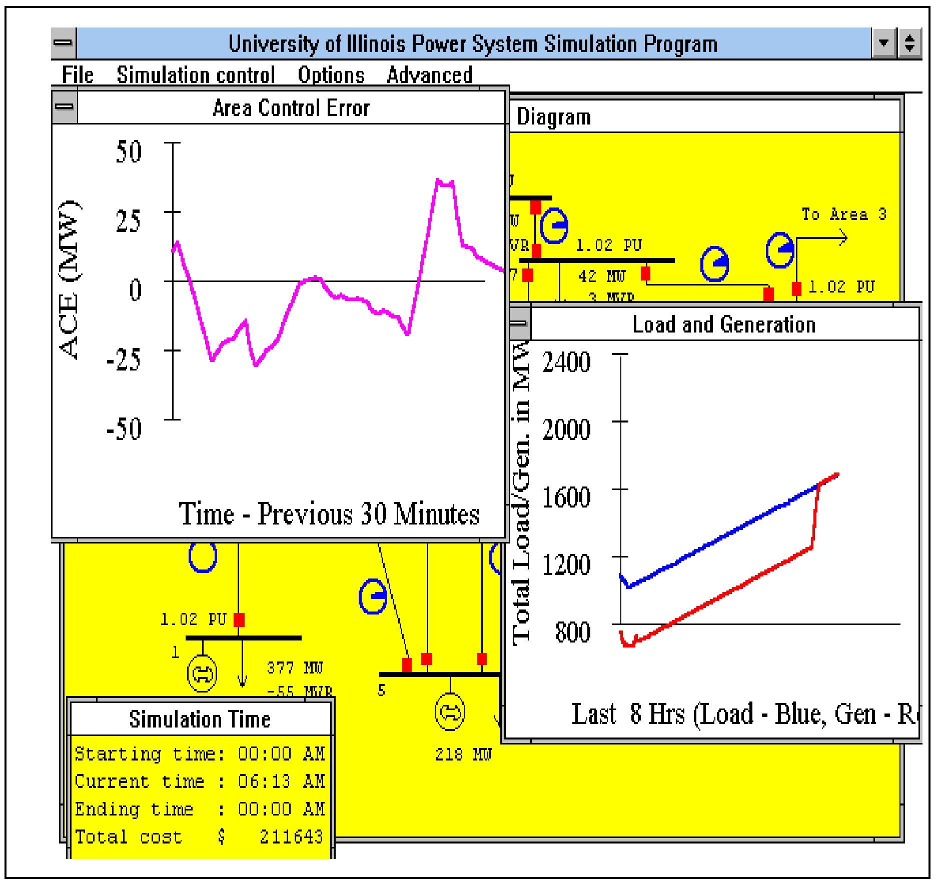
Figure 3: Program Object Hierarchy

be scheduled in this file to occur at specific times during the simulation.

Automatic System Variation

For each time step in the simulation a number of adjustments are automatically made to the system. First, the real and reactive loads at each bus are changed according to their piecewise linear model. Second, any scheduled events are enacted. Third, the system is checked for limit violations. All transmission lines are assumed to have thermal loading limits, with the line being capable of indefinite operation for any loading at or below this value. However, for loadings above this limit, the line temperature starts to increase. Line heating is approximated by integrating over time the square of the line current. Eventually, if the line flow does not decrease to a value below its limit, the line is automatically removed from service. When this occurs the user experiences one of a number of different cost penalties. Additionally, as an option, the program permits the simulation of random events, such as a transmission line outage due to lightning. This capability is included to convey to students the critical role such unexpected events can have on power system operations. In the simulation the user specifies the frequency of these events by choosing an option of either “never”, “sometimes”, or “often”. Whether a particular line is opened in a given time step depends on the selected option, the line outage probability, and the time step size.

Following these adjustments, topology processing may need to be performed to take into account any automatic line status changes, and any changes made by the user during the previous time step. Topology processing removes all buses that are not connected either directly or indirectly to the system slack. Load and generation at the removed buses are set to zero. Conversely, loads at buses that had been removed but are now reenergized are immediately reset to their correct (time-dependent) values; generation is optionally either immediately set to its pre-outage value, or ramped up to this value, or required to remain at zero (to simulate when a generator has tripped off-line and cannot be immediately put back on-line).



```
Starting time: 00:00 AM
Current time : 06:13 AM
Ending time  : 00:00 AM
Total cost   $   211643
```

Figure 4: ACE and Load/Generation Strip Charts

The simulation uses a full Newton-Raphson power flow solution. Sparsity techniques have been used to decrease both execution time and storage requirements. In order to maximize the region of convergence of the algorithm, the optimal multiplier technique [13] is used. However since the user is free to modify the system from the one-line diagram, it is certainly possible to stress the system beyond its point of maximum loadability. If this were to occur in actual operation, the system would probably experience a voltage collapse induced blackout. In the simulation anytime the power flow fails to converge, it is assumed that such a situation occurs. A message is displayed on the screen indicating that a blackout has occurred, the windows dim to a dark gray, and the simulation pauses.

Generator Models and Area Control

An important use of the program is to familiarize students with the requirements for area control, and the mechanisms of energy interchange between different operating regions. For an actual operating area, the area control error (ACE) is defined as a weighted sum of the difference between the actual and scheduled real power flow out of a control area, and the frequency error. The simulation assumes that the actual frequency is always equal to the scheduled frequency, so the ACE reduces to the difference between the actual interchange and the scheduled interchange. The user may select the Options menu to open “strip-chart recorder” windows that show either the ACE or the load/generation as a function of time. These windows are updated at each time step, with new data appearing on the left. These windows can also be resized and/or moved. The vertical (horizontal) scales can also be changed by double clicking on the vertical (horizontal) axis. As an example, Figure 4 shows the ACE over the most recent 30 minutes and the load/generation since the beginning of the simulation for the 13 bus system. In the load/generation window the upper curve is the load while the lower one is the generation; for this case the two curves differ primarily due to the presence of several scheduled transactions.

To keep the ACE close to zero, the user has three options for area control. First, the area can be controlled manually by clicking on the generator MW field to manually vary generation. By using the ACE strip-chart, the student gets instantaneous feedback of the effect of the generation change on the ACE. Second, the area can be placed on an automatic control where the ACE is allocated to the area generators according to specified participation factors. The third option is to use an economic dispatch (ED) routine to determine the generation dispatch. For both types of automatic control the change in generation can be (optionally) limited to take into account maximum up and down ramp rates (specified in MW/minute). Unit minimum/maximum MW limits are also enforced, and the generator options dialog (shown in Figure 2) can be used to take a particular unit off automatic generation control. For all cost calculations, the generator fuel-cost curves are represented by a quadratic cost function.

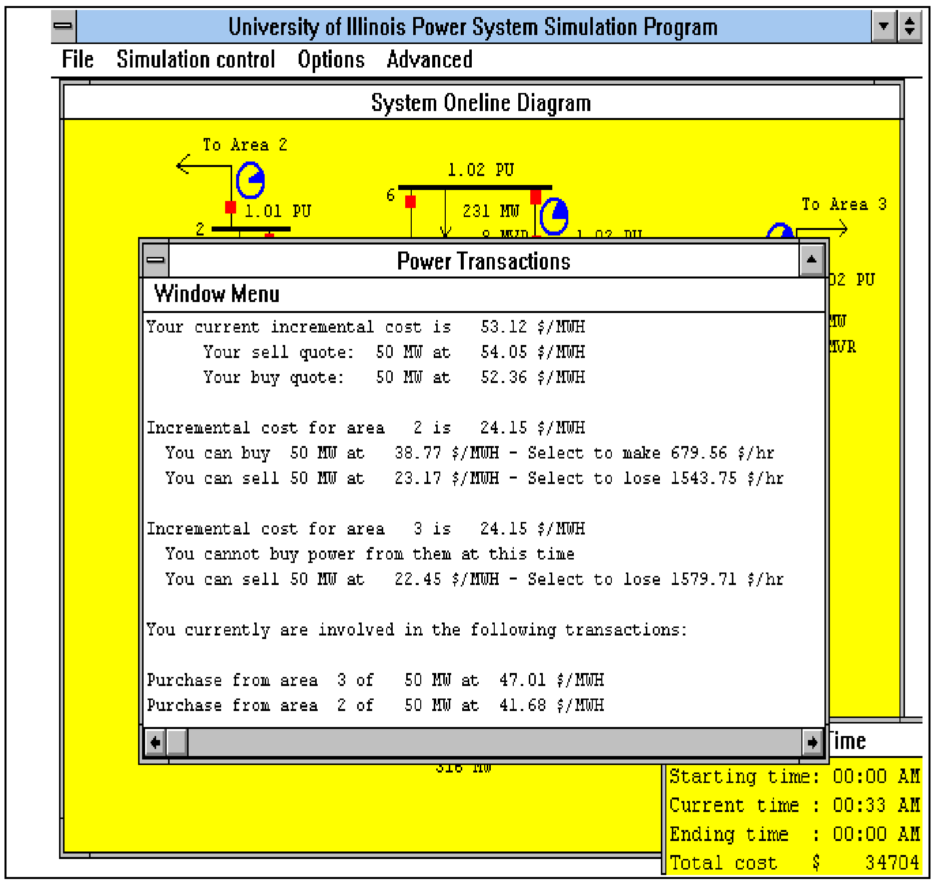
One of the most important benefits of interconnected operations is the ability for areas to undertake energy transactions. With increasingly more open transmission regimes, many more people involved in the electricity business will need at least a basic familiarity with how these transactions are implemented, and their potential impacts upon the transmission network. While in actual operations there can be an almost infinite variety of different types of power transactions, for simplicity the program currently only allows two generic types. One can either buy or sell a fixed block of power from or to another area at a fixed price. Such transactions are normally initiated as a result of economy, but can also be used to alleviate transmission system limit violations.

Power transactions are initiated using the Power Transactions window shown in Figure 5. The window first shows the incremental cost for the user's area, along with buy and sell quotes for a specified block of power (in $/MWh). The buy quote is determined by dividing the generation cost decrease due to proposed purchase by the MW purchase amount, while the sell quote is the generation cost increase divided by the sale amount.

Next the window shows the buy and sell prices for the other areas in the system calculated using a similar algorithm. Note that if an area does not have sufficient generation to do the anticipated transaction, then that transaction cannot occur and no quote is available. For example, this is the case for the Figure 5 system where buying power from area 3 would push that area's generators past their upper limits. The actual transaction price is then calculated to be the average between one area's purchase price and the other's sale price if the transaction is economical, or the actual purchase/sale price of the other area if it is not. For convenience, the total cost savings for each possible transaction are also shown in the window.

A transaction is initiated with an area by double clicking on its buy/sell price. All transactions begin to take effect immediately, ramping to their full value over a ramp time (set by default to 10 minutes). The bottom of the window shows all of the currently active transactions.

The total simulation cost is modified to take into account the cost and/or income from each transaction. Additionally, the program keeps track of the inadvertent interchange error. This value is calculated by integrating the ACE over time and has units of MWh. At the end of the simulation a negative inadvertent error indicates that the user has (overall) undergenerated; the total simulation cost is therefore increased by the inadvertent multiplied by a high cost per MWh. At anytime during the simulation the inadvertent error, along with a number of other area values, can be viewed by displaying the area information window.



```
Your current incremental cost is   53.12 $/MWH
      Your sell quote:  50 MW at   54.05 $/MWH
      Your buy quote:   50 MW at   52.36 $/MWH

Incremental cost for area   2 is   24.15 $/MWH
  You can buy   50 MW at   38.77 $/MWH - Select to make 679.56 $/hr
  You can sell 50 MW at   23.17 $/MWH - Select to lose 1543.75 $/hr

Incremental cost for area   3 is   24.15 $/MWH
  You cannot buy power from them at this time
  You can sell 50 MW at   22.45 $/MWH - Select to lose 1579.71 $/hr

You currently are involved in the following transactions:

Purchase from area  3 of   50 MW at  47.01 $/MWH
Purchase from area  2 of   50 MW at  41.68 $/MWH
```

```
Starting time: 00:00 AM
Current time : 00:33 AM
Ending time  : 00:00 AM
Total cost   $    34704
```

Figure 5: Power Transactions Window

Limitations of the Simulation

The basic idea of the program is to simulate the evolution of load on a power system over a relatively long period of time. Therefore the program only models constant frequency operation, using the power flow "infinite bus" approach. As the load and other parameters change throughout the scheduled period, the program solves the power flow equations and displays the results. As such the program is a sequence of static simulations. The only dynamic aspects of the program are the enforcement of ramp rates on generation and real power transactions, the pre-programmed load variations, and the thermal overloading limits on transmission lines. Additionally, only two general types of MW transactions are allowed.

Using the Program as a Teaching Tool

The program may be used in many ways to teach the fundamentals of power system operations and control. Included with the program are a number of test systems ranging in size from two buses to thirteen buses. The program can be used by the instructor as a lecture tool to demonstrate phenomena such as load cycling during a time period, power flows on parallel lines, correlation between reactive power and voltage, economic dispatch, area control error, automatic generation control, wheeling, and the impact of contingencies on system operations.

As a lecture tool, the instructor may choose a simple system and begin the simulation. As the program executes, the instructor may display various quantities such as ACE, incremental costs, power transaction options, and other features. Generation units may be removed from participation in AGC and adjusted manually by the instructor in an attempt to maintain zero ACE as the load changes with time. The effects generators have on transmission line flows can also be demonstrated. The instructor could simulate line outages during the simulation to show the impact of lines out for maintenance or lost due to contingencies. If the random contingency events option is activated, the program would randomly remove lines, showing the effects stochastic events have on system operation. Intervention by the lecturer is sometimes required to provide manual generation rescheduling to remove any resultant overloads.

Outside of the classroom (or for assignments), the students can use the program to gain a more intuitive feel for power system operations and control. If the students run the test systems without taking any action, there will be a total cost associated with the load scheduled cycle. They could first simply observe the automatic operation of the system to get a physical feeling for the interaction between loads, generation, line flows, bus voltages, and the power system economics. Then, rather than being passive observers, the students may be challenged to improve the economic outcome of the simulation.

Using one of the test systems with a given load cycle, the students could be tasked to operate the system to minimize cost. By undertaking power transactions, the students can take advantage of cheaper energy from neighbors. The student who goes through one load cycle with minimum cost would be the "winner". However, someone who tries to make too much money on transactions may find that overloads are encountered, and if the overloads are severe enough, they may result in line outages with potentially severe financial consequences or even total system collapse. Students quickly learn the need to minimize cost while maintaining system security and hence the need for applications such as optimal power flow. Also, the presence of stochastic events reinforces the need for the consideration of contingencies when operating a power system. An instructor could stimulate a classroom discussion on the importance of considering such events in system planning and operations, and thus the need for applications such as contingency planning.

Program Distribution

The program and user's guide are available for a nominal fee from the authors. The nominal fee is charged to cover the cost of duplicating and mailing.

Conclusion

This paper has described a user-friendly power system simulation program. The program is particularly effective in providing users with an intuitive feel for power system operations. The object-oriented design approach allows users great flexibility to dynamically interact with the simulation, while retaining a small size for the source and executable files. The use of data files and option windows allow users to simulate a wide variety of different operating conditions.

Acknowledgments

The authors would like to acknowledge the support of the Edison Electric Institute (EEI), the Power Affiliates program of the University of Illinois at Urbana-Champaign and the Grainger Foundation in the development, implementation and documentation of this program. We are indebted to the encouragement and advice of Bruce Edelston, Paul McCurley and Ellen Vancko of EEI.

References

[1] R. Podmore, et. al., “An advanced dispatcher training simulator,” *IEEE Trans. on Power App. and Sys*., Jan 1982, pp. 17-25.

[2] M. Prais, et. al., “Operator training simulator: algorithms and test results,” *IEEE. Trans. on Power App. and Sys*., Aug. 1989, pp. 1154-1159.

[3] Operator Training Simulator, EPRI Report, EL-7244, 1991.

[4] IEEE PES Current Operational Problems Working Group, “Dispatcher training simulators lessons learned,” *IEEE Trans. on Power Sys*., vol. 6, May 1991, pp. 594-604.

[5] D.C. Yu, S.-T. Chen, R.F. Bischke, “A PC oriented interactive and graphical simulation package for power system study, *IEEE Trans. on Power Sys*., Feb. 1989, pp. 353-360.

[6] N. Pahalawaththa, C.P. Arnold, and M. Shurety, “A power system CAD package for the workstation and personal computer environment,” *IEEE Trans. on Power Sys*., vol. 6, Feb. 1991, pp. 400-406.

[7] J.A. Huang and F.D. Galiana, “An integrated personal computer graphics environment for power system education, analysis and design,” *IEEE Trans. on Power Sys*., vol. 6, Aug. 1991, pp. 1279-1285.

[8] B.H. Chowdury and D.E. Clark, “COPERITE - computer-aided tool for power engineering research, instruction, training and education,” *IEEE Trans. on Power Sys*., vol. 7, Nov. 1992, pp. 1565-1570.

[9] M. Foley, A. Bose, W. Mitchell and A. Faustini, “An object based graphical user interface for power systems,” *IEEE Trans. on Power Sys*., vol. 8, Feb. 1993, pp. 97-104.

[10] S. Li and S.M. Shahidehpour, “An object oriented power system graphics package for personal computer environment, *IEEE Trans. on Power Sys*., vol. 8, Aug. 1993, pp. 1054-1060.

[11] *Borland Pascal with Objects User’s Guide*, Borland International, Scotts Valley, CA, 1992.

[12] H.E. Pierce Jr. et. al., “Common format for exchange of solved load flow data,” *IEEE Trans. Power App. & Sys*., vol. PAS-92, pp. 1916-1925, Nov/Dec. 1973.

[13] S. Iwamoto and Y. Tamura, "A load flow calculation method for ill-conditioned power systems," *IEEE Trans. Power App. & Sys*., vol. PAS-100, pp. 1736-1743, Apr.1981.

Thomas J. Overbye (S'87, M'92) received his BS, MS, and Ph.D. degrees in electrical engineering from the University of Wisconsin-Madison in 1983, 1988 and 1991 respectively. He was employed with Madison Gas and Electric Company from 1983 to 1991. Currently he is an Assistant Professor of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. In 1993 he was the recipient of the IEEE PES Walter Fee Outstanding Young Engineer Award.

Peter W. Sauer (S’73, M’77, SM’82, F’93) received the BS degree in electrical engineering from the University of Missouri at Rolla, and the MS and Ph.D. degrees in electrical engineering from Purdue University in 1969, 1974 and 1977, respectively. He is currently a Grainger Associate and a Professor of Electrical Engineering at the University of Illinois at Urbana-Champaign. He is a registered Professional Engineer in the Commonwealth of Virginia and the State of Illinois.

Caroline M. Marzinzik (S’94) received her BS degree in electrical engineering from the University of Illinois at Champaign-Urbana in 1994. Presently she is working on her Master’s degree.

George Gross (F’ 88) is the Grainger Professor of Electrical and Computer Engineering and Professor, Institute of Government and Public Affairs, at the University of Illinois at Urbana-Champaign. His current research and teaching activities are in the areas of power system analysis, planning, economics and operations, utility regulatory policy and industry restructuring. His undergraduate work was completed at McGill University, and he earned his graduate degrees from the University of California, Berkeley. He was previously employed with Pacific Gas and Electric Company.