

Building-Level Power Network Analysis

Steven B. Leeb¹, James L. Kirtley, Jr.¹,
David S. Woodruff¹, and Michael S. LeVan²

Industrial and commercial facilities managers are turning to software tools to assist with the design and maintenance of building power distribution networks. In principle, these tools are ideal for tracking variable load configurations and analyzing their operation. However, few inexpensive tools combine the following capabilities with a simple user interface that enables fast, flexible updates to the database describing a building's power network:

- ✧ Understanding of national and proprietary electrical codes
- ✧ Electric load simulation
- ✧ Basic checks for criteria such as steady state power balance, overloading, and short circuit analysis.

This article features the Radial Panel Installation Designer (RAPID), a PC software program developed at MIT's Laboratory for Electromagnetic and Electronic Systems. The minimum hardware platform for running the program is a PC-AT class computer with an EGA monitor. A mouse is helpful, but not required.

The program enables a user to quickly and easily create, move, or alter the data that constitutes a building description stored in the program's database. Like a spreadsheet, the program can be used to conduct what-if studies on a network. Automatic recalculation permits the creation and modification of power distribution networks far more easily than pencil and paper allow. The program remembers the relationships between the various network elements, e.g., the location and type of transformers, upstream breakers, service entry capacity, etc., and will compute relevant network parameters.

At each stage of the specification process, while loads are being added or altered in the network, the user may enter a specific choice of components such as wire, conduit, and breaker size, or select choices determined by

***Facilities managers
can quickly and easily
create, move, or alter
the data that constitutes
a building description
stored in the program's
database***

the program on the basis of information the user has entered about the load. If a user's choice is in conflict with the program's, the user is warned. The program determines its choices by referencing tables derived from the National Electrical Code. These tables are stored in American Standard Code for Information Interchange (ASCII) files and may be modified by a knowledgeable user to reflect relevant local standards.

The program is capable of answering questions about the dynamic operation of parts of the network. Two different types of simulators are available in separate program versions.

- ✧ Version 1 uses piece-wise linear waveform representations to model the *power profile* of a load.
- ✧ Version 2 uses state space or algebraic models to represent load dynamics.

The first type of simulation is particularly valuable for approximate studies of cold load pick-up transients. In this simulation, a piece-wise linear waveform specified by the user represents the level of apparent power drawn by each load from turn-on to steady-state operation. A simulation is conducted by aggregating these waveforms for all of the loads in a section of interest in the building. Parameters such as peak power drawn on start-up and peak power with respect to average load in steady-state may be determined. This type of simulation is particularly useful for examining the impact of different operation schedules on the power demand for the building.

The second type of simulation is more sophisticated. A detailed software routine describes the electrical behavior of each load. This software description may be a set of algebraic equations, a state space description, or a prepackaged model built into the program for certain loads, such as induction motors. The program incorporates a recursive descent parser for interpreting the code and uses advanced numerical integration schemes to solve the differential equation systems. Phase currents for each load are computed for every electric line cycle. Current waveforms for each load are aggregated to find

¹ Laboratory for Electromagnetic and Electronic Systems, Massachusetts Institute of Technology

² Amp Incorporated

the composite waveforms for any section of the network. This simulation is considerably more numerically complex than the piece-wise linear simulation. However, this type of simulation is capable of yielding extremely sophisticated and accurate results given accurate load models. The user may save data from a simulation for later examination in other analysis programs. These results may be especially useful, for example, for analyzing the harmonic content of current waveforms in the network. To maximize simulation efficiency, common loads are represented by prepackaged, optimized models built into the program.

Raising a Building

The program models the electric distribution system in a building as a radially-distributed network of panels. A panel may represent an actual circuit breaker panel in a building, or a busbar, or a transformer. The root-panel in a building model typically represents an actual circuit breaker panel fed by the main three-phase utility service entry into the building. Each switch on a panel represents a circuit breaker or bus connector that initiates a connection to a load or a subpanel, busbar, or transformer.

To create a building model, the user first answers questions that define the utility service entry (details like the line-to-line voltage, entry fusing, entry transformer type, and so on). Once the program has the basic details of the service entry, a main panel file is created in a building database. This main panel is the root-panel for what will eventually be a network of panels, distributed radially from the main panel, which describe all or part of the building's electric wiring.

Describing the Wiring

Initially, the main panel contains no switches. The user specifies the details of each switch or breaker on the panel, one switch at a time. Figure 1 shows the main panel screen for a building named TEST. Several of the switches on this panel have already been specified.

To add a new switch to the panel, the computer's mouse is clicked in the window labeled *Switch* in Figure 1. A blank switch is created in the window. The program

queries the user for some basic boiler-plate information about the device to be added. The program recognizes eight different device classes:

Load represents most general electrical loads.

Motor includes induction, dc, and synchronous electric machines, or loads whose primary electric component is a rotating electric machine.

Receptacle represents an electric outlet, recognized only on 120/208 volt panels in the current version.

Spare and space represent unused portions of the panel.

Panel, Transformer, and Bus represent the classes that can be panels. The program understands the differences between the various transformer winding types, such as wye-delta, delta, wye-wye, and delta-delta transformers. Phase currents are transformed correctly across the transformer types selected by the user.

Based on the device name input by the user, the program attempts to interpret the device class. The user is offered a default class for the device or may select any other choice from the remaining seven classes. The program also asks for the nominal load power, power factor, and number of phases used by the load.

The remaining questions about the load are cross-referenced against the program's knowledge of the electrical code and the user-specified load characteristics. The user is asked, for example, to specify the breaker or fuse type, wire size for the hook-up, associated conduit, turn-on and turn-off times, etc. Some questions may be device class specific. For each question, the user is allowed to enter a specific choice or select a programmed default. If the user's choice conflicts with the program's in an unsafe manner, e.g., the wire size is too small, the user is warned.

During this device creation process, the user is also asked to specify the transient electrical characteristics of the load. Depending on the version of the program, this specification may involve the creation or selection

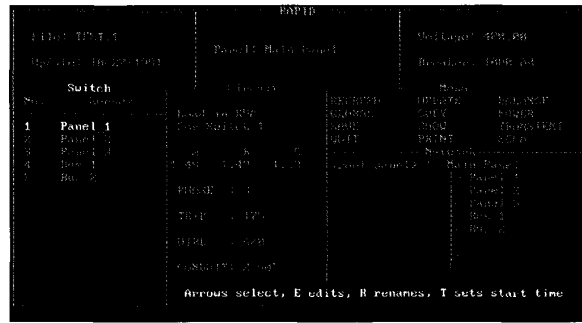


Figure 1. Main interface window

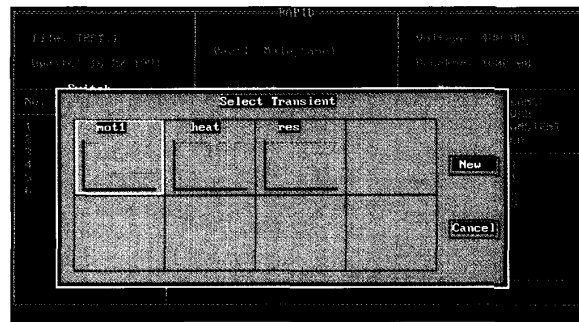


Figure 2. Transient profiles

of a previously saved piece-wise linear waveform, or it may involve the creation of a code-fragment that specifies the device's electrical characteristics. As an example of the first type of transient specification, Figure 2 shows the selection window for some typical transient waveform profiles as they are presented to the user. Figure 3 shows the edit window for the second version, with a code fragment describing an induction motor using one of the built-in rotating machine models.

The *Circuit* window (shown in Figure 1) displays pertinent information about a specific device under consideration. For example, when a new device is fully specified, the circuit window will be updated to display the steady-state power and hook-up specification for the new load. The information for any load on the panel may be viewed in this window by first selecting the load in the switch window.

If the new load is a panel, busbar, or transformer, a new database file will be created for the load, and the *Network* window will be updated. This window shows the local structure of the building power network. The top of the *tree* in this window (shown as *Main_Panel* in Figure 1) is the panel currently under examination. To the left of the current panel is the *parent* panel for the current panel. (In Figure 1, the root panel in the network is under examination, so the parent panel is listed as *<root-panel>*). The new *child* panel, busbar, or transformer which has just been added will appear as a *leaf* below the current panel. The user may move up and down the network by selecting a parent or child panel with the mouse. When a new panel is selected, the entire screen shown in Figure 1 will be updated. The program updates the top of the screen in Figure 1 to display the status information for the new panel, e.g., line-to-line voltage, feeder breaker size, etc. The switch and circuit information for the new panel will be displayed, and the network window will be updated to show the current location in the network.

Any previously speci-

***The program's
analysis tools provide
speedy safety checks and
answers to hypothetical
load scheduling questions
with a high degree
of certainty
and flexibility***

fied switch may be deleted. The specifications for any load that is not a panel, busbar, or transformer may be altered or copied. The ease with which data may be entered and altered makes the program ideal for tracking and analyzing changing load configurations.

Tailoring the Program

The user may adjust some of the parameters governing the program's operation by selecting the GLOBAL command in the *Menu*

window of the main interface screen (Figure 1). This command activates the tailoring window, shown in Figure 4. By selecting one of the options, the user may set the selected parameter. Examples include the smallest wire size allowed in the network, the number of switches permitted on a single panel, and the maximum over-current to be used in computing fuse or breaker sizes.

The user may engage in more sophisticated tailoring of the program by altering the ASCII files in which the program stores its knowledge of basic wiring requirements. These tables cross-reference wire ampacity and gage with breaker and conduit size and type. Default power factor information for a variety of common loads is also saved in these tables. These tables are initially set to reflect requirements specified in the National Electrical Code. Altering the entries in these tables will tailor the program for a wide variety of local or proprietary standards.

Analyzing a Network

Many of the commands in the *Menu* window provide information about the operating characteristics of the network. The BALANCE and POWER commands may be used to find the relative steady-state phase loading and total steady-state power per phase, respectively, for the current panel under examination. The program automatically aggregates the total power used on all subpanels. Any changes to the loading of subpanels are reflected in the total steady-state power and balance calculations for the current panel.

The PRINT command produces a summary sheet of the loads on the current panel by switch number; this summary is suitable for updating the legends frequently incorporated in circuit breaker panels.

The program is also capable of providing a rudimentary short circuit current analysis (SCCA) through the SCCA command. The user may place a hypothetical fault on a one-line diagram of the network, as shown in Figure 5. The program will compute the available short circuit current at the fault and also at the service entry based on the calculated impedances of the connecting wires

```

mot
outfile ind.dat 10
algor rk
step .00104166666
rs = 0.0453
Xm = 2.0420
rr = 0.0222
X1s = 0.0775
X1r = 0.0322
we = 377
w = 377
H = 1.0
PUV = 1
PUl = 1
PUT = 1
Ti = 9
induc
end

```

Figure 3. Induction motor electrical specification

and the known impedances of transformers in the network.

The TRANSIENT command engages the load simulator, described as follows.

Studying Electric Transients

The piece-wise linear simulator is particularly useful for studying the impact of different operating schedules on the peak power demand over certain intervals. Part of the specification information for entering a new load into the program is an absolute start and stop time for the load, assuming that no load starts before time zero. The user may easily change the start and stop times for any load, permitting the study of a limitless range of operating schedules. For example, the program can give rough estimates of the cold load pick-up transient, i.e., the electric demand when many loads are suddenly turned on. The program can be used to assist in the timing of load operation to help avoid excessive electric demand during any single short time billing period. The simulation may be conducted for the whole or any part of the network.

For an example of typical results from the piece-wise linear simulator, consider the plot of power versus time shown in Figure 6. This plot shows the actual aggregated apparent power drawn by a bank of three thermostatically-controlled resistive heaters. Figure 7 shows a simulated plot of this bank of heaters, in which the cycle times for the various heaters have been approximated and

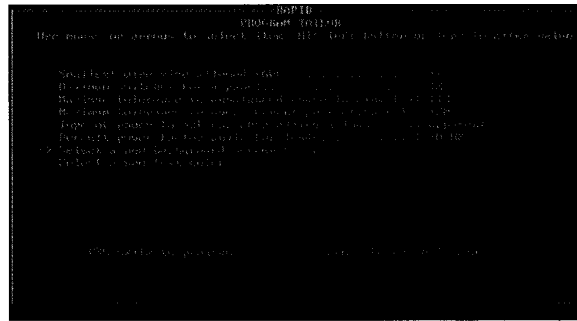


Figure 4. Program tailoring

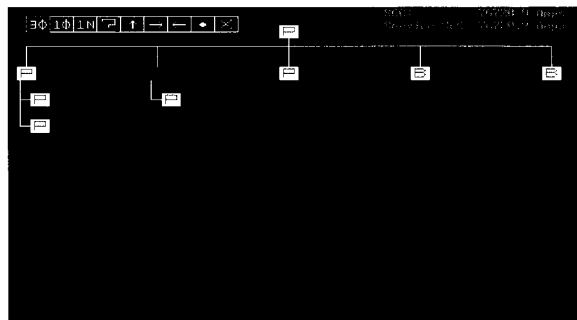


Figure 5. One-line-network diagram

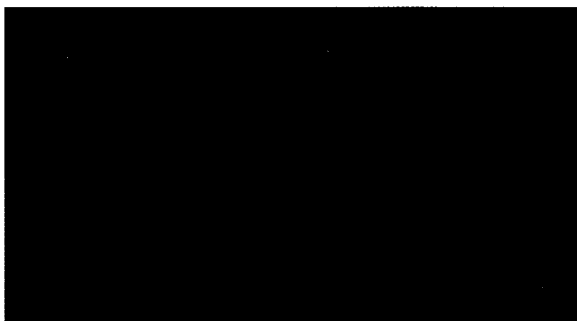


Figure 6. Observed power profile

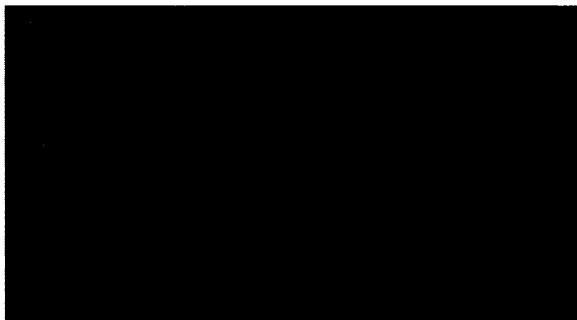


Figure 7. Simulated power profile

entered as part of the load model. Given a reliable load model, the program is capable of constructing accurate apparent power usage profiles. A one-time investment in collecting basic load information such as the cycle time for the heaters, either through online data collection or offline calculation, can yield generally applicable models for a variety of load simulations. Because the program stores the entire power waveform profile for each load, computation during the simulation is minimized and many simulations may be conducted in a relatively short time.

Figure 8 shows the three-phase currents computed by a state space simulation of a panel with two loads, a three-phase induction motor, and a single-phase resistive heater. This type of simulation is more sophisticated than the piece-wise linear simulation. The induction motor starts first at time 0. The resistive heater starts at time 1, after the induction motor has entered steady-state. The plot in Figure 8 shows the three-phase currents for a simulation interval of 1.5 seconds. The induction motor model is a standard induction motor state space representation that has been built into the program. (A typical code fragment used with this type of model is shown in Figure 3.) The resistive heater is modeled with a simple set of algebraic equations. While more time-consuming, this analysis is capable of giving detailed, cycle-by-cycle representations of the load dynamics. The level of

The level of detail presented by the state space simulation is particularly suited for harmonic content analysis and high resolution transient studies. Point-by-point data from the simulations may be downloaded to an ASCII file for analysis in other programs.

Benefits

This program provides the industrial or commercial facilities manager many important design and maintenance tools that are not available in many standard CAD and database packages. The ability to prototype and alter the database that describes a building's electrical network facilitates good documentation habits. More significant, the program's analysis tools can provide speedy safety checks and answers to hypothetical load scheduling questions with a greater degree of certainty and flexibility than most other programs. The custom interface and analysis routines make the program a more natural tool for tracking building operation than a generic spreadsheet or database program.

The program is an invaluable tool for generating raw data for initial checks of many electronic circuits and software algorithms. It has been used in the Laboratory for Electromagnetic and Electronic Systems for providing test data for harmonic analysis hardware and electrical transient pattern recognition algorithms, and may eventually be used in forecasting studies.

Acknowledgements

Primary funding for this work was provided by AMP Incorporated. The authors are especially grateful for the visionary guidance, enthusiasm, and unflagging support of Joseph Sweeny. Additional funding for this work was provided by the Electric Power Research Institute. The authors thank David Hitz and Jerry Weintraub for their assistance in collecting transient data, and Professors George Verghese and Jacob White for their assistance in understanding relevant issues associated with state space simulation.

For Further Reading

National Fire Protection Association, *National Electrical Code*, 1990. Bussman Division, *Electrical Protection Handbook*, McGraw-Edison Company, bulletin SPD81.

The program allows the user to prototype and alter the database that describes a building's electrical network

W. Press, B. Flannery, S. Teukolsky, W. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1988.

S. Fenton, *An Expert-System Approach for Evaluating Start-Up Transient Behavior via a Microcomputer-Based Software Package*, S.M. Thesis, MIT Department of Mechanical Engineering, June 1985.

Biographies

Steven B. Leeb is a graduate student in the Laboratory for Electromagnetic and Electronic Systems at the Massachusetts Institute of Technology (MIT), Cambridge. He received the S.B., S.M., and E.E. degrees from MIT in 1987, 1989, and 1990, respectively. He is currently pursuing a doctoral degree at MIT in the area of power systems and system identification. He is a member of the IEEE Power Engineering Society.

James L. Kirtley, Jr., received the S.B. and PhD degrees from the MIT in 1968 and 1971, respectively. He joined the MIT faculty in 1971 and is now associate professor of electrical engineering. During 1974 and 1975, he worked as a generator engineer in the Large Steam Turbine Generator department of General Electric, Schenectady, New York. He is secretary of the Electric Machinery Committee, a member of the Synchronous Machinery

Subcommittee, and is chairman of the Machinery Theory Subcommittee. He was chairman of the 1990 International Conference on Electric Machines. He is a member of CIGRE, a member of the editorial board of *Electric Machines and Power Systems*, and a registered professional engineer in Massachusetts.

David S. Woodruff received the BSEE from Manhattan College in 1963. He received the MS degree in physics in 1965 and the PhD in math in 1974 from Polytechnic University. He is a member of the Research Staff of the MIT Laboratory for Nuclear Science, where he is manager of Computer Applications and Systems Programmer. He has worked with the MIT Laboratory for Electromagnetic and Electronic Systems in several computer-related projects. He has taught math and physics at a number of institutes. Currently he is teaching Discrete Math for Computer Science at the Metropolitan College of Boston University.

Michael S. LeVan is the manager of Design Engineering Services of the Facilities Services Division of Amp Incorporated in Harrisburg, Pennsylvania.

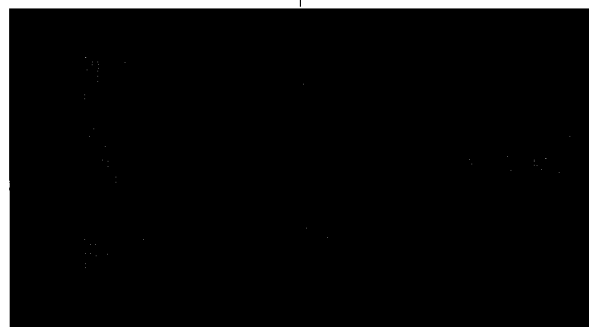


Figure 8. State space simulation