SIMULATION TOOL FOR DISTRIBUTION-SYSTEM MODELING, ANALYSIS AND ALGORITHM TESTING

Khoi Vu Kevin Timko Damir Novosel
ABB Transmission Technology Institute
Raleigh, NC, USA

Tapio Hakola
ABB Transmit Oy
Vaasa, Finland

ABSTRACT

PQ-SMART—Power Quality Simulation, Monitoring And Relay Testing—was designed by a group of power engineers at ABB to support the R&D work of the company. This tool integrates MATLAB and EMTP, two commercial softwares widely available at the company's research labs. Although the tool is, and will always be, restricted to internal use, the experience can serve as an example for future EMTP development.

Keywords: Power Quality, Time-domain Simulation, EMTP, Algorithm Testing, MATLAB, PQ-SMART.

INTRODUCTION

At present, user-friendly software tools are available to study steady-state behavior of distribution circuits in connection with power quality. For example, the magnitude of voltage sags can be studied with such tools. However, if one is interested in the duration of the sags, the steady-state model is not adequate. This is also the case if the behavior of complex devices such as arc furnaces, electric drives, surge arrestors, etc. is to be included in the system model. Another deficiency of steady-state tools is that one cannot verify or test the operation of protective and monitoring algorithms.

The EMTP (ElectroMagnetic Transients Program [1]) has been used extensively by power engineers to study transient behavior of electrical systems. A major problem with EMTP is that it is difficult and time consuming to set up a case. Even the task of modifying an existing file to generate new simulation cases requires some level of expertise. At present, the EMTP's development group, a consortium of utility and industrial organizations, is planning to make the EMTP more user-friendly by adding a graphical interface; however, it may take several years before this new feature is fully developed.

Recognizing that the EMTP has been the industry standard for over 20 years and that a significant and valuable work has been invested in modeling power-system components, a team at ABB decided to build a new simulation tool on top of EMTP. This tool is designed to be run on personal computers. Basically, it relies on an extensive library of EMTP modules, each module being a parameter-based description of a particular power-system component (transformer, load, motor, etc). This library is at the heart of the simulation tool because it directly influences how realistic the simulation results will be.

The software tool is designed with a number of objectives. It is to provide an easy assessment of protection, fault location, and especially power-quality monitoring functions. It is to support the sale of equipment to utility customers. Also, it must be user-friendly (i.e., the user does not have to learn EMTP or any language) and easy to expand.

The graphical user interface is built entirely with MATLAB [2]. (The design philosophy here is totally different from MatEMTP [3] which aims at re-developing the transients program completely.) The choice of MATLAB is driven by two forces. Firstly, MATLAB is already in use at all ABB research labs. Secondly, the numerical capability of MATLAB is very useful for algorithm development and evaluation.

DESCRIPTION AND APPLICATIONS OF SOFTWARE TOOL

The software tool was originally intended for the simulation and analysis of power quality in distribution systems. It has the code name PQ-SMART, for Power Quality Simulation, Monitoring And Relay Testing. Unlike competing tools for Power Quality study that rely on steady-state methods, PQ-SMART involves simulation in the time domain.

The tool combines the simulation and modeling capabilities of the EMTP with the data processing and analytical tools in MATLAB. Communication between the EMTP and MATLAB is accomplished via the data input and output files (Fig. 1).

Three main elements of the tool are (1) EMTP model library, (2) MATLAB-based graphical user interface (GUI), and (3) output post-processing/analysis facilities.

Each of the library components is represented by a single icon in the MATLAB-based one-line diagram (see Fig. 7). By using the computer's mouse, the user can assemble icons to build and modify a distribution network. The program compiles the user's input and starts the simulation automatically. Retrieval of simulated waveforms (voltages and currents) is done by clicking on icons designated as "sensors".

A set of drop-down menus allow the user to apply a number of power-quality analysis functions to the observed waveforms. The list includes not only functions being used in the power-quality field, but also new functions that have potential applications. Most importantly, the analysis package allows the user to "plug and play" his/her own algorithms. This flexibility is very valuable to the product-

development process because the engineer can quickly evaluate and fine-tune an algorithm in question.

An important application of PQ-SMART is in equipment testing. The simulated voltages and currents are sent to a test apparatus via power amplifiers. The user can run a large number of cases and evaluate the performance of the apparatus under simulated conditions. For example, a harmonic analyzer can be tested and its results are compared against the output of the MATLAB-based routines.

Another application of PQ-SMART is the use of its DSP package for post-processing of recorded field data. In this case, the user bypasses the EMTP simulation and imports the data directly into MATLAB.

EMTP Model Library

The EMTP Data Module (EDM) feature has been used extensively in the simulation tool to simplify the use of the basic EMTP equipment models and to extend the modeling capabilities of the EMTP to more complex distribution system equipment models. The model library developed for this purpose relies heavily on the enhanced EDM features of Version 3.0 of the EMTP (EMTP96). In particular, the new EDM calculation section allows simple calculations to be performed with module arguments and internal module variables using a pseudo-Fortran language. The results of the calculations are then substituted in the EMTP data case at run time when the module `include` file is loaded. This feature permits the module input data, for example, to be entered in the form most convenient to the user while the EMTP handles the data conversion to the required units at runtime. In addition, the free format structure of the EDM module calls, together with the sorting capabilities of the revised EMTP, allow composite models of complex systems to be constructed from elementary EMTP components and encapsulated in modular form. Modules can be reused within the same data case without fear of name and variable conflicts, provided the modules are properly designed and implemented.

The modules that have been developed to date include:

1. Line Models
2. Transformer Models
3. Breaker/Switch Models
4. Source/Equivalent Models
5. Load Models
6. Protection Models
7. Power Electronic Models
8. Miscellaneous Models, including a sensor model, shunt and series capacitor bank models, and a generic evolving fault module.

At the time of this writing, models are being developed for arcing faults, various protection/relaying and power electronic equipment, such as ac drives, converters, etc.

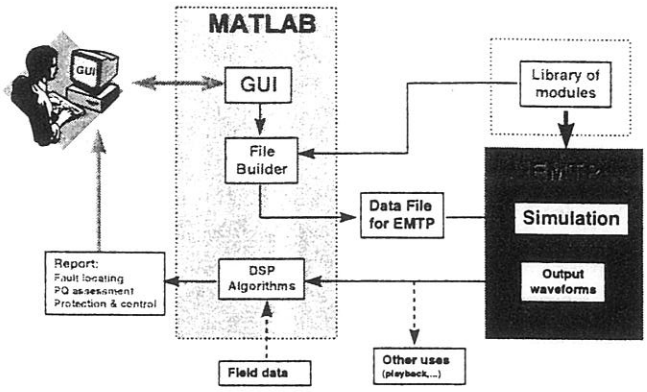

Figure 1: Architecture of PQ-SMART.

To illustrate the EDM module concepts used in PQ-SMART, several basic and composite EMTP distribution equipment models developed for the tool are described below.

Example 1: Feeder Model

The FEEDER model, Fig. 2, is a constant impedance, lumped-element representation of an unbalanced multi-phase distribution circuit based on the EMTP branch model. Up to three coupled phases can be represented, depending on the settings of the module phase selectors, with the phase or sequence impedance data for the multi-phase circuit supplied in ohms/mhos per unit length. Line charging is represented provided the susceptance data for the circuit is supplied. Input data for the model consists of the phase or sequence conductor resistances and reactances in Ω/mi., the phase conductor susceptances in mhos/mi., the feeder length in mi., and the plotting selector. Three phase selectors are included in the module argument list to determine the number of phases to be modeled and to select the appropriate feeder model.

Example 2: General Harmonic Source Model

The HARMSR1 general harmonic source model in PQ-SMART consists of a three-phase TACS-controlled Type 60 slave current source. Input data for the model consists of the source kVA and kV ratings for the source, the amplitude of the dc offset in pu, and the magnitudes and phases of the fundamental frequency and harmonic currents up through 15 in pu and degrees, respectively. This magnitude and phase

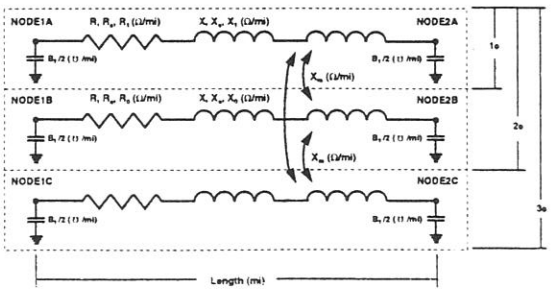

Figure 2: Feeder Model

information is then used to construct the TACS Fourier series which is used to control the three phases of the Type 60 current sources. The phase relationship between individual harmonic currents in each phase of the source at each harmonic is assumed to be positive sequence, i.e., the injected currents are shifted individually by 120°. This is the general case for ungrounded systems.

Example 3: Power Electronic Converter Model

The RECT01 model shown in Fig. 3 represents a three-phase industrial rectification system, and illustrates the type of composite models using the enhanced EDM of Version 3.0 of the EMTP. The basic rectification system consists of a 6-pulse diode midpoint converter with a resistive dc load which as shown, can be connected in parallel with appropriate phase shifts between rectifier units to simulate 6-pulse, 12-pulse, and 24-pulse modes of operation. The major components of the rectification system include the LTC regulating transformer, the unit transformer, the diode bridge, and the control and protective equipment for the system.

Each unit is connected to the ac system via the LTC regulating transformer, the primary means of voltage control for the rectifier. (Due to the time delay inherent in automatic LTC operation, unit control was not included in the converter model.) The dc system is supplied through a three-phase, double-wye-connected unit transformer. The transformer is of the variable phase shift extended-delta type and supplies the 100 V diode rectifier bridge through phase shifts of ±7.5° or ±22°, as specified by the user. A 30° phase shift between a pair of rectifier units provides for 12-pulse operation with cancellation of the characteristic 6-pulse harmonics, assuming both rectifiers are loaded equally. A 15° phase shift between two pairs of 6-pulse units, with each pair operating in 12 pulse mode, provides for a 24-pulse system and cancellation of the 12-pulse harmonics.

The diode bridge consists of two three-phase diode groups connected in parallel at the midpoint through an interphase transformer. The interphase transformer forces a balanced current to flow in both groups, thereby forcing 6-pulse operation of the bridge. A combination resistive/ inductive load based on the user supplied kW rectifier load is included as the default load in the system model.

Example 4: Fault Model

The FAULT module is a generic, evolving fault model simulating up to two independent impedance-grounded fault scenarios, each involving one or more phases at the specified node. The duration of each scenario, as well as the point-on-wave fault inception angle of the first scenario, is specified by the user. The fault is initiated at the desired point-on-wave after the pre-fault simulation specified by the user.

The model is currently implemented with back-to-back, opposing EMTP Type 11 switches, i.e., thyristor switches, and a TACS monitoring/control system to detect the specified

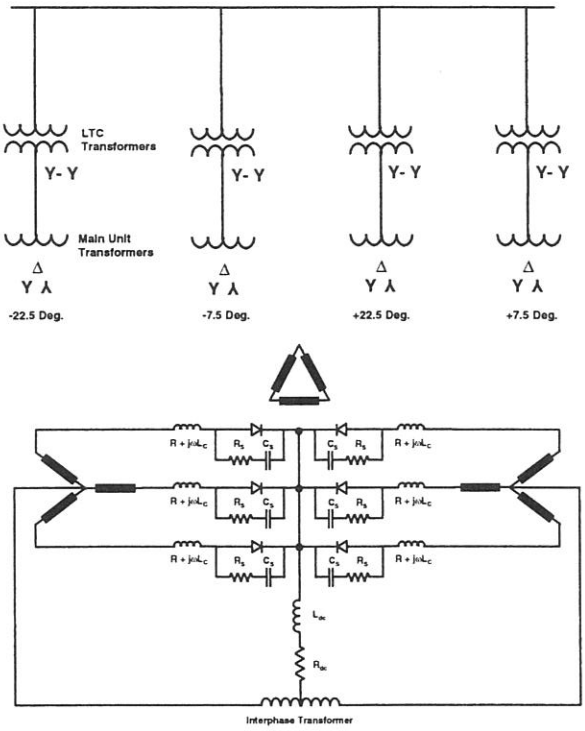

Figure 3: Converter Model

point-on-wave and fire both fault switches. The point-on-wave is determined by detecting the voltage wave zero crossing in the selected voltage signal and timing to the desired point on the signal based on nominal system frequency. The duration is measured from the point of inception. The second fault scenario is initiated once the first scenario is completed.

MATLAB-Based Graphical User Interface

User's interaction with PQ-SMART is strictly via the GUI. The GUI consists of many windows, with the main one being the one-line diagram. An example of a one-line diagram is in Fig. 7, which shows a realistic distribution network of 2 substations and more than 60 load nodes. On an actual screen, colors are used to denote multi- and single-phase feeders (lines). The user can zoom in/out the one-line diagram, draw lines, and add/remove devices. To inspect, specify or change parameters of a particular component, a double-click on the device icon opens the parameter window of interest. For example, the window for a feeder is as shown in Fig. 5. Most parameters can be changed by typing a new value in a labeled box, and the database is updated automatically. Some types of parameters, when changed, may trigger additional actions. For example, in the Feeder window, when a phase (a, b or c) is turned on or off, the corresponding feeder in the one-line diagram changes its color.

The Feeder window is among the simplest type of all GUI windows, and most windows fall into this category. However,

other windows have "special effects" to help the user visualize his/her action. An example is the advanced window for Harmonic Source, Fig. 6. Besides the boxes for the source rating, the window has two columns of "sliders" via which the user adjusts the harmonic amplitudes and the harmonic angles. The graph at the top of the window shows the resulting harmonic waveform, giving the user a clear picture of the wave to be injected into the network.

MATLAB-BASED POST-PROCESSOR OF SIGNALS

In addition to the EMTP library and the basic set of GUIs, the tool contains utilities that allow post-processing of signals that have been generated during an EMTP simulation. However, the post-processing feature is not restricted to EMTP-generated signals; field-recorded signals can also be processed after a data conversion.

With a doubleclick of the mouse button on top of a sensor icon, the voltages and currents recorded at the sensor's location are fetched and plotted on the screen. The user can zoom in on any portion of the waveform and apply a number of analysis functions to the displayed time window.

Power Spectral Density

This function employs the FFT (Fast Fourier Transform, a built-in MATLAB function) in the calculation of the power spectrum of the signal. The spectrum can be computed using all the data points, or just a selected window of data points; the choice is up to the user by zooming in or out of a section of the waveform.

The Total Harmonic Distortion (THD) is also computed for each phase and displayed on the graph automatically:

$$THD = \frac{\sqrt{A_2^2 + A_3^2 + \ldots + A_n^2}}{A_1}$$

where A_k's denote the harmonic amplitudes found by the FFT method.

Voltage Imbalance

The voltage imbalance V.I. is defined as the ratio between the negative- and positive-sequence voltages,

$$\text{V.}I. = \frac{|\overline{V}_2|}{|\overline{V}_1|} = \frac{|\overline{V}_a + (\overline{V}_b \cdot 1\angle 240°) + (\overline{V}_c \cdot 1\angle 120°)|}{|\overline{V}_a + (\overline{V}_b \cdot 1\angle 120°) + (\overline{V}_c \cdot 1\angle 240°)|}$$

where V_a, V_b and V_c are the voltage phasors of the three phases a, b and c (phase sequence is abc). The method for calculating the phasor based on data samples of the waveform is a standard one [4].

Voltage imbalance can be represented statistically in terms of cumulative probability and probability density. The cumulative $F(x) = \Pr\{\text{V.I.} \le x\}$ refers to the percentage of observation time that the imbalance V.I. is below a given level x. The probability density function is $f(x) = dF/dx$.

Voltage Sag/Swell

Voltage sags and swells refer to rapid deviation of the voltage amplitude from its nominal value. By default, a voltage is deemed acceptable if its amplitude falls within the [90%-105%]-range of nominal value.

The analysis function performs the following tasks concerning voltage sag/swell:

1. Tracking and displaying the signal amplitude as a function of time.
2. Sag-Level-versus-Duration characteristic. Each point on the characteristic represents the magnitude of the sag and the time duration over which the magnitude stays below that level. Such a characteristic is super-imposed on an industry standard such as CBEMA [5]. If the sag-level-versus-duration characteristic crosses the acceptable boundaries specified by the standard then the underlying voltage disturbance can be declared as violating the standard; otherwise, the disturbance is tolerable.

"Test-Your-Own-Algorithm"

The analysis functions listed in three preceding subsections are typical in a power-quality assessment. However, there are situations where the user wants to test his/her own functions or algorithms with the simulated waveforms. The menu "Test-You-Own-Algorithm" is designed to meet this needs.

To apply his/her test to the simulated waveform, the user first clicks on a particular sensor to initiate the sensor window and display the waveforms. The "Test-Your-Own-Algorithm" menu displays a list of all M-files (i.e., files written in MATLAB) available for testing. The user can select a file to edit it, or to execute it. The content of such a file is given in Figure 4. The first few lines of the file are to retrieve the waveform data `t, va, vb, vc, ia, ib, ic` that are displayed in the sensor window. These data are then operated upon by the user's algorithm. In the example, the apparent impedance for phase c is computed and plotted. The user can replace these lines with whatever that suits his/her needs. Of course, some familiarity with the MATLAB syntax is required.

EXPERIENCE AND CONCLUDING REMARKS

PQ-SMART was designed to support the R&D work of ABB. The tool integrates MATLAB and EMTP, two packages widely available at the company's research labs. It took about 5 man-months to design the software structure, GUIs, basic power-quality functions and debug the tool. The effort resulted in a software link to an extensive library of EMTP components that had been built in the past. PQ-SMART is now a fully functional package and does not require the user

to learn MATLAB or EMTP. The structure of PQ-SMART allows the design team to easily add new equipment models.

Beside power-quality analysis and algorithm testing, there are other important applications. Engineers who involve in transients studies at ABB-TTI have found this software a valuable tool because it has a wide variety of system components and it eliminates the laborious task of making and modifying systems. Sales representatives can run the tool from a laptop computer to demonstrate to their customers the operation and benefit of an equipment. Finally, the tool is not restricted to distribution and low-voltage systems (which are the initial goal); by expanding the component library, the tool can simulate events in the transmission system as well.

Even though PQ-SMART will always be restricted to applications within the company, the design experience can serve as an example for future EMTP development by other parties.

ACKNOWLEDGMENT

The authors thank their colleague Adam Yeh for providing realistic test data for the first prototype of PQ-SMART.

REFERENCES

[1] Electric Power Research Institute, EMTP Development Coordination Group, EPRI EL-6412-L: Electromagnetic Transients Program Rule book, Version 2.

[2] MATLAB, The Mathworks, Inc., 1993.

[3] J. Mahseredjian and F. Alvarado, "Creating an Electromagnetic Transients Program in MATLAB: MatEMTP," IEEE PES, 96 WM 098-4 PWRD.

[4] A. Phadke, J. Thorpe and M. Adamiak, "A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency and Rate of Change of Frequency," IEEE Trans. PAS, May 1983, pp. 1025-1038.

[5] ANSI/IEEE Std. 446-1987, "IEEE Recommended Practice for Emergency and Standby Power Systems for Industrial and Commercial Applications."

```
t  = getwave('time');          %Extract time,
va = getwave('volt','a');      %phase voltages
vb = getwave('volt','b');      %and phase
vc = getwave('volt','c');      %currents
ia = getwave('amp','a');       %from the
ib = getwave('amp','b');       %displayed
ic = getwave('amp','c');       %window
%Example:  Compute apparent impedance for
%          phase c and plot it.
Vc = tophasor(vc,t(2)-t(1),60,1);
       %Convert phase-c voltage to phasor
Ic = tophasor(ic,t(2)-t(1),60,1);
       %Convert phase-c current to phasor
Zc = Vc./Ic;
figure;
plot(Zc)
xlabel('R, phase c')
ylabel('X, phase c')
title('Impedance trajectory seen on phase c')
```

Figure 4: An example of "Test-Your-Own-Algorithm"

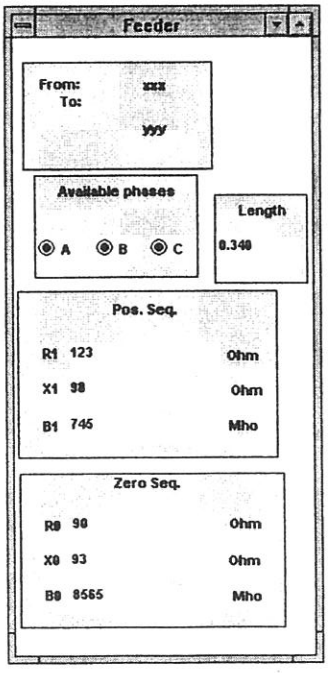

Figure 5: Window for Feeder.

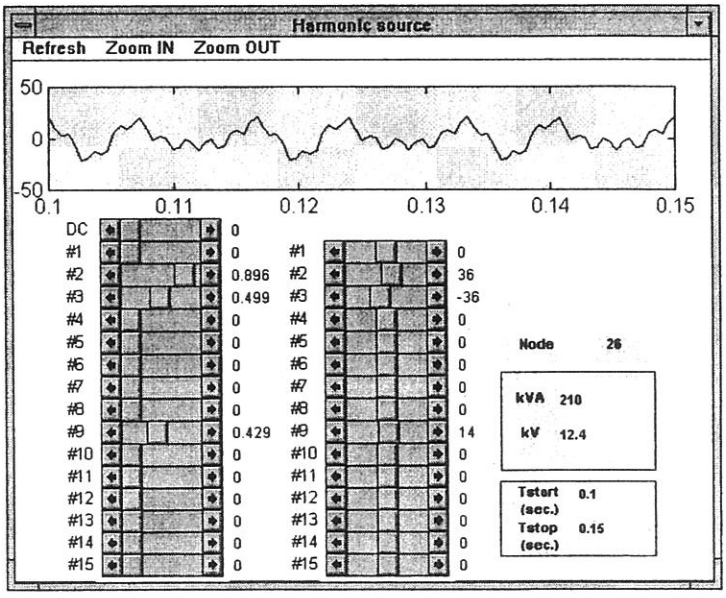

Figure 6: Window for Harmonic Source.

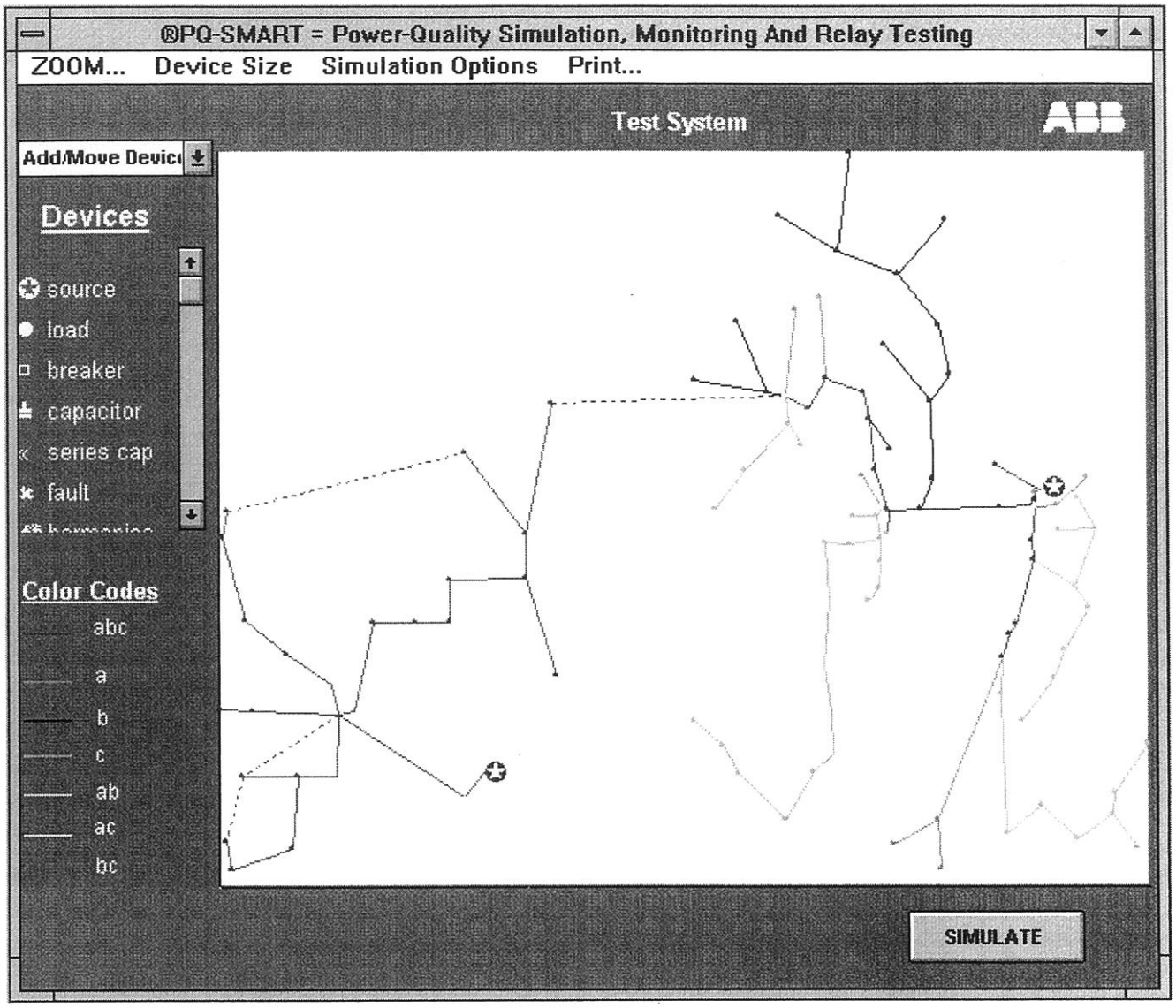

Figure 7: One-line diagram of a test system.