Virtual lab for power system simulation

Part 1 Interactive design of power networks and load-flow analysis

Improvements in the speed of PC microprocessors have enhanced the possibilities of building PC-based simulation tools in the area of power systems. This article describes a low-cost PC-based tool that can be used in college laboratory simulation of networks and systems, as a continuing training programme for professional engineers and as a system design and analysis tool. The tool provides the user with the capability of drawing one-line diagrams and interactively enters component parameters and saves data into a file. The tool also includes load-flow analysis packages using the methods available (Gauss-Seidel, Newton-Raphson, Jacobi, Fast Decoupled) for comparative purposes in laboratories and training sessions.

by J. M. Ngundam, E. R. Ngalemo Ngalemo and F. Kenfack

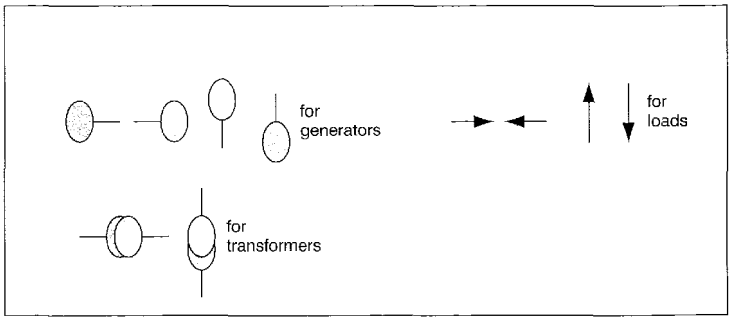
Installation of power engineering laboratories in engineering schools and training institutions in general is costly and takes up enormous amounts of space. Given these difficulties, practical work on power systems is limited if at all to a few experiments that are less costly in terms of equipment and space. For developing countries in particular, additional difficulties such as limited availability of experts in the area must be taken into account. Developments in electrical engineering are attracting many more undergraduates to the glamorous areas of computer engineering and science, computer/telecommunications/information technology and control engineering, and not power systems. It is likely that the more developed countries are experiencing these difficulties to a far lesser extent.

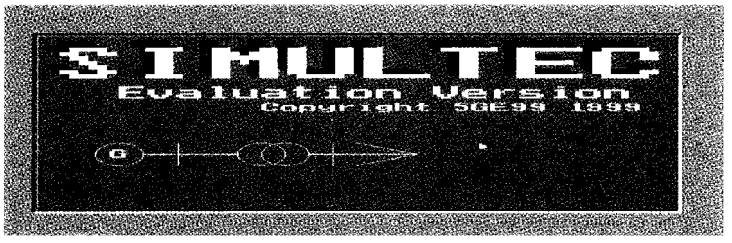
A possible solution to the above difficulties is to develop a virtual power systems laboratory for use in practical work, training and system simulation and design. In order to cut costs and enhance its affordability even to poor countries, the tool presented in this article has been developed to run on low-performance computers as well.

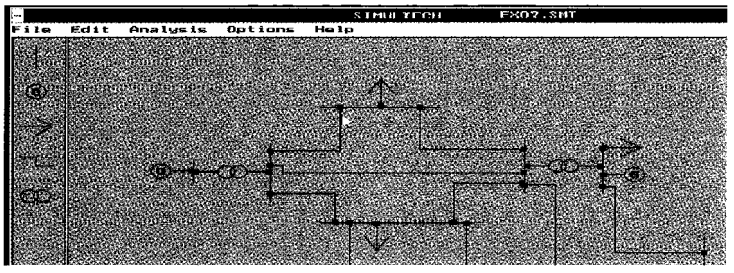
This article therefore presents a new PC-based graphical interface for analysing power systems networks. Its capabilities include:

- drawing one-line diagrams
- data entry of device parameters via simple click on device symbols
- use of several methods (Gauss, Gauss Seidel, Newton-Raphson, Jacobi and Fast Decoupled).

1 Power system component symbols







2 Illustration of first screen

3 Example of a network designed using the symbols shown in Fig. 1

The tool has been developed with Turbo Pascal as the programming language. The program runs on Microsoft DOS Version 4.0 and higher. It can run on IBM PCs or compatibles. Although Fortran compilers have also developed visual capabilities, it is not available to us. In the future, it might be useful to develop a similar tool in Fortran for comparison purposes. The objective in doing so should be to determine the best programming approach in terms of speed, accuracy etc.

A user manual to make it easy for users to undertake simulation and system design as described in this article accompanies this tool.

Representation of the network

Electric power networks are generally represented in the form of one-line diagrams with each element connected to a bus and buses linked through lines. A network is composed of generators, transformers, buses, lines and loads. Particular elements of real networks such as breakers are not of interest now because such elements are passive with no particular effect on system analysis. They are considered as part of the control system. Each component has a name and characteristics. Thus we have power (active and reactive), resistance and reactance for generators, resistance and reactance for transformers and lines, power (active and reactive) for loads and for buses we have just power. However, their nominal voltages and currents further describe all of these elements. Characteristics are transformed into per unit values to simplify calculations

User interface

Accessibility of input and output data is made simpler by drawing the network on the screen by selecting any element symbol individually. In this way, input data can be taken from a menu command when a component is selected. In order to achieve this, we have designed a graphical interface with the following properties:

- Each component has a graphical representation and an internal set of parameters.
- Each component is created through a simple click in the work field after its style has been selected in a toolbox.
- Each component can be repositioned by dragging it to new locations.
- A device can be connected to a bus by bringing the end of the device near the bus. This depends on the orientation of the components For instance, a horizontal generator can not be connected to a horizontal bus.
- Each component can be selected through a simple click on the graphical symbol representing it.

Input and output data can be accessed in a window created from a menu command with the possibility of editing input values. Data can be saved in a database, which consists of one text file. Internal data representations and manipulations are completely transparent to the user. The tool includes the possibility of panning horizontally and vertically across the diagram. This is accomplished merely by moving the mouse at the edge of the design screen. Copying, cutting and pasting features are also available from the 'edit' menu command. The Y matrix of a network can be shown from the 'analysis/Y matrix' menu command. Analysis parameters such as convergence error or initial values can be changed through the 'options/initial values' menu command. Particular information relating to the method of analysis is shown after each analysis. Thus, the number of iterations, the computation time and the memory required are shown for each method after analysis. Via the integrated help engine, the user is quickly provided with an online help, oriented diagram manipulations, analysis and component parameters.

Implementation

The program was developed in Turbo Pascal Version 5.5 environment. We chose this en-

vironment because it is a simple programming language that we are used to, and it provides a large number of routines for graphics management. Furthermore, a module for window interface representation was already developed in this language.

Each component is internally represented as a set of records representing co-ordinates, input and output parameters. It has the following form:

- component = record
- co-ordinates
- input parameters
- output parameters
- style.

Components are drawn according to their style and direction and, when selected, with a different colour and marker surrounding it. The shapes are as shown in Fig. 1.

Line shapes are automatically generated and depend on positions and the directions of buses linked by the line. The set of components is represented as a chained list. Therefore, the maximum number of components depends only on the memory capacity of the computer.

A module for complex calculations was developed to enable Y matrix computation. A complex quantity is defined as a set of two real values as follows:

complex = record x, y: real end

The module contains basic operations on the complex quantity. They are addition, subtraction, product, conjugate, division.

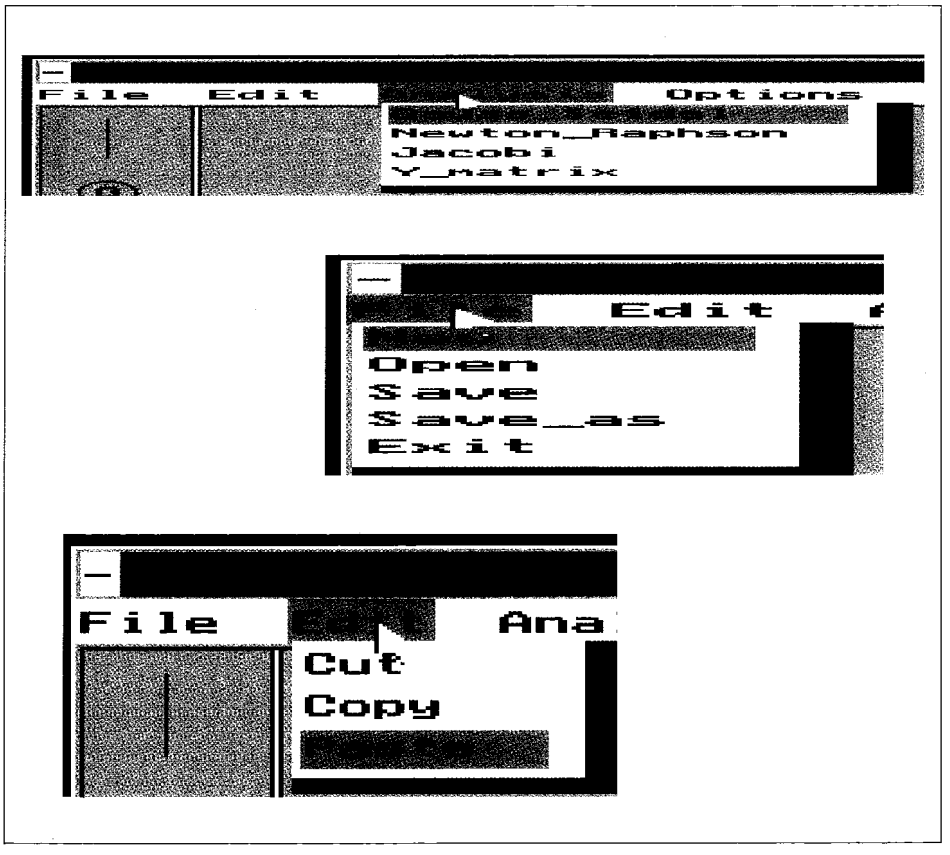
This tool has been named SIMULTEC. It is software for DOS, which is in the form of a standalone executable program associated with a help and a graphic interface file. It runs on a computer with the following minimal characteristics:

- 300 kbyte of hard disk space
- 286 processor with an arithmetical co-processor
- 640 kbyte of RAM

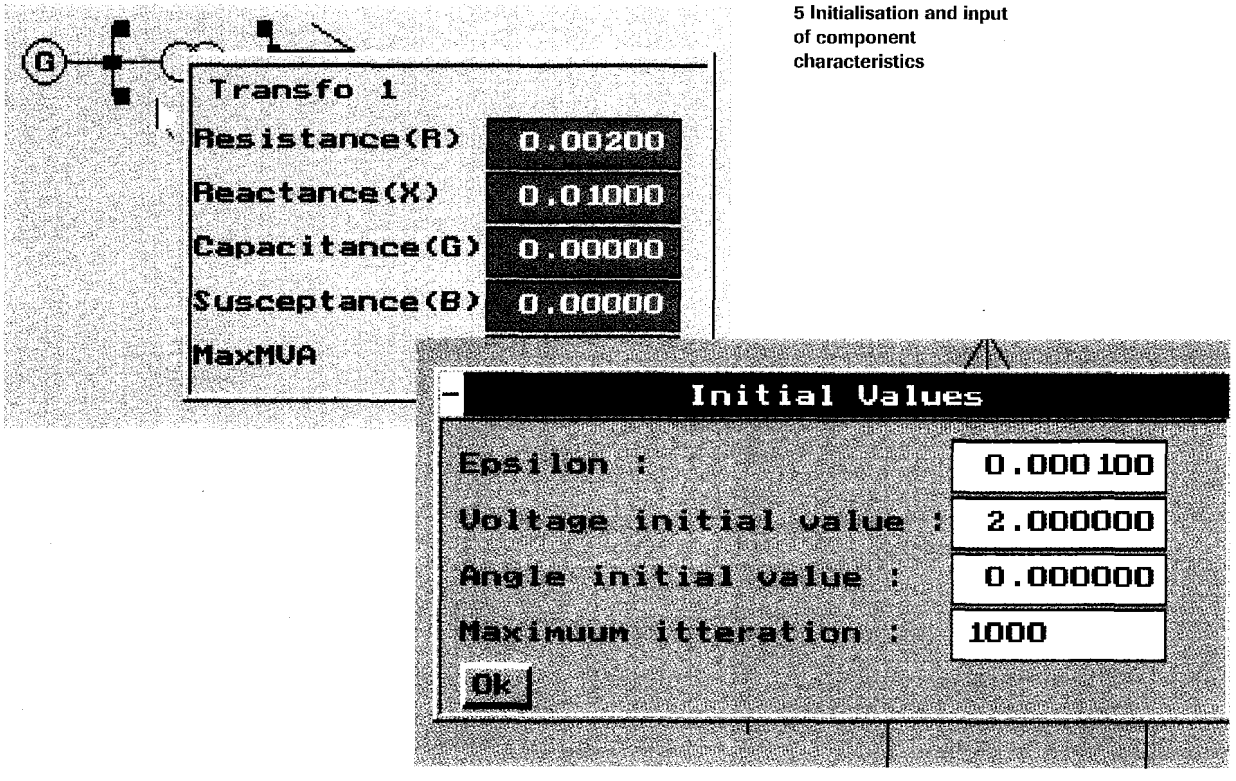
Example of use of SIMULTEC

Design of the network

From the file menu select 'new file': the work



4 Menus of options



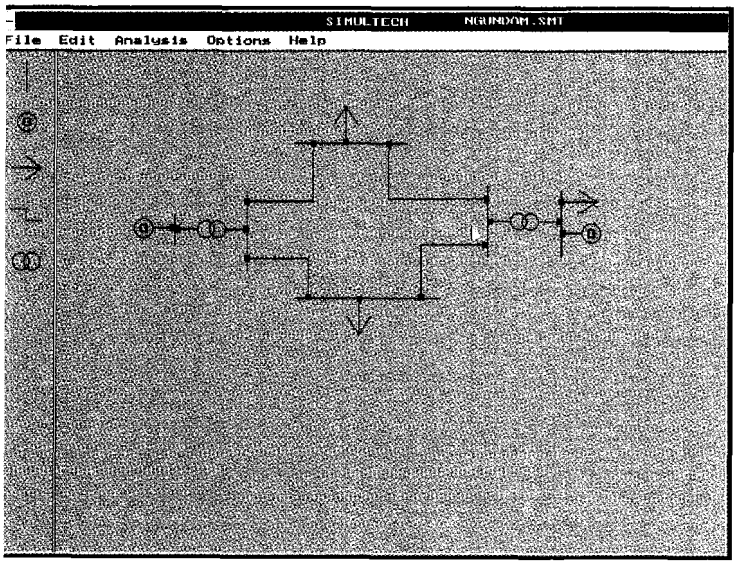
5 Initialisation and input of component characteristics

field is blanked. In the toolbox, click on generator, then click in the work field: a generator is created at the position where you want it. Do the same for other components. Double click on each element and enter its parameters in the dialogue box that appears.

Run analysis

Select the analysis preferred from the options menu. In the analysis menu select the method. After completion of the analysis, the results are given in a dialogue box. From the file menu select 'save' and enter a filename in the dialogue box that appears (Figs. 2-5).

6 Example of application to a network

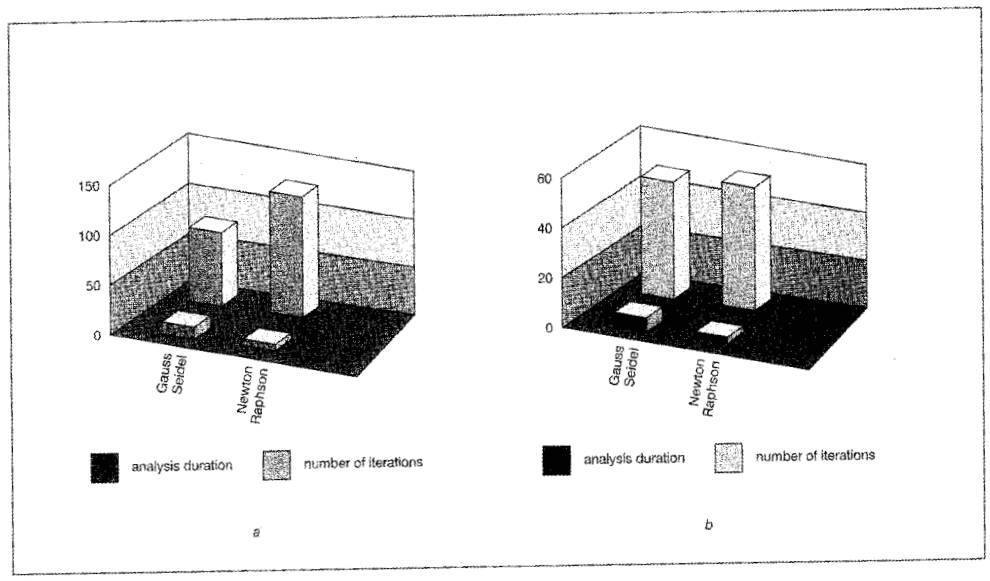


Application

The network in Fig.6 has been analysed using the methods described above and the results of two of the important methods are provided in two bar charts (Fig. 7) for two different tolerance limits. Memory space required is 2·2kbyte.

Comment on the results

The charts in Fig. 7 show, for analysis of just two methods, that for a lower tolerance limit, the Newton-Raphson method has a shorter duration of analysis and the number of iterations are almost double that obtained with the Gauss-Seidel method. For a higher tolerance limit, even though the duration of analysis is still lower for the Newton-Raphson method, the number of iterations is about the same in this case compared to the Gauss-Seidel method. In fact, these results are in conformity with theory. The Newton-Raphson method is a powerful method for solving non-linear algebraic equations. It works faster and is sure



7 Results of calculations for different tolerance limits: (*a*) Iteration tolerance of 0·0001; (*b*) Iteration tolerance of 0·001

to converge in most cases as compared to the Gauss-Seidel method.[2] Although results of load-flow analysis with the fast decoupled method, a modification of the Newton-Raphson approach, have not been included, this tool, however, includes the possibility of undertaking load-flow analysis using the method.

Conclusion

We have developed a tool for power system load-flow analysis with a graphical interface designed for training purposes, network design and extension. The program was developed for the DOS mode, which is real, and faster than the projected mode. It requires very little memory. However, graphical interfaces required far more memory, leading to the impossibility of generating very large networks. These difficulties do not, however, reduce its usefulness as a laboratory tool for training students. In any case, for more professional purposes, the use of high-performance computers should eliminate these difficulties. The tool could be extended to include short-circuit analysis of networks, and the final objective is to develop a real-time system monitoring and operational control tool for security enhancement and load management.

Comparison of results obtained with Turbo Pascal and those obtained with Fortran have been left for further development work. In its present state, the tool is quite useful for training technicians and student engineers. Because there is a shortage in skilled manpower in the area of power systems analysis and design in developing countries, this tool may be able to solve the shortage problem by providing a means for rapidly training experts in the field.

Bibliography

1 YU, D. C., FLINN, D. G. and KRIEGER, R. A.: 'Facilitating engineering analysis via a graphical database', *IEEE Transactions on Power Systems*, February 1995, **10**, (1)
2 NAGRATH, I. J., and KOTAHARI, D. P.: 'Modern power systems analysis' (Tata McGraw-Hill Publishing Company Ltd., New Delhi, 1993, 2nd edn.)
3 Reliability task system task force, 'IEEE Reliability Test System', *IEEE Transactions on Power Apparatus and Systems*, November/December 1979, **PAS-98**, (6),
4 FRANKLIN, A. C., and FRANKLIN, D. P.: 'The J & P transformer book' (Butterworth, 1983, 11th edn.)
5 NGUNDAM, J. M.: 'Reliability techniques in power system planning', Ph.D thesis, Imperial College of Science and Technology, London, 1982
6 'CYNFLOW, Guide d'utilisation et manuel de reference' (CYME International Inc., 1991)

The authors are with the Automation and Control Laboratory, Ecole Polytechnique, University of Yaounde I, BP 8390, Yaounde, Cameroon. Tel.: +237 230113; Fax: +237 230103; e-mail: ngundam@polytech.uninet.cm