

COMPREHENSIVE AUTOMATED SYSTEM FOR STUDYING THE THEORY OF ELECTRICAL CIRCUITS

Gudkov A., Shylin L.

Department of Information Technology Automated Systems,
Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus

E-mail: gudkov_fit@mail.ru, dekgitu@bsuir.by

The article considers the implementation of a package of application software that performs a number of specific tasks for students and teachers of disciplines related to the studying of electrical circuits in higher educational institutes. The C++ programming language technologies were used with Qt5 framework, window of WinForms with a graphical context OpenGL.

INTRODUCTION

In the process of studying the disciplines related to the calculation of electrical circuits, students of higher educational institutes sometimes have difficulties finding tasks on a certain topics for practicing them, and the teachers have the need to manually check the students' solutions.

The authors developed a comprehensive automated system for studying the discipline "Theory of Electrical Circuits". This system consists of three independent modules, which are the software packages that perform a number of specific tasks. The first two modules perform the generation of electrical circuit data and its schemes for students. And the third one perform the calculation of electrical circuits with providing solutions for teachers.

I. SYNTHESIS OF ELECTRICAL CIRCUIT COMPONENT DATA FOR TRAINING TASKS

For ease of generating, we will represent an electrical circuit in the form of a directed graph formed according to certain rules [1]. The directions of its branches coincide with the directions of the currents in the branches wires, and the vertices coincide with the nodes of the electrical circuit scheme. In our program you need to set initial conditions such as:

- range of values for the resistances;
- range of values for current voltage sources;
- quantity (range) of current and voltage sources;
- quantity (range) of resistances;
- number of equations to be solved by the method of contour currents;
- number of equations to be solved by the method of nodal potentials.

Based on the calculation method and the number of equations to solve, the program starts to synthesize the chain graph. Next, the resistance values are generated by random variables from the preconditions. We perform similar actions for the voltage and current sources. You also need to find out the number of branches in the circuit based on the number of voltage and current sources. After that, using the randomize method, we arrange the initial ele-

ments of the chain in any form, observing certain rules.

As a result, we obtain a model of an electrical circuit represented as a directed graph. According to its graph, we get an electrical circuit data, which is presented in format of a text table that is convenient for exporting files between applications (formats like *TXT*, *CSV*, *JSON*, *XML*). The data table shows the following circuit parameters:

- number of branches;
- direction of currents in branches from the start node to the end node;
- resistances values;
- voltage source values;
- current source values.

In addition to generating direct current (DC) circuit data the program generate data of alternating current (AC). This way the data table is extended by Complex values by addition the imaginary part into resistances of capacitors and inductance coils, as well as arguments for voltage and current sources.

II. PROCESSING OF COMPONENT DATA AND ITS REPRESENTATION IN THE FORM OF ELECTRICAL CIRCUIT SCHEME

After generating file with data table of circuit the program process it and presents as an image of the electrical circuit scheme. The first step of the graphical visualization algorithm is converting component data into a mathematical model – a set of structures that store the begin node and the end node of a current branch, and data matrices that store the characteristics of branches. When we read the data from a table it is effective to immediately form component matrices, since each column of the table reflects a specific branch of the circuit.

A universal class is implemented for data obtained from component matrix information in a text document for both AC and DC circuits. In the case of direct current, the complex fields (such as imaginary parts and arguments) of the object have the *NULL* value.

A class for graphical visualization was also implemented. It describes an algorithm for graphical representation of a mathematical model using the

capabilities of the programming language *C++*, and also uses the capabilities of its graphical framework *Qt5*, especially its library – *QtGL* [2]. The result is provided in the *WinForms* main application window with a *OpenGL* graphical context, the menu of which allows you to print graphic diagrams with export them to *PDF* file or save it as a binary image [3].

III. CALCULATION THE ELECTRICAL CIRCUITS

After circuit data visualization in the form of schemes for students the software calculates these schemes and show the solutions for teachers. In this problem, the mathematical model of an electrical circuit is a set of matrices that are divided into two types [4]:

- topological matrices that characterize the structural features of the circuit, determined by the method of connecting the basic components of two-poles, while the type of components doesn't play any role;

- component matrices that reflect the values of the circuit components.

The currents in the circuit are calculated based on the matrix method of nodal potentials, which is valid for both DC and AC circuits [5]. Firstly a topological nodal matrix is formed based on the directed graph. It is very convenient to take the last generated node of the graph as the basis node (t.i. remove the last vertex from consideration), which is a significant advantage for the software implementation of this approach over the method of contour currents – there is no need to search for independent circuit contours.

Given that each branch of the circuit can be represented as a generalized one, component column-matrices are formed with source data of the chain for E , J , R , and the diagonal-matrix RD , which dimensions are equal to the number of branches of the circuit. The representation of an electrical circuit in the form of a mathematical model allows us to describe its calculation on the algorithmic way, which can be represented as a block diagram (Fig. 1).

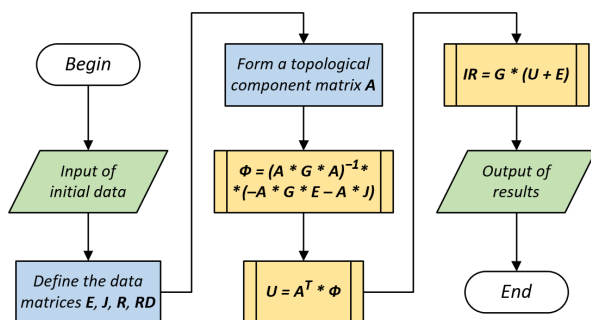


Figure 1 – Block diagram of the nodal potentials method algorithm

An Object-Oriented Programming approach allows us to display the abstraction of matrix calculations in the application software. Classes *MatrixFloat* and *MatrixComplex* (inherited from the base class *Matrix<T>*), fully represent the logic of mathematical operations for real-valued (*Float*) and Complex-value (*Complex*) matrices, respectively. They contain the dimensions of the matrix (*int rows, columns*), its order (*int order*) and a two-dimensional dynamic array of element values (*T buffer[rows][columns]*) Based on the formulas and theorems of matrix calculus (like Laplace, Jordan-Gauss, etc.), the necessary algorithms were implemented and overloaded corresponding arithmetic operations in the program code. To calculate the sinusoidal current circuit, a Complex matrix element is introduced *MatrixComplex* – class *Complex*, it allows performing transformations on Complex numbers and their interactions in both algebraic and exponential forms [6].

In the class-interface of an electrical circuit *Circuit* the algorithms described earlier are implemented in the form of a program code-generalized matrix equations of the nodal potentials method. The return value of the method *CircuitCalculator::CalculateCircuit(const E)* is the result of calculating the electrical circuit – matrix-column IR of currents in the branch resistances.

CONCLUSION

As a result of the work, the authors implemented a package of application software that performs a number of specific tasks for students and teachers of disciplines related to the study of electrical circuits in higher educational institutes. Software efficiency is determined by the ability to generate arrays of practical tasks of both DC and AC electrical circuits schemes with any topology, as well as to obtain the most accurate values of their calculations in the shortest possible time.

REFERENCES

1. Артым, А. Д. Новый метод расчета процессов в электрических цепях / А. Д. Артым, В. А. Филин, К. Ж. Есполов. – СПб.: «Элмор», 2001. – 192 с.
2. Lazar, G. Mastering Qt 5.X – Second Edition: Create stunning cross-platform applications using Qt, Qt Quick, and Qt Gamepad / G. Lazar, R. Penea. – Packt Publishing, 2018. – 613 p.
3. Gordon, V. S. Computer Graphics Programming in OpenGL with C++ / V. S. Gordon, J. Clevenger. – Mercury Learning & Information, 2018. – 384 p.
4. Harary, F. Graph theory / F. Harary. – CRC Press, 1994. – 288 p.
5. Атабеков, Г. И. Теоретические основы электротехники / Г. И. Атабеков. – М.: «Энергия», 1978. – 592 с.
6. Lafore, R. Object-Oriented Programming in C++, Fourth Edition / R. Lafore. – SAMS Publishing, 800 East 96-th St., Indianapolis, Indiana 46240 USA, 2002. – 1038 p.