An Algorithm to Solve Unbalanced Ladder Circuits

(Developed by Joe Nathan Abellard)

Table of Contents

I. Introduction

In this project, I built a specialized Java application capable of solving a purely resistive ladder circuit. Building an application capable of solving any kind of electrical circuit is immensely complex and requires a lot of coordinated team effort, and sustained dedication. I decided to not explore such a complex problem not because of a lack of ambition but for because of a lack of resources and the time constraint of one semester.

II. Objective

My objective was to build a Java application capable of solving a ladder circuit for the following quantities:

1. Total Resistance
2. Source Current
3. Current across each branch
4. Voltage across each branch
5. Power dissipated by each branch
6. Conductance of each resistor

III. Theory

A ladder circuit is a special type of electrical circuit with the configuration illustrated below in figure 1.

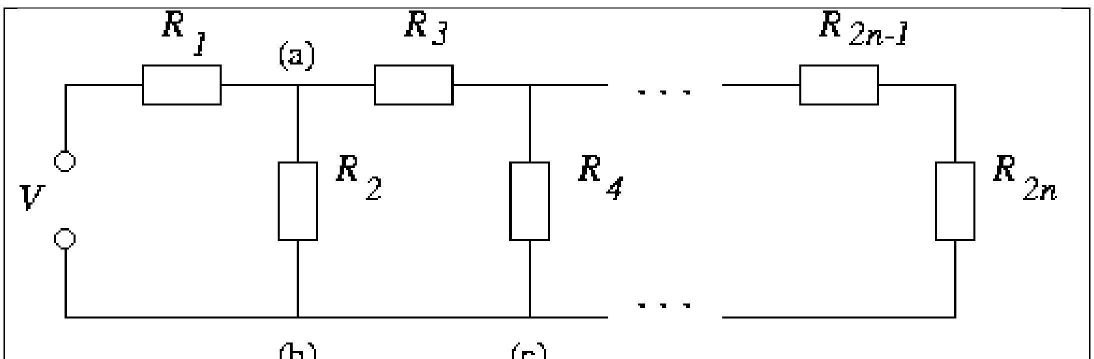

Fig. 1: Configuration of a purely resistive ladder circuit.

As aforementioned, building an application akin to Multisim that is capable of solving a broad range of electrical circuits is immensely complex. However, building the specialized Ladder Circuit Solver is a more narrowed, but still a very challenging problem. That is because a ladder circuit has some special properties. By exploring these properties to seek out patterns, formulas can be derived. Consequently, those formulas can be implemented in a programming language such as Java. The circuit (purely resistive ladder circuit) of interest has the following properties:

(a) Contains a voltage source (Vs)
(b) Contains an even amount of resistors
(c) Contains at least 4 resistors

These properties served as the initial point of analysis for the project. From there, I investigated further by analyzing multiple circuits to seek out patterns. Here is what I found:

1. Total Resistance:
   Before the total current (source current – Is) of the circuit can be computed, the total resistance of the circuit must be known. A ladder circuit is series-parallel circuit. Therefore, the total resistance cannot be immediately calculated, as is the case for a one-looped purely resistive series circuit. The total resistance is calculated by first computing a series of intermediate resistances. The question, then, is, how many of these intermediate total resistances must be computed before the final total resistance can be determined. For a ladder circuit with n resistors, the number of intermediate total resistances that must be calculated is n-1. The (n-1)th total resistance is the final total resistance of the circuit.

   Special Case: for RTx such that x = 1

$$RT_1 = R_n + R_{(n-1)}$$

   For RTx such that n is even:

$$RT_x = \frac{\left(RT_{(x-1)}\right)\left(R_{(n-x)}\right)}{\left(RT_{(x-1)}\right) + \left(RT_{(n-x)}\right)}$$

   For RTx such that n is odd:

$$RT_x = RT_{(x-1)} + R_{(n-x)}$$

2. Current Through each branch (resistor)

The manner in which the current through a branch is calculated depends on whether or not the resistor connected to that branch is an even-numbered resistor or an odd-numbered resistor.

For Rx such that n is even:

$$IR_x = \frac{\left(IR_{(x-1)}\right)\left(RT_{((n-1)-1)}\right)}{\left(RT_{((n-1)-1)}\right) + (R_x)}$$

For Rx such that n is odd:

$$IR_x = \frac{\left(IR_{(x-2)}\right)\left(RT_{(x-1)}\right)}{\left(RT_{(n-x)}\right) + (R_{(x-1)})}$$

3. Total Current:
   Once the total resistance of the circuit is known, the total current of the circuit can be determined using the following formula:

$$I_s = IR_1$$

4. The voltage through a resistor:

$$VR_x = (R_x)(IR_x)$$

5. The power dissipated by a resistor:

$$PR_x = (VR_x)(IR_x)$$

6. The conductance of a resistor:

$$GR_x = \frac{1}{R_x}$$

7. The total resistance of the circuit:

$$R_T = RT_{(n-1)}$$

IV. Java Implementation

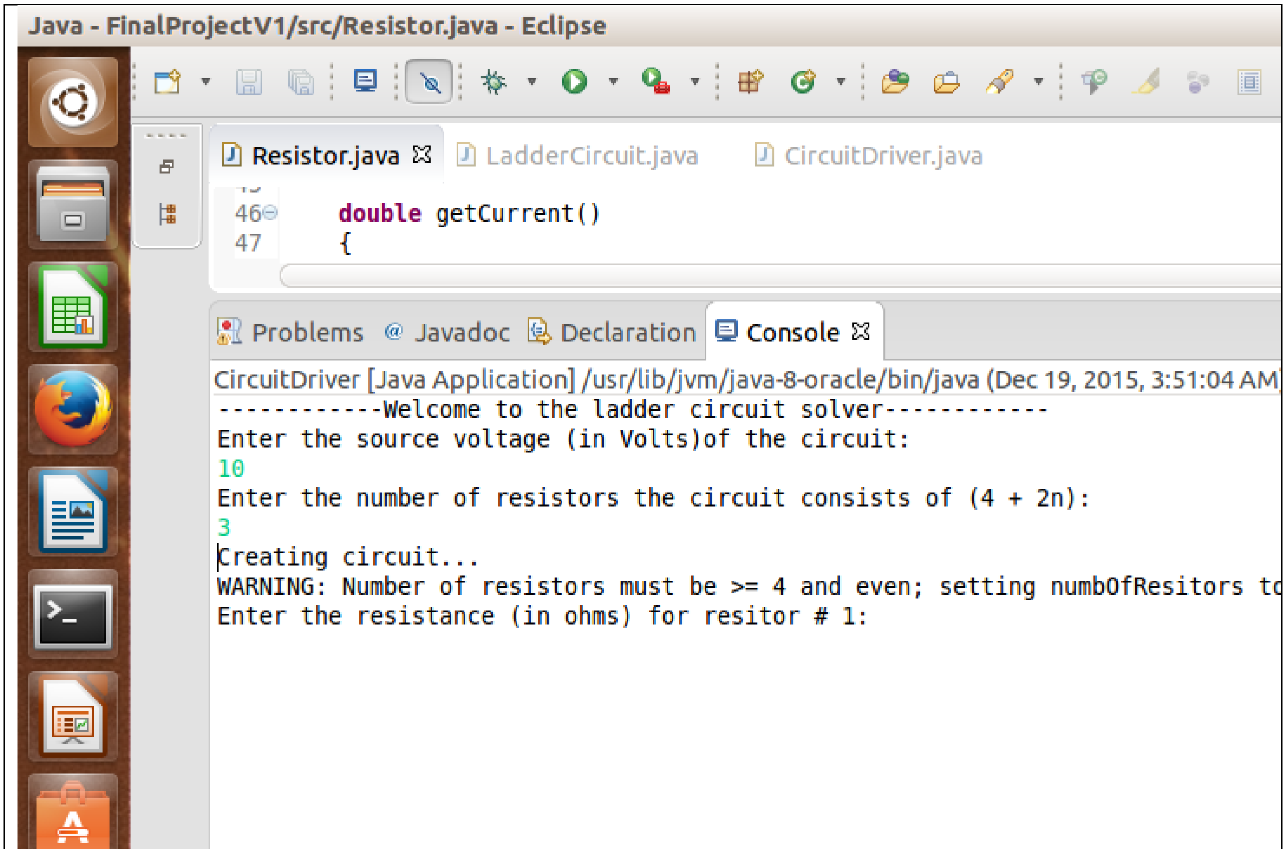

Fig 1: VALIDATION OF RESISTOR NUMBER: Number of resistors must be greater than or equal to 4.

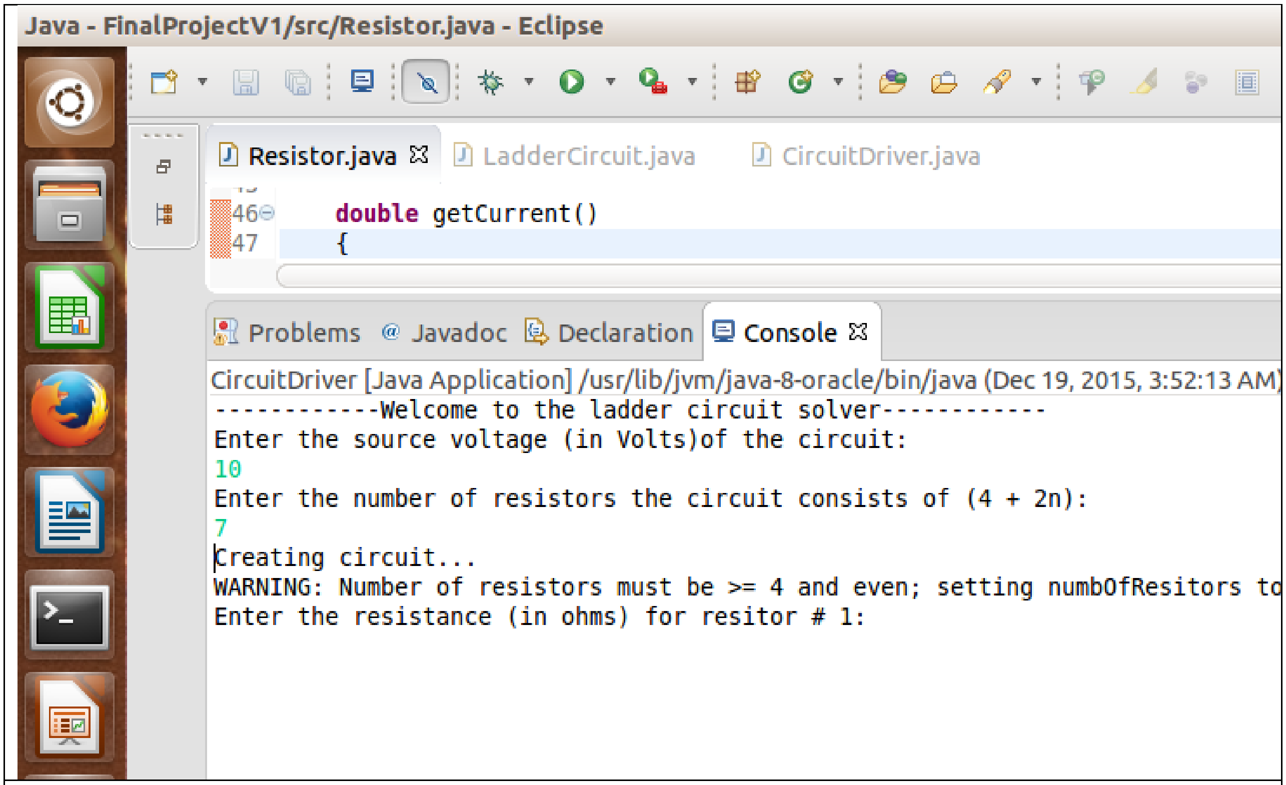

Fig 2: VALIDATION OF RESISTOR NUMBER: Number of resistors must be even.

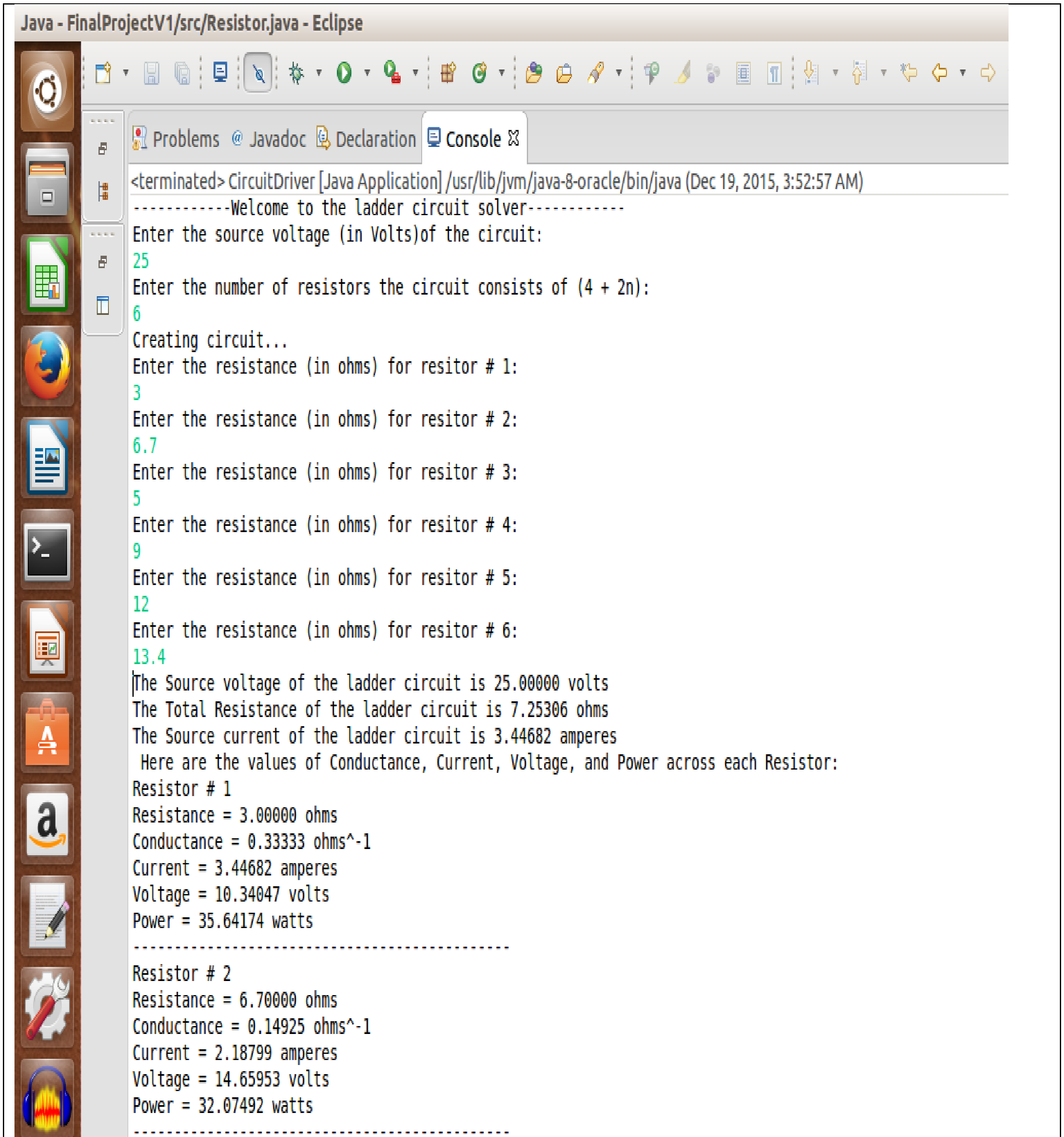

Fig 3: SAMPLE RUN: Number of resistors = 6

Fig 4: SAMPLE RUN: Number of resistors = 6 (continued)

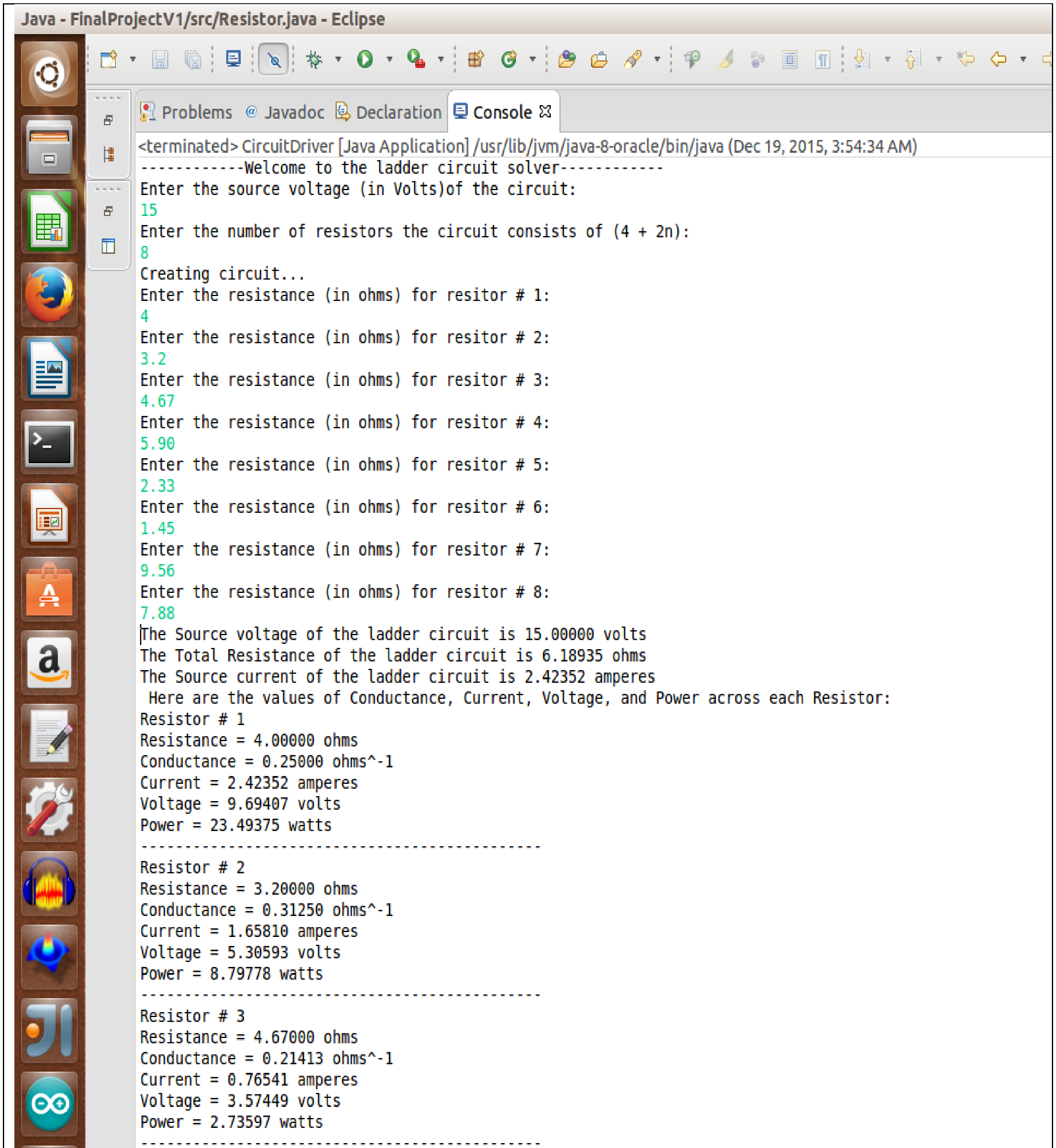

Fig 5: SAMPLE RUN: Number of resistors = 8

```
---------------------------------------------
Resistor # 3
Resistance = 4.67000 ohms
Conductance = 0.21413 ohms^-1
Current = 0.76541 amperes
Voltage = 3.57449 volts
Power = 2.73597 watts
---------------------------------------------
Resistor # 4
Resistance = 5.90000 ohms
Conductance = 0.16949 ohms^-1
Current = 0.29346 amperes
Voltage = 1.73144 volts
Power = 0.50812 watts
---------------------------------------------
Resistor # 5
Resistance = 2.33000 ohms
Conductance = 0.42918 ohms^-1
Current = 0.47195 amperes
Voltage = 1.09964 volts
Power = 0.51898 watts
---------------------------------------------
Resistor # 6
Resistance = 1.45000 ohms
Conductance = 0.68966 ohms^-1
Current = 0.43572 amperes
Voltage = 0.63180 volts
Power = 0.27529 watts
---------------------------------------------
Resistor # 7
Resistance = 9.56000 ohms
Conductance = 0.10460 ohms^-1
Current = 0.03623 amperes
Voltage = 0.34633 volts
Power = 0.01255 watts
---------------------------------------------
Resistor # 8
Resistance = 7.88000 ohms
Conductance = 0.12690 ohms^-1
Current = 0.03623 amperes
Voltage = 0.28547 volts
Power = 0.01034 watts
---------------------------------------------
```

Fig 6: SAMPLE RUN: Number of resistors = 8 (continued)

V. Remarks/ Next Steps

This project afforded me the opportunity to apply all of the concepts learned in my Introduction to Java class. It was a very challenging project that kept me at the edge of my conform zone, and encouraged me to explore new knowledge. The next iteration of this algorithm will be more generalized. Consequently, it will be capable of solving RLC (Resistor, Inductor, Capacitor) AC circuits, in which the electrical properties of the circuits elements are represented by vectors (i.e., impedance values), instead of simple scalar quantities.