Integrated Masters in Aerospace Engineering, Técnico, University of Lisbon

Circuit Theory and Electronics Fundamentals

Laboratory Report 1

Grupo 21:

95791, Francisco Carvalho

95805, João Matias

95846, Simão Gonçalves

March 25th 2021

Índice

1 Introduction

For the first experimental activity in the Circuit Theory and Electronics Fundamentals course, we were given a simple circuit to analyse, in relation to the contents presented in the first three weeks of the semester. This circuit is constituted by a dependent and independent voltage source, a dependent and independent current source and 7 resistors (fig. 1). In order to solve the circuit we performed a theoretical analysis, using the node and mesh methods (Section 2), and a simulation, using the software Ngspice (Section 3). We then compared both results and tried to validate our theoretical analysis (Section 4).
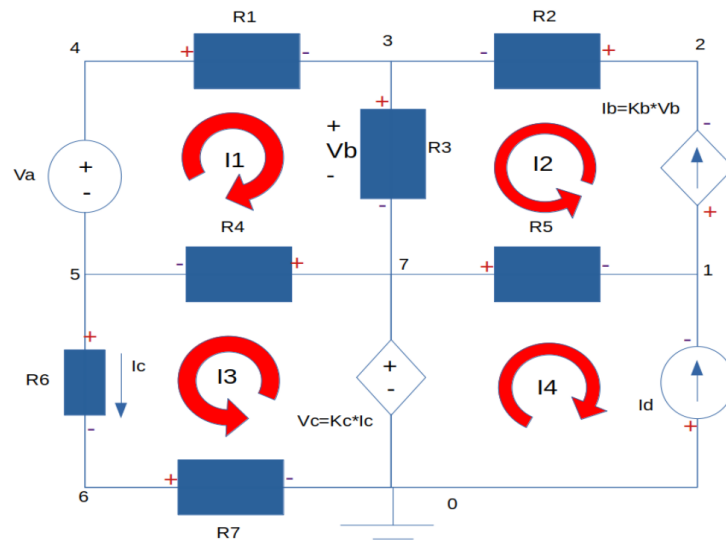

Figura 1: Analyzed Circuit

To obtain the initial data, we ran a python script provided by our teacher, which generated the following data:

Name	Generated Data
R1	1.04899828982
R2	2.0555682231
R3	3.06018802573
R4	4.16897961659
R5	3.07395007732
R6	2.0428100493
R7	1.03756256625
Va	5.11422921556
Id	1.01155034907
Kb	7.33855517177
Kc	8.2581000183

Tabela 1: Inicial data. Resistors and Kc in kOhm, Voltage in V, Current in mA and Kb in mS

2 Theoretical Analysis

In this section, we expose and explain the methods used to solve and analyse the circuit. In this specific case: the Mesh and the Node Methods.

2.1 Mesh Method

The Mesh Analysis consists in defining circulating currents in all the simple meshes, and then calculating them. So, with this method we determine the currents I1, I2, I3 and I4. This results were achieved looking to the loop formed by R1, R3, R4 and Va and other loop formed by R4, R6, R7 and Vc. The circulating currents are I1 and I3, respectively. The third independent equation was obtained equaling I2 to Kb*Vb and Vb to (I2+I3)*R3. The last and fourth equation is a very simple one, we just equal I4 to -Id. To solve this system of linear equations we rearranged them in the matrix form below, and we obtained the solution using Octave math tools.

$$\begin{bmatrix} R1 + R3 + R4 & -R3 & -R4 & 0 \\ -R4 & 0 & -Kc + R4 + R6 + R7 & 0 \\ -Kb*R3 & Kb*R3 - 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I1 \\ I2 \\ I3 \\ I4 \end{bmatrix} = \begin{bmatrix} Va \\ 0 \\ 0 \\ -Id \end{bmatrix}$$

Name	Value [A]
I1	0.00022930
I2	-0.00023999
I3	0.00094758
I4	-0.00101155

Tabela 2: Circulating currents obtained through the application of the Mesh Method in Octave

	Resistor's Current
R1(i)	2.2930e-4
R2(i)	-2.3999e-4
R3(i)	-1.069e-5
R4(i)	1.17688e-3
R5(i)	-1.25154e-3
R6(i)	9.4758e-4
R7(i)	9.4758e-4

Tabela 3: Current values through each Resistor

2.2 Node Method

The objective of this method is to determine every node voltage. To do this we have to first number nodes arbitrarily, assign potential 0V (Ground) to one of the nodes, and then calculate all the voltages. We had 7 unknown variables, so we determined 7 independent equations. This equations were obtained doing KCL in nodes 1, 2, 3, 6 and 7; knowing that Va=V4-V5; and using the fact that V7 equals Vc, that in turn equals Kc*Ic. The equations were rearranged in the matrix form below, in order to find the solution using Octave math tools.

$$
\begin{bmatrix}
G5 & 0 & Kb & 0 & 0 & 0 & -Kb-G5 \\
0 & G2 & Kb-G2 & 0 & & 0 & Kb \\
0 & -G2 & G1+G2+G3 & -G1 & 0 & 0 & -G3 \\
0 & 0 & 0 & 0 & -G6 & G7+G6 & 0 \\
-G5 & 0 & -G3 & 0 & -G4 & -G7 & G3+G4+G5 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & Kc*G7 & -1
\end{bmatrix}
\begin{bmatrix}
V1 \\ V2 \\ V3 \\ V4 \\ V5 \\ V6 \\ V7
\end{bmatrix}
=
\begin{bmatrix}
Id \\ 0 \\ 0 \\ 0 \\ -Id \\ Va \\ 0
\end{bmatrix}
$$

Name	Value [V]
V1	11.67246
V2	7.299291
V3	7.792599
V4	8.033134
V5	2.918905
V6	0.983174
V7	7.825301

Tabela 4: Node's voltage obtained through the application of the Node Method in Octave

With these results, we are able to compare both methods. Calculating $(V4-V3)/R1$ with the voltages of the node method we get I1. Repeating the same process for V2, V3 and R2 we obtain I2 and for V5, V6 and R6 we get I3. The calculations lead us to the following table:

Name	Value [A]
I1	0.000229300
I2	-0.000239986
I3	0.000947582
I4	-0.00101155

Tabela 5: Current results

Just like we expected, the results of both methods are coincident. Therefore, we can conclude that the two analyses are correct.

3   Simulation Analysis

3.1   Operating Point Analysis

With the purpose of solving the given circuit, we began by drawing a sketch of it, which can be seen in the picture below.
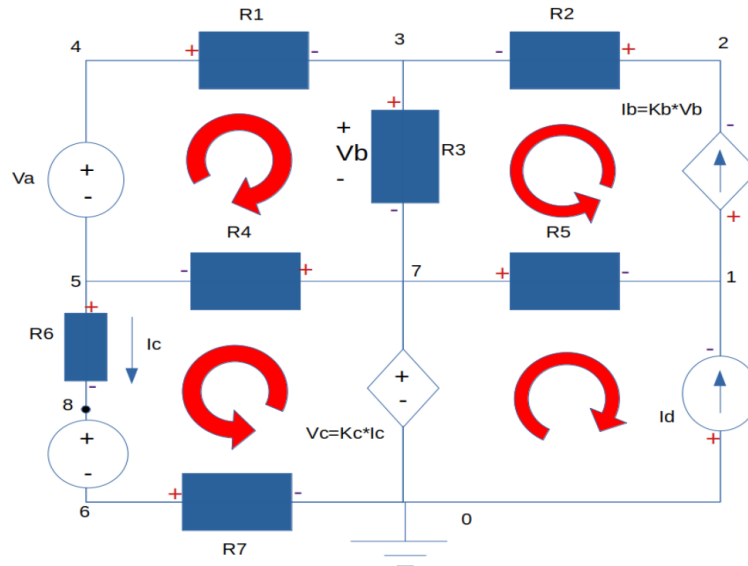

Figura 2: Analyzed Circuit in Ngspice

In order to understand our approach, it is important to note the following:

- We considered the node to which we connect resistor R7, Current Source Id and Dependent Voltage Source Vc to be the ground (node 0 in the picture).

-To compute the voltage drop through the dependent voltage source (Vc), we needed to obtain the current value through resistor R6. However, Ngspice doesn't allow us to input Resistor R6 current in the computation. Hence, in order to solve this problem, we introduced a voltage source with no voltage drop in series with the R6 resistor. This way, we managed to obtain the current through it, and therefore we were able to compute the dependent voltage source Vc.

Ngspice is a really useful tool, which allows us to compute the circuit and obtain all the necessary parameters. After carefully writing the script, in which we specify the nodes where we connect every single component and create the necessary dependencies for the dependent sources, we obtain the following table after typing the *ngspice script.name* command in a terminal window.

In this table, we can see the voltage at any given node, the current through each resistor and the voltage and current sources. (We use the Ngspice terminology: @-current value; Also, voltage value at node 0 (0V) is not shown)

Name	Value (A or V)
@gb[i]	-2.39986e-04
@id[current]	1.011550e-03
@r1[i]	2.293000e-04
@r2[i]	-2.39986e-04
@r3[i]	-1.0686e-05
@r4[i]	1.176882e-03
@r5[i]	-1.25154e-03
@r6[i]	9.475818e00
@r7[i]	9.475818e00
v(1)	1.167246e+01
v(2)	7.299291e+00
v(3)	7.792599e+00
v(4)	8.033134e+00
v(5)	2.918905e+00
v(6)	9.831754e-01
v(7)	7.825301e+00
v(8)	9.831754e-01

Tabela 6: Results from Ngspice

4   Results Analysis, Errors and Conclusion

After performing the computation, it's time to compare its' results to the ones obtained theoretically, through the application of the mesh and node methods with the help of Octave's math tool. In order to do so, we will calculate the error for every single value obtained.

Name	Relative Error (%)
V1	0
V2	0
V3	0
V4	0
V5	0
V6	0
V7	0
IA	0
IB	0
IC	2.11064e-05

Tabela 7: Relative Error between Octave's and Ngspice's values

As we can see the error values are minimal for every single component of the circuit. Even though we could have expected a bigger error margin due to the number of circuit components, that was not the case. The main reason for this perfect match between computation results and theoretical results is the circuit simplicity, since its components are only DC voltage sources, current sources and resistors. We believe that the main cause for errors was probably because of rounding issues, due to Octave's limited floating point representation.