Java Based Symbolic Circuit Solver For Electrical Engineering Curriculum

http://dx.doi.org/10.3991/ijoe.v8i4.2155

Ruba A. Amarin[1], Ehab Shoubaki[1] and Issa Batarseh[2]
[1] University of Central Florida, Orlando, USA
[2] Princess Sumaya University for Technology, Amman, Jordan

***Abstract*—The interactive technical electronic book, TechEBook, currently under development at the University of Central Florida (UCF), introduces a paradigm shift by replacing the traditional electrical engineering course with topic-driven modules that provide a useful tool for engineers and scientists. The TechEBook comprises the two worlds of classical circuit books and interactive operating platforms such as iPads, laptops and desktops. The TechEBook provides an interactive applets screen that holds many modules, each of which has a specific application in the self learning process.**

This paper describes one of the interactive techniques in the TechEBook known as Symbolic Circuit Solver (SymCirc). The SymCirc develops a versatile symbolic based linear circuit with a switches solver. The solver works by accepting a Netlist and the element that the user wants to find the voltage across or current on, as input parameters. Then it either produces the plot or the time domain expression of the output. Frequency domain plots or Symbolic Transfer Functions are also produced. The solver gets its input from a Web-based GUI circuit drawer developed at UCF. Typical simulation tools that electrical engineers encounter are numerical in nature, that is, when presented with an input circuit they iteratively solve the circuit across a set of small time steps. The result is represented as a data set of output versus time, which can be plotted for further inspection. Such results do not help users understand the ultimate nature of circuits as Linear Time Invariant systems with a finite dimensional basis in the solution space. SymCirc provides all simulation results as time domain expressions composed of the basic functions that exclusively include exponentials, sines, cosines and/or *t* raised to any power.

This paper explains the motivation behind SymCirc, the Graphical User Interface front end and how the solver actually works. The paper also presents some examples and results to better explain the concept.

***Index Terms*—Electrical Circuits, Interactive book, Linear Circuits , Symbolic Circuit Solver, Solver, Tool.**

I. INTRODUCTION

The interactive TechEBook consists of 16 chapters with a total of 75 sections representing typical content for the introductory circuit course at most universities in the world. Each section discusses a new theory and concept supported with examples and problems [1]. Different topics are presented in the discussion text that provides full understanding of the concept. At the end of each section, QuizMe Modules are provided to quiz the students' understanding of the section [2]. Also, TutorMe Modules are intended to help understand basic concepts in a step-by-step manner [3, 4]. Design Modules (DM) are intended to help students develop their ability in designing real life problems and to link the theories they study in books with real design challenges, while the Practical Relevance Modules (PRM) are set to enhance the students' capacity to think about real life problems and also to teach them how to relate the theories they have learned with practical applications. The SymCirc Symbolic Solver operates as a standalone web application, or as a utility integrated with the TechEBook, and allows users to get symbolic transfer functions and time domain expressions for any circuit with switches, which will enhance the user's capabilities to build a circuit, run a simulation, and obtain results in simple time domain functions.

II. MOTIVATION

At the University of Central Florida, efforts are underway to develop the full content of the TechEBook using the above attributes of each section. The SymCirc Symbolic Solver is one major component of the TechEBook currently being developed through partial funding from NSF. The main objective of including this module is to provide engineering students in the electrical circuit classes with a more robust understanding of the electrical network theory through building and designing real time examples.

The SymCirc Symbolic Solver provides all simulation results as Time domain expressions composed of the basic functions that exclusively include exponentials, sines, cosines and/or *t* raised to any power. One big motivation for taking the symbolic simulation route is the fact that for circuits with switching components, convergence problems are avoided, since elements are represented as symbols rather than numbers. Also subsequent simulation of the same circuit with different parameter values is much faster, due to the fact that the system of equations representing the circuits need not be solved more than once.

III. TRANSFER FUNCTION BASED SYMBOLIC SIMULATION

Considering any circuit as being a system with multiple inputs and outputs, and input being any dependent source and an output being the voltage or current of any circuit element, it is possible to obtain transfer functions generally. The stepping stone to symbolic simulation are the set of transfer functions between all inputs and nodes of interest. A way to begin is to first obtain what is called the "Indefinite Admittance Matrix", which is like a regular admittance matrix but with the ground assumed to be

outside the circuit, and each node is considered a port. The Indefinite admittance matrix is constructed by populating every row i and column j with admittance terms y_{ij} as follows,

$$y_{ij} = \begin{cases} Sum\ of\ admittances\ connected\ to\ node\ i & ,if\ i = j \\ Negative\ sum\ of\ admittances\ between\ i\ and\ j & ,if\ i \neq j \end{cases} \quad (1)$$

That takes care of all passive elements. Active elements represented by dependent sources are populated into the matrix through different disparaging methods. For the sake of elegance and ease of code implementation a unified approach was developed which populates all dependent sources by transforming them to Voltage dependent current sources. Those are populated as shown in Figure 1. The transformation of other dependent sources into a Voltage dependent source is outlined in Figure 2.

As can be seen the unified approach depends on the introduction of an infinitesimal resistance X_o that can be later made to approach zero.

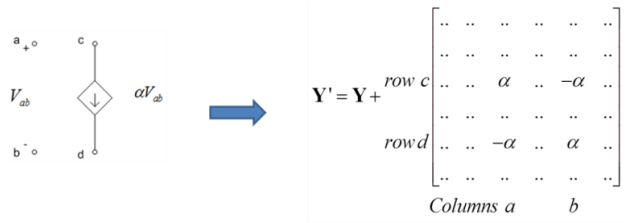


Figure 1. Populating voltage dependent current sources into the indefinite admittance matrix.

To obtain the transfer function from the indefinite admittance matrix, two constructs must be defined, the first order and second order cofactors defined respectively as follows,

$$Y_{ij} = (-1)^{i+j} \det \mathbf{Y_{ij}} \quad (2)$$

$$Y_{pq,rs} = \mathrm{sgn}(p-r)\,\mathrm{sgn}(q-s)(-1)^{p+q+r+s} \det \mathbf{Y_{pq,rs}} \quad (3)$$

Where Y_{ij} represent the admittance matrix with rows i and columns j removed, and $Y_{pq,rs}$ represents same matrix with rows p and r and columns q and s removed.

Taking into account definitions (2) and (3), the transfer function representing the voltage gain between arbitrary output at nodes p, q and input at nodes r, s is given by,

$$\frac{V_{pq}}{V_{rs}} = \frac{Y_{rp,sq}}{Y_{rr,ss}} \quad (4)$$

And the transfer impedance representing the voltage at nodes p, q versus an input current injected into node s and extracted from node r is given by,

$$\frac{V_{pq}}{I_{rs}} = \frac{Y_{rp,sq}}{Y_{uv}} \quad (5)$$

Calculating the determinants of matrices that contain infinitesimal elements X_o is simplified by substituting an

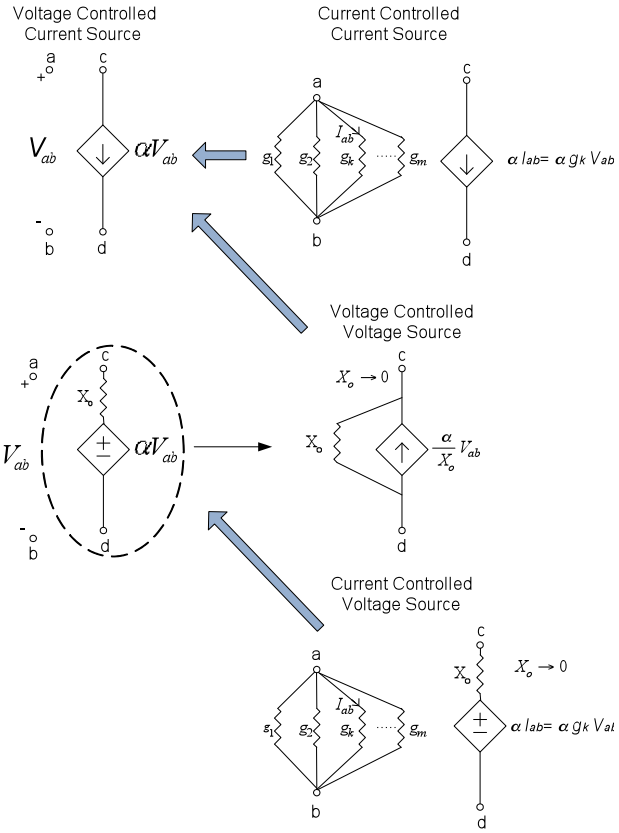


Figure 2. Outline of transforming other dependent sources into voltage dependent current source.

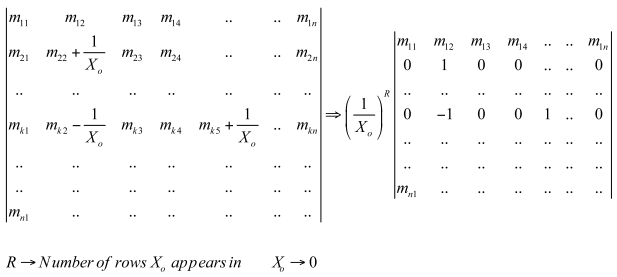


Figure 3. Pre-determinant evaluation of limit finding.

arbitrary matrix with another matrix as shown in Figure 3. This follows from the determinant property that states that multiplying any row by an element changes the determinant by the same proportion. Infinitesimal elements are then cancelled out between the numerators and denominators of (4) and (5).

The end product of applying the techniques to matrices with symbolic elements is symbolic transfer functions that can be used to initiate the simulation.

IV. JAVA IMPLEMENTATION

The object oriented language Java was chosen to implement the symbolic simulator, since the object paradigm allows for more modular and easily reusable code. This allows the concentration on each part of the design separately. Java also allows for easy web deployment for distributed use. The main results are coded into a versatile symbolic simulation engine that processes circuits inputted as netlists for simulation. The engine is a standalone component that can be interfaced with any user interface

front end and is responsible for all the symbolic manipulations necessary for finding time domain expressions for the voltages and currents of different circuits. The user needs to provide the engine with a netlist, whose source might be a graphical schematic or just a typed-in description of the circuit.

The symbolic solver engine can be decomposed into three main components, as follows,

- The Netlist Parser module: a module that converts a netlist generated by the GUI into a corresponding symbolic indefinite admittance matrix, while storing the values of the elements and inputs and other parameters like switch states for further use.
- Symbolic module: a module that implements the capability of constructing symbolic variables, aggregating them into expressions (multivariate polynomial or rational), and manipulating them in different ways (simplification, substitution, finding limits, etc.). Those symbolic objects can also be combined into Symbolic Matrix Objects on which symbolic determinant calculation, cofactoring and population operations can be performed. This module is a critical part of SymCirc with which the tenets of the Indefinite Admittance Matrix approach for circuit analysis are implemented.
- Time Domain Module: a module that picks up a rational function of *s* with numerical coefficients that represents the Laplace transform of the output and through a process of root extraction, residue evaluation and applying the inverse Laplace transform arrives at the time domain expressions of the signal at the output.
- Switching Module: The part of SymCirc that collects information about the switches within the circuit and their states with respect to time.

The SymCirc main components are described next.

A. Time and Frequency Domain Modules

The time and frequency domain modules of SymCirc consist of the Java implemented classes: Complex, Polynomial, Rational Function and LTI response. The purpose of this set of modules is to represent any transfer function in the *s* domain with numerical parameters, and have the capability of finding the inverse Laplace transform of those transfer functions to get the Time domain transform, and vice versa.

B. Symbolic Manipulation Modules

The previous section dealt with the modules relating to manipulating transfer functions with numerical coefficients and time expressions to obtain the response of a particular circuit. However, to obtain the transfer functions it is necessary to apply the General symbolic objects. The classes responsible for modeling those symbolic objects are “sop”, “rsop” and “SopMatrix” classes.

C. Graphical and User Interfaces

A graphical circuit drawer, developed at UCF, is added to SymCirc to speed up and ease the circuit construction process (Figure 4). This GUI communicates with the top level of SymCirc by passing the netlist or simulation results and receiving simulation expressions. The top level modules include the parser, switchset module and source module.

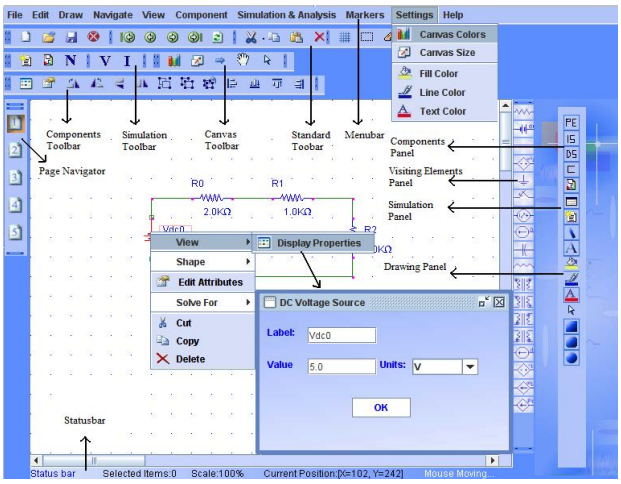


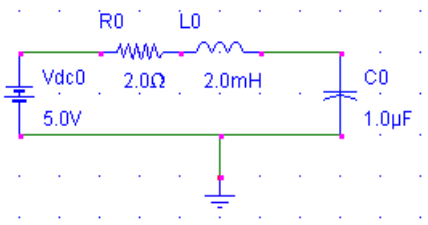
Figure 4. Graphical User Interface Front end for SymCirc.

Different components can be dragged and dropped, then connected in the drawing area (canvas) from the component panel that includes:

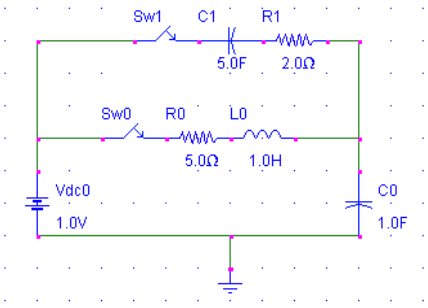
- PE (Passive Elements): From the PE bin, the user can access resistors, capacitors, inductors or transformers, then subsequently drag them into the grid.
- IS (Independent Sources): From the IS bin, the user can drag and drop voltage and current independent sources both AC or DC types.
- DS (Dependent Sources): The DS bin contains items that are normally used to model active circuits.
- C (Connectors): From the C bin, the user can access the wiring between element functions, or drop on the canvas a ground connection or a switch.

V. DEMONSTRATION

For demonstration purposes the circuits in Figure 5(a) and (b) were built into SymCirc, and various simulation results were obtained as shown in Figure 6 through Figure 10. The circuit in Figure 5(a) is a simple RLC circuit with a DC voltage source, while the circuit of Figure 5(b) is an RLC circuit with switches.



(a)



(b)

Figure 5. Demonstration circuits.

The transfer functions and time domain expressions for the circuit in Figure 5(a) is shown in Figure 6. The frequency response with respect to the input voltage for the circuit in Figure 5(a) is shown in Figure 7. The following waveforms were obtained for the circuit in Figure 5(b): The voltage waveform across C0 (Figure 8), the current waveform across C0 (Figure 9) and the power dissipated through R1 waveform (Figure 10).

VI. SUMMARY

There is no doubt that learning is the world's new language of communication and knowledge delivery, and the TechEBook has been used to translate streaked old circuit basic books to a new, well developed, and easy-to-use tool. This paper, addresses the TechEBook SymCirc Symbolic Solver in particular. It helps users understand the ultimate nature of circuits as Linear Time Invariant systems with a finite dimensional basis in the solution space by providing all the simulation results as time domain expressions composed of the basic functions. SymCirc will maximize the students learning experience, in their self-learning process (me-applying, me-designing).

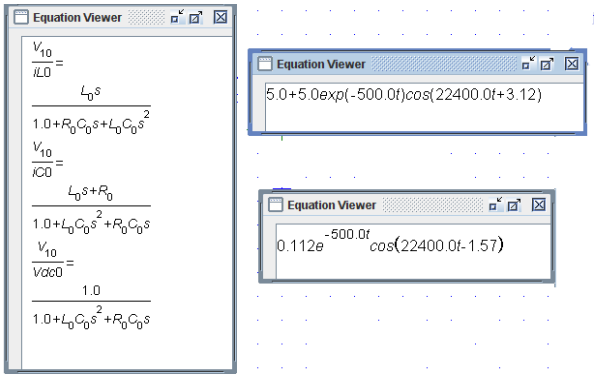


Figure 6. Transfer functions and time domain expressions for the circuit in Figure 5(a).

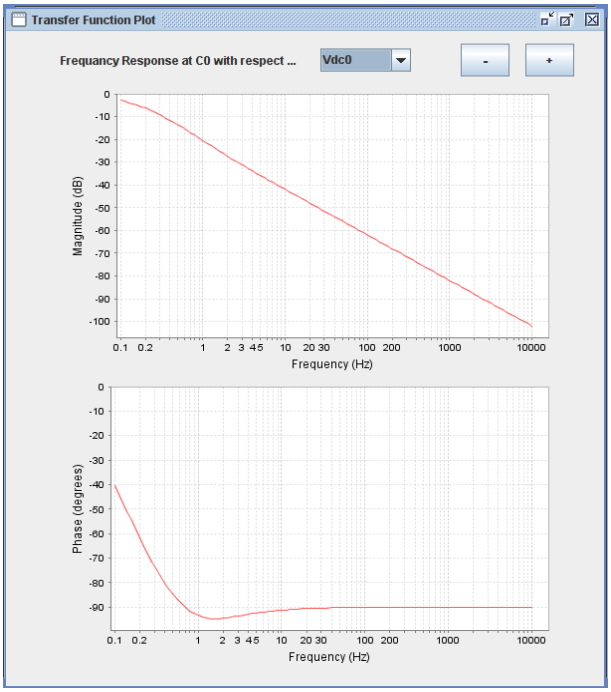


Figure 7. Frequency response with respect to the input voltage for the circuit in Figure 5(a).

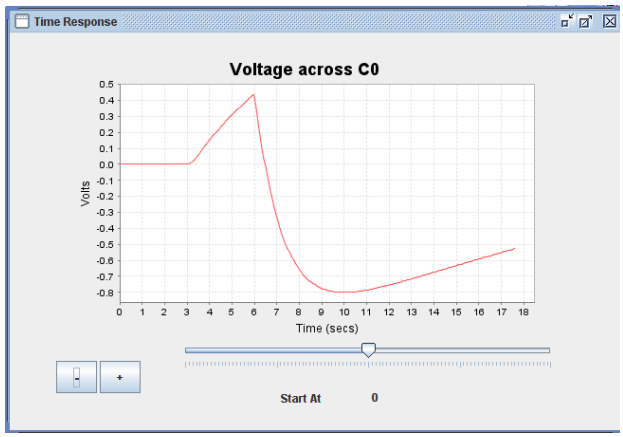


Figure 8. Voltage waveform across C0 for the circuit of Figure 5(b).

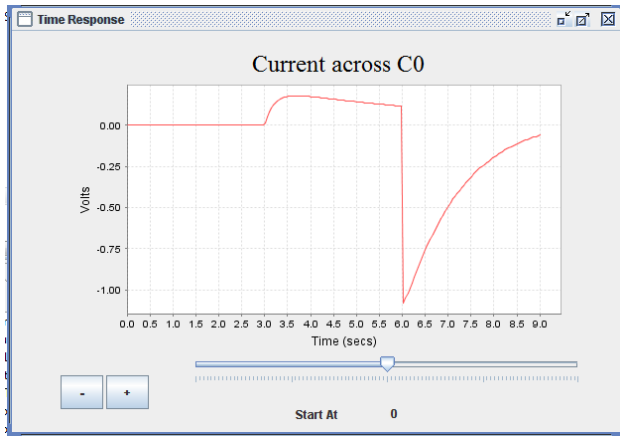


Figure 9. Current waveform across C0 for the circuit of Figure 5(b).

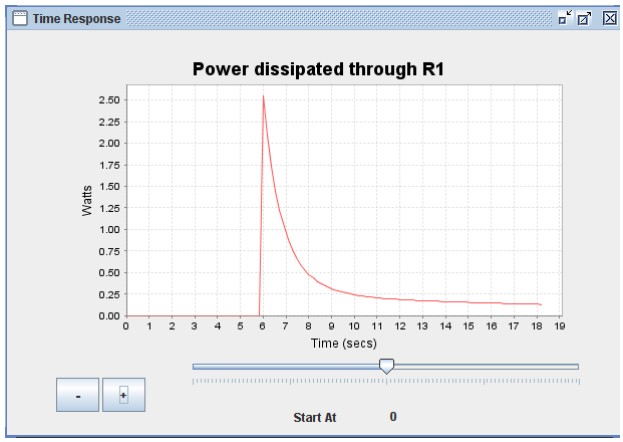


Figure 10. Power dissipated through R1 waveform for the circuit of Figure 5(b).

REFERENCES

[1] Issa Batarseh, Ghaith Haddad, Rawad Haddad and Rashad Oreifij, 'Interactive Electronic Book Operating Systems And Methods,' United States Patent 20080222552.

[2] Ruba A. Amarin, Feras Batarseh and Issa Batarseh, 'Adaptive Electronic Quizzing Method for Introductory Electrical Circuit Course', International Journal of Online Engineering, 2009.

[3] Ghaith Haddad, Gustavo Gamboa and Issa Batarseh, 'Interactive Electrical Circuit Tutoring Tool. eTutor,' 2008 ASEE Southeast Section Conference, Memphis, TN, April 2008.

[4] Ruba A. Amarin, Issa Batarseh, 'eTutor – An Interactive Module for Electrical Engineering Curriculum', ASEE Conference, Philadelphia, PA, 2011.

AUTHORS

R. A. Amarin is with the University of Central Florida, Orlando, FL 32816 USA (e-mail: ramarin@mail.ucf.edu).

E. Shoubaki is with the University of Central Florida, Orlando, FL 32816 USA.

I. Batarseh is with the Princess Sumaya University for Technology, Amman, Jordan (e-mail: issa.batarseh@psut.edu.jo).

This work is partially funded by NSF under CCLI Award Number: 0837364. It is a modified and extended version of a paper presented at the International Conference EDUCON2012, held April 2012, at University Mohammed V Souissi, Marrakesh, Morocco. Manuscript received 15 June 2012. Published as resubmitted by the author 23 October 2012.