ELEC40006: 1st Year Electronics Design Project 2020

Specification May 2020

Introduction

The purpose of this module is to bring together the knowledge and skills you have gained throughout this academic year by applying them to a practical problem.

You will also acquire new skills that will help you complete the project successfully.

The project is conducted in groups of three.

Timeline

Deadline: 14th June

Deliverables

1. **Report (50%):** The report is a formal documentation of all the technical and non-technical work you have done on the project. By this time all your design decisions will have been made and you will be able to document the performance of various aspects of your system / simulation. You should also have a clear plan for any work outstanding before you can complete the demonstration. One team member should act as overall editor to ensure that the report is consistent.

2. **Video demonstration (50%):** The team will record a video explaining their final design, alongside a demonstration.

3. Code and source files for plagiarism checking

Additional information for all three deliverables will be given later.

Circuit Simulator

1:Parse Netlist File

The netlist should be described in a file using a reduced SPICE format, which will be provided. You will need to read the file and store the circuit in a suitable data structure.

2: Setting up simulation

A transient simulation takes place by calculating the node voltages at each successive instant in time. The temporal resolution is known as the timestep and the duration is the stop time. A simulation with a stop time of 100ms and a timestep of 1µs would need to calculate 100,000 time instances.

Some components have a value that changes over time with a predefined function. For example, a sinusoidal voltage source has a time-varying voltage:

$$\mathrm{v}_{SIN}(t) = Asin(\omega t)$$

Some components have a value that depends on the previous history of the circuit. For example, a capacitor has a voltage proportional to the integral of the current through the capacitor:

$$\mathrm{v}_{CAP}(t) = \int \frac{i(t)}{\mathrm{C}}\, dt$$

3: Construct and solving conductance matrix

You have seen how nodal analysis can be performed by writing an equation for each node that satisfies Kirchhoff's current law, then solving these equations simultaneously to find the unknown node voltages. Such systems of linear equations can be solved systematically by writing them in matrix form and solving for the vector of unknowns.

$$\begin{bmatrix} G_{11} & -G_{12} & \cdots & -G_{1n} \\ -G_{21} & G_{22} & \cdots & -G_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -G_{n1} & -G_{n2} & \cdots & G_{nn} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix}$$

Equation for finding a vector of unknown node voltages from a conductance matrix

Here, G_{12} is the conductance directly connecting nodes 1 and 2, G_{11} is the total conductance connected to node 1, i_1 is the total current from current sources entering the node 1, and v_1 is the unknown voltage of node 1. The reference node is not included in the conductance matrix and instead its voltage is defined as zero.

To construct the conductance matrix, inductors are treated as current sources since their current cannot change instantaneously and it is effectively fixed during each simulation iteration. Similarly, capacitors are treated as voltage sources (discussed later) since their voltage cannot change instantaneously.

The solution is found by calculating the inverse of conductance matrix. This is complex but common operation in computing, so it makes sense to use a library, but you must justify your selection of a library and how you have used it in your report.

4: Process Voltage Sources

Voltage sources must be treated specially since the conductance of an ideal voltage source is infinite and so it cannot appear in the conductance matrix. If one terminal of the voltage source is connected to the reference node then the node connected to the other terminal can be expressed simply as something like $v1 = vsrc$, e.g. for a source v_{src} volts connected to node 1. The conductances connected to that node are ignored and in matrix form that would look like this:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ -G_{21} & G_{22} & \cdots & -G_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -G_{n1} & -G_{n2} & \cdots & G_{nn} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} v_{src} \\ i_2 \\ \vdots \\ i_n \end{bmatrix}$$

Circuit analysis equation containing a voltage source between node 1 and reference

If the voltage source is connected between two non-reference nodes then two rows are needed in the matrix, since there are two unknown voltages. The first row represents the voltage source, but it now shows the difference between two nodes rather than an absolute, e.g. $v_1 - v_2 = v_{src}$

The second row is KCL for the supernode enveloping the voltage source. It is equal to the sum of the KCL equations for both the nodes in the supernode, which cancels out the conductance of the voltage source. For example, here a voltage source between nodes 1 and 2 creates a supernode S:

$$\begin{bmatrix} 1 & -1 & \cdots & 0 \\ 0 & G_{SS} & \cdots & -G_{Sn} \\ \vdots & \vdots & \ddots & \vdots \\ -G_{n1} & -G_{n2} & \cdots & G_{nn} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} v_{src} \\ i_S \\ \vdots \\ i_n \end{bmatrix}$$

Circuit analysis equation containing a voltage source between node 1 and node 2

The principle is extended if more than one voltage source is connected to a node. Each voltage source creates one matrix row that defines its potential difference, and if the supernode enveloping all the connected voltage sources does not include the reference node, then there is also a row that describes KCL for the supernode.

5: Write output

The results of the simulation are reported by creating a file describing the voltage at each node and the current through each component at every instant in the simulation. Write the output in CSV format where the columns are the nodes and components and the rows are the instances in time. Use the MATLAB script provided on blackboard to plot the results.

6: Adding support for non-linear components [Advanced]

When the circuit contains non-linear components, like diodes, there is no analytic solution. Instead a numerical solution is calculated using the iterative Newton Raphson method.

The component is converted to a linear approximation (a Thevenin equivalent) by guessing the potential difference and differentiating the I-V characteristic to obtain its gradient. Then, nodal analysis is performed as usual. The results of the nodal analysis are used to find a new potential difference, and from that a new linear approximation. The process is repeated until the node voltages stop changing between each iteration and converge on the solution.

7: Evaluation

Your solution should be evaluated against the following criteria:

1. **Accuracy**: Compare the outputs to pen and paper solutions.
    a. For simple circuits you should be able to calculate an exact solution as a reference. You can also compare with LTspice.
2. **Efficiency**:
    a. Find how long the simulation takes and, by estimating the power consumption of your computer, the amount of energy needed.
    b. How does it scale with the number of nodes in the circuit?
    c. Are there any implementation choices that affect the efficiency?