وَمَا أُوتِيتُم مِنَ الْعِلْمِ إِلَّا قَلِيلًا

Analog/Mixed-Signal Simulation and Modeling
Lab 02
Create Your Own Circuit Simulator!

Objectives

1. Parse a netlist description of a linear circuit with arbitrary number of nodes. The circuit may contain resistors, voltage sources, and current sources.
2. Build the modified nodal analysis (MNA) equation system of the circuit using element stamps.
3. Solve the circuit (linear DC and AC analysis) and print the following results for the user:
   a. Symbolic matrix
   b. Numerical matrix (substitute with the values given in the netlist)
   c. Symbolic solution of every unknown
   d. Numerical solution of every unknown (substitute with the values given in the netlist)

Instructions

1. Use Octave or Matlab to write your program (use the Symbolic Toolbox and define symbolic variables using the 'syms' command).
2. Use LTSpice for design entry and simulation. Use this link: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html
3. It is recommended to use Notepad++ to write your netlist (use SPICE highlighting mode: Language -> S -> Spice). Do NOT use a schematic entry GUI.
4. If necessary, make any reasonable assumptions.
5. Check LTSpice help if needed.
6. Submit your solution on Canvas as a single zip file that contains all the deliverables.

Part 1

Index	Deliverable
1.	Complete the missing lines in "Solve_Circuit.m". This is a program written as a single function. The function should accept the netlist as an input (a path to a text file). The function should return **the symbolic and numerical results** as explained in the objectives section. You may assume the following for the netlist: ● Node 0 is the ground node. ● Other nodes are named as positive integers.
2.	Complete the missing lines in "Solve_Circuit_Examples.m". This is a script that calls your function and prints the output clearly in the command window for the two given netlist files ("circuit_1.cir" and "circuit_2.cir").
3.	Simulate the netlists and compare LTspice output with your simulator output in a table. Comment.

Part 2

Index	Deliverable
1.	Create “Solve_AC_Circuit.m” to add support for inductors, capacitors, and AC analysis.
2.	Create a script ““Solve_AC_Circuit_Examples.m” to test your code with three netlists of an RLC LPF circuit to cover three cases: overdamped, critical damped, and underdamped. Your simulator should perform AC analysis and plot magnitude and phase vs frequency.
3.	Report your simulator results.
4.	Simulate the netlists using LTspice and compare with the results of your simulator. Comment.

Part 3

Index	Deliverable
1.	Create “Solve_AMP_Circuit.m” to add support for VCCS and VCVS.
2.	Create a script ““Solve_AMP_Circuit_Examples.m” to test your code with the op-amp circuit you created in Lab 01 connected in unity-gain feedback configuration. Your simulator should perform AC analysis and plot magnitude and phase vs frequency.
3.	Report your simulator results.
4.	Simulate the netlist using LTspice and compare with the results of your simulator. Comment.

FAQ

Q: Octave symbolic library is not loaded.
A: Run the following command in the command window: “pkg load symbolic”, then try to run the file again.

Q: Incrementing vs overwriting inside for loops: “foo = foo +1” vs “foo = +1”.
A: For every loop iteration we can add one to the old value like: “foo = foo + 1”, so we are incrementing foo inside the for loop.
On the other hand, “foo = +1” only overwrites the previous value with “+1”.
Appending is needed while populating the G matrix for example, to preserve the old passive elements stamps.

Thanks to all who contributed to these labs. If you find any errors or have suggestions concerning these labs, please contact Hesham.omran@eng.asu.edu.eg.