Analysis Module Documentation

Adam Rehman, Brandon Cann, Xin Wang [1]

May 16, 2020

[1]document compiled by Xin Wang

Contents

1 Introduction

This document will research the theory of analysing a circuit and design the *Analyser* module. The *Analyser* module will create the matrices required to perform the nodal analysis and, using the matrices, perform a quick scan to ensure the circuit described is a practical circuit. As stated previous in the Netlist documentation, the netlist input format may be correct but the circuit is unrealistic e.g. a resistor is only connected on one end.

The initial stage is will be to implement a version of the software that supports basic components like resistors and voltage source. Support for capacitors and inductors, and non-linear components will be added at a later stage.

This document is primarily used for planning the most effective way to approach the problem as well as allowing fellow teammates to engage productively on a equal knowledge footing.

2 Modified Nodal Analysis

2.1 Node Voltage Method

Building from first principals, node voltage method is the foundational concept required to analyse a circuit but it is not easily converted into an algorithm. Modified Nodal Analysis seeks to establish an universal algorithm for solving a circuit.

- Establish a reference node (Ground)
- Name remaining nodes
- Name current through each voltage source
- Apply KCL at each node. **Currents out of node is taken to be positive.**
- Write equation and solve for unknowns

$$
\begin{aligned}
\frac{v_a}{R_1} - i_{v1} &= 0 \\
i_{v1} + \frac{v_b}{R_3} + \frac{v_b - v_c}{R_2} &= 0 \\
i_{v2} + \frac{v_c - v_b}{R_2} &= 0 \\
v_b - v_a &= V_1 \\
v_c &= V_2
\end{aligned}
\qquad
\begin{bmatrix}
\frac{1}{R_1} & 0 & 0 & -1 & 0 \\
0 & \frac{1}{R_2} + \frac{1}{R_3} & -\frac{1}{R_2} & 1 & 0 \\
0 & -\frac{1}{R_2} & \frac{1}{R_2} & 0 & 1 \\
-1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix} v_a \\ v_b \\ v_c \\ i_{v1} \\ i_{v2} \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ V1 \\ V2 \end{bmatrix}
$$

Figure 1: Process of conversion of equations into a matrix [1]

2.2 Characteristics of the MNA matrix

All circuits when presented in matrix for, will always result in the form:

$$Ax = b$$

In the given example matrix[1]:

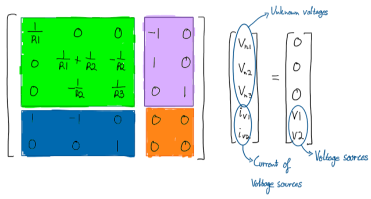

Figure 2: Characteristics of $Ax = b$

The following characteristics will always hold:

- Matrix A
  - The highlighted part is size $[N \times N]$ and contains values of passive elements
  - The main diagonal is positive values only. Non-diagonal values are either 0 or negative.
  - If an element is grounded, it will appear along diagonal.
- Matrix x: Matrix of unknown quantities:
  - General dimension: $[(N + M) \times 1]$
  - Top N elements are simply node voltages
  - Bottom M elements are the currents related to voltage sources[2]
- Matrix b: Matrix containing known quantities:
  - General dimension: $[(N + M) \times 1]$
  - Top N elements are either 0 or sum of independent current sources
  - Bottom M elements are independent voltage sources

Solution is found by:

$$x = A^{-1}b$$

[1] Red highlight is $[N \times N]$ and matrix is $[(N + M) \times (N + M)]$ where N is the nodes in the circuit and M is the number of independent sources.

[2] Dependent current and voltage sources have not been considered yet

2.3 Notations

The naming notation is the following:[1].

- Ground is labelled **Node 0**
- Each node is given a label starting from 1 to N
- Node voltage name: v_N
- Current through V sources: $i\_VoltageSourceName$
- Independent voltage sources: $vname$
- Independent voltage sources: $iname$

3 General Algorithm design for MNA

Matrices need that need to be generated:

- A
- x
- b

3.1 A matrix

The A matrix has dimension $[(M+N) \times (M+N)]$ and is combined from four smaller matrices: A_a, A_b, A_c and A_d [1].

The respective matrices are defined as:

- A_a: $n \times n$ matrix - Passive element connections
- A_b: $n \times m$ matrix - Voltage source connections
- A_c: $m \times n$ matrix - Similar to matrix A_b
- A_d: $m \times m$ matrix - 0 if independent sources are considered

3.1.1 A_a matrix

- Size $N \times N$
- Each element in diagonal matrix is the conductance of each element connected to respective node
- Elements not on diagonal are negative conductances of the element connected to respective node.
- Non-grounded element will have one entry.
- Non-grounded element will have four entries.

For example:

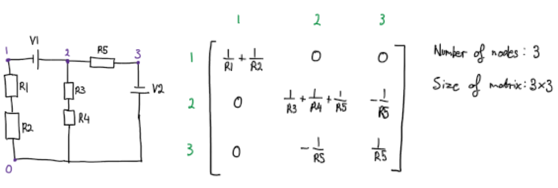

Figure 3: The relationship between diagonal and non-diagonal elements

3.1.2 A_b matrix

- Size $N \times M$
- Only 1, 0 and -1 entries
- Direction of voltage source matter. Negative terminal is -1 and positive terminal is 1.
- A grounded voltage source will have one entry.
- Non-grounded voltage source will have two entries.

The positive terminal of V1 in the above circuit is connected to node 2 as indicated in matrix element $[1, 2]$ where 1 is Vi and 2 is the node the terminal is connected to.

For example:

$$\begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & -1 \end{bmatrix} \end{array}$$

Figure 4: Matrix showing the direction of voltage sources

3.1.3 A_c matrix

- Size $M \times N$
- **Usually the transpose of A_b**
  - Not the case if dependent sources are involved.

3.1.4 A_d matrix

- Size $M \times M$
- Usually composed of 0
  - Not the case if dependent sources are involved.

3.2 x matrix

The x matrix has dimension $[N \times M]$ and is combined from two smaller matrices: x_a and x_b.

The respective matrices are defined as:

- x_a: $N \times 1$ - Unknown node voltages
- x_b: $M \times 1$ - Unknown currents through voltage sources

3.2.1 x_a matrix

The naming follows the notation set out earlier.

There is no entry for Ground (Node 0).

$$\begin{bmatrix} V\_1 \\ V\_2 \\ \vdots \\ V\_n \end{bmatrix}$$

Figure 5: General notation of the x_a matrix

3.2.2 x_b matrix

Corresponds to the number of voltage sources. Example given in Figure 2.

3.3 b matrix

This matrix has dimension $[(N \times M)) \times 1]$ and contains the independent current and voltage sources.

The matrix is composed of two matrices: b_a and b_b:

- b_a: $N \times 1$ Either 0 or the sum of independent current sources.
- b_b: $M \times 1$ Values of independent voltage sources.

4 Support for C and L elements

Adding support for reactive elements, the code in requires modifications and additions. Before that is possible, further insight is required as to how C and L can be represented in MNA. This will allow us to identify what functions need to be implemented and how to best incorporate it into the existing codebase.

4.1 Theory behind capacitor and inductor

Capacitors can be seen as a current source defined by the equation:

$$i_{(t)} = C\frac{dv}{dt}$$

Inductors can be seen as a voltage source defined by the equation:

$$v_{(t)} = L\frac{di}{dt}$$

4.2 Changes required

- New parameter ω needs to be defined
  - Default 0 if no frequency value entered
    * Capacitor appears as a open circuit
    * Inductor appears as a short circuit
- Data structures need to be changed to support complex numbers

5 Dependent sources

5.1 Voltage Controlled Current Source (VCVS)

6 The use of Eigen library

Eigen is a linear algebra library used in the C++ environment. For our project, we chose to implement Eigen functions to handle the matrix related operations such as finding inverse matrix and matrix initialisation.

The design decision is reached when we considered the disadvantages of writing our own matrix libraries, mainly:

- **Poor performance** unless significant time is spent optimising which will come at the cost of neglecting other aspects of the project.
- **Buggy code** unless significant time is spent considering edge cases, going back to first point.

By using an established library like Eigen, we have several advantages:

- **Up to date and well tested code**
- **Dynamic matrices and optimised structures** this will allow us to easily accomodate optimisations in other aspects of the software package.
- **Very optimised code**

7 Implementation

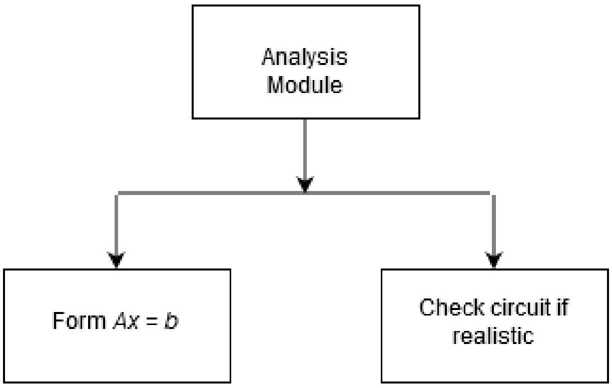

Figure 6: *Analysis module* breakdown

7.1 Analysis

```
Analysis
{
    Get N and M
    Initialise matrices
    Form Ax = b
    Check circuit semantics
    Find inverse of A
    Solve for x
}
```

7.1.1 FormMatrix

```
Analysis(vector<CirElement>)
{
    For loop over entire circuit vector
    {
        Identify component type

        Case: R,C or L
        {
            Check if grounded
            Insert in A matrix [Diagonal and non-diagonal]
        }

        Case: I
        {
            Check if grounded
            Insert in b matrix
        }

        Case: V
        {
```

```
            Check if grounded
            Add 1 or −1 in A matrix
            Insert in b matrix
            Insert in x matrix [Unknown values are 0]
        }
    }
    return matrix x;
}
```

7.1.2 SemanticCheck

```
SemanticCheck(vector<CirElement>)
{
    Check if node are at least paired [Ignore node 0]
}
```

8 Version history

8.1 Version 1.0

9 Miscellaneous

References

[1] Professor Erik Cheever. *Analysis of Circuits*. 2005 - 2019. DOI: `https://lpsa.swarthmore.edu/Systems/Electrical/mna/MNA1.html`.