A SIMPLE FORMULATION METHOD FOR REDUCED NUMBER OF
CIRCUIT EQUATIONS IN MNA

S. Natarajan, Senior Member IEEE
and
A. Balendra

Department of Electrical Engineering
Tennessee Technological University
Cookeville, TN 38505

ABSTRACT

A memory efficient procedure is necessary to formulate circuit equations in microcomputers for large networks to reduce the memory space and run-time. This paper describes a simple but effective modification to a well known circuit formulation method, Modified Nodal Analysis (MNA), to reduce the number of equations while formulating the circuit equations. For this purpose, a set of simple rules are presented in this paper. This new method is compared with the original MNA for simplicity and memory space. The new method is found to be superior to MNA in memory requirement and run time. The complete method of implementing the rules with examples of its use is presented.

I. INTRODUCTION

The computer formulation of circuit equations for general class of networks requires the use of modified nodal analysis (MNA) [1]. This method is more straight forward and easy to implement. In this method of formulation by one-graph technique, current variables of elements, such as op. amps., voltage sources (both dependent and independent) and inductors, are used in addition to the node voltages. The resulting circuit equations are in the form of

\[(G + sC)X = W\]  

(1)

where \(G\) and \(C\) are two real square matrices, \(X\) is a column vector containing the node voltages and additional currents, \(W\) is the source vector containing the strengths of independent sources and initial conditions. The formulation of (1) can be done easily using the rubber stamps available in [2]. To formulate (1), one starts with two matrices \(G\) and \(C\) of sizes \(n \times n\) and a column vector, \(W\) of size \(n\), where \(n\) is the total number of non reference nodes. The contributions of the elements, which have an admittance description, are added in either \(G\) or \(C\) matrix depending upon the element. Whenever there is a need for entering the characteristic equation of an element, that does not have an admittance description, its current variable(s) is(are) added to the existing variables and its branch characteristic(s) is(are) added as additional equation(s). This necessitates the increase in the dimensions of \(G\), \(C\), and \(W\) matrices.

The problem with the use of MNA is that, it leads to large number of equations even for small size networks. This is especially true with the active networks. For example, in a Tow-Thomas biquad shown in Fig. 7 [3], there are only 7 non reference nodes. The use of MNA for this network results in 11 equations because 4 more additional current variables are added. The memory requirement is proportional to \(n^2\) and run-time is proportional to \(n^2\), where \(n\) is the total number of equations. This means that, when we add the 4 additional current variables, the size of \(G\) and \(C\) matrices increase approximately to 2.5 times. Therefore the memory requirement increases by the same factor and the run-time will also increase.

One can employ the two-graph method [2] which uses separate voltage and current graphs and formulate the circuit equations. The two-graph method eliminates some of the redundant current variables and so the size of the matrices will be less. However this method requires renumbering of the nodes and several "book-keeping" procedures to keep track of node numbers in the voltage and current graphs. The "book-keeping" procedures in the two-graph method require a lot of programming effort. These facts bring out the difficulties involved in formulating circuit equations using the two-graph method.

There are some methods, such as in [4], [5], in which the circuit equations are reduced after formulating the matrices using MNA. In this way the memory requirement, when initial equations are formulated, can be high.

In this paper some simple rules are presented for not including some of the current variables while formulating the circuit equations. In this way the number of equations is, except in some cases where current variables have to be included, such as in CCT, CVT and inductors, equal to the total number of non reference nodes. In many practical circuits, one does usually find CCT and CVT elements. Thus the sizes of the matrices \(G\), \(C\), and \(W\) are not increased to add currents except the inductor currents. Therefore the memory requirement for the matrices will be lower. This is achieved by implementing some special rules while formulating the circuit equations. In micro-computers, where memory requirement is critical [5], this reduction in memory requirement is very important. So this method enables one to simulate larger size networks, particularly active networks, in micro-computers. This new method does not use the renumbering of nodes and thus it avoids a lot of "book-keeping" procedures.
keeping" procedures. Therefore it requires less additional programming effort and this new method is found to be much simpler than the two-graph method.

II. DEVELOPMENT OF THE RULES

The circuit equations, formulated by using MNA, can be expressed as

\[ TX = W \]  

(2)

where \( T = G + sC \) and \( G \) and \( C \) have the dimension of \( n \). \( n \) is equal to the sum of the number of non reference nodes and the number of additional current variables that are added to describe the branch characteristics that do not have admittance description. In one-graph method, the elements are split into two groups. The first group consists of elements which have admittance description. The second group does not have such an admittance description. All conductance values and frequency independent values are entered in the \( G \) matrix, whereas the capacitor and inductor values and those values that are related to frequency variable, \( s \) are entered in the \( C \) matrix. All the inductors must be entered in the impedance form and its current must be included. Similarly the controlling currents of active devices have to be included and one cannot avoid them. To formulate (2), the rubber stamps presented in [2] can be used for adding the contributions of the various circuit elements.

Assume that the contributions of all elements, except voltage sources (both independent and dependent) and op.amps., have been entered in the \( G \), \( C \), and \( W \) matrices. Then we need to see how one can enter the contributions of the voltage sources and op.amps. without including their currents as is normally done in MNA. The main aim of this paper is to provide a set of simple rules for this purpose.

Now consider a floating voltage source, shown in Fig. 1, connected between the nodes \( J \) and \( K \). This voltage source does not have an admittance description. To add the contribution of this element in the conventional one-graph method, one adds the current, \( I_{n+1} \) as an additional variable and the characteristic equation of the same, namely \( V_J - V_K = E \), is added as an additional \( (n+1) \)th equation. This is nothing but the constraint equation for the voltage source. Thus the rubber stamp, corresponding to the contribution of the voltage source, is given below.

\[
\begin{pmatrix}
J & V_J & V_K & I_{n+1} & \text{Source Vector} \\
K & 1 & -1 & 0 & \text{Source Vector} \\
n & 1 & -1 & 0 & \text{Source Vector} \\
n+1 & 1 & -1 & 0 & \text{Source Vector}
\end{pmatrix}
\]

The increase in the size of the \( T \)-matrix is indicated by \((n+1)\). Assume that we do not need this current, \( I_{n+1} \) as an output. This is usual case in most network analysis procedures. In such a case we do not need this additional variable. However, in MNA, when we apply KCL at nodes \( J \) and \( K \), this additional current variable is to be included. This is reason for the \(+1\) and \(-1\) in the column corresponding to \( I_{n+1} \). The addition of this current variable could have been avoided, if we had applied KCL to the super node as indicated in Fig. 1. The KCL equation corresponding to the super node is nothing but the addition of the KCL equations applied at the nodes \( J \) and \( K \), respectively. This is equivalent to the addition of the rows \( J \) and \( K \) in both \( T \) and \( W \) matrices. The linear independence of the equations as well as the solution of the network will not be affected by replacing any row with a sum of any two rows. Therefore, assume that row \( K \) is replaced with the sum of the rows \( J \) and \( K \). Then the equivalent set of rubber stamp is

\[
\begin{pmatrix}
J & V_J & V_K & I_{n+1} & \text{Source Vector} \\
J+K & 0 & & 0 & \text{Source Vector} \\
n & 1 & -1 & 0 & \text{Source Vector} \\
n+1 & 1 & -1 & 0 & \text{Source Vector}
\end{pmatrix}
\]

Now note that the contribution of \( I_{n+1} \) is included only in the \( J \)th equation and the \( J \)th equation can only be used to solve for the current, \( I_{n+1} \). If we are not interested in this current (as per our earlier assumption), we can simply delete the \( J \)th row and \((n+1)\) column without affecting the solution for the remaining circuit variables. Instead of adding a \((n+1)\)th row and deleting the \( J \)th row, one can as well replace the \( J \)th row with the \((n+1)\)th row. We never want to change the \( J \)th row again. Therefore, one has to store the numbers \( J \) and \( K \) in a matrix, \( R \). For adding the contributions of other voltage sources or op.amps., connected to these nodes, \( J \) and \( K \), we need the information about these nodes later. This point will be elaborated later in this section. The above method can be summarized as follows.

a) Set \( T_{KI} = T_{KJ} + T_{J1} \), \( I = 1, 2, \ldots, n \).

b) Set \( T_{J1} = 0 \), \( I \neq J, K \).

c) Set \( T_{JK} = 1 \), \( T_{J1} = -1 \) and \( W_J = E \).

d) Store \( J \) and \( K \) in the matrix, \( R \).

The rubber stamp of a grounded voltage source, shown in Fig. 2, in the one-graph formulation will be

\[
\begin{pmatrix}
J & V_J & I_{n+1} & \text{Source Vector} \\
K & 1 & & \text{Source Vector} \\
n & 1 & 0 & \text{Source Vector} \\
n+1 & 1 & 0 & \text{Source Vector}
\end{pmatrix}
\]

In this case the equation of \( J \)th row can be deleted if the current, \( I_{n+1} \) is of no interest. Instead of deleting of \( J \)th row and adding \((n+1)\)th row, the \( J \)th row can be replaced by the \((n+1)\)th row. Now \( J \)th row should have no entries.
in the future. So the number J should be stored in the matrix, R.

The rules for grounded voltage can be summarized as follows.
- Set \( T_J = 0 \) for \( I = 1, 2, \ldots, n, I \neq J \).
- Set \( T_J = 1 \).
- Set \( W_J = E \).
- Store \( J, 0 \) in the R-matrix.

For a VVT, shown in Fig. 3, if the current, \( I_{n+1} \), is of no interest, then the same procedure as given for a floating voltage source can be used and the variable, \( I_{n+1} \), need not be included. However the constraint equation will be \( kV_K - kV_J + V_L - V_{JJ} = 0 \). In the case of CVT also, one current variable need not be included in this way. Unfortunately, in this case, the controlling current of a CVT has to be included as an additional variable.

The contributions of an op.amp. can be treated in the same way, its output resistance, \( R_o \), is equal to zero and the output current of the op.amp. is of no interest. Consider the op.amp. shown in Fig. 4, with a finite gain A. The rubber stamp of this is given below.

\[
\begin{pmatrix}
J & V_J & V_K & V_L & I_{n+1} \\
K & & & & \\
L & & & 1 & \\
n & & & &
n+1 & 1 & -1 & -1/A
\end{pmatrix}
\]

The L\textsuperscript{th} row can be replaced by the (n+1)\textsuperscript{th} row. The current variable, \( I_{n+1} \), need not be added as a new variable.

In this method also, there is a need for a simple "book-keeping" procedure. To see this, assume that we first added the contribution of a voltage source, \( E_1 \) connected between two terminals, \( J \) and \( K \). Without loss of generality, also assume that we replaced the J\textsuperscript{th} row with the constraint equation of \( E_1 \). It is entirely possible that we may have to add the contribution of another voltage source (or an op. amp.) either connected between the J\textsuperscript{th} node and an L\textsuperscript{th} node or between the K\textsuperscript{th} node and an L\textsuperscript{th} node. These two possibilities are illustrated in Figs. 5 and 6 respectively where we have used two independent voltage sources for the purposes of discussion. First consider the situation in Fig. 5. Since the J\textsuperscript{th} row is not to be altered, one can show that we need to replace the L\textsuperscript{th} row with the constraint equation of the new voltage source, \( E_2 \). However, if the new voltage source is added between the K\textsuperscript{th} and L\textsuperscript{th} node as illustrated in Fig. 6, then we replace the L\textsuperscript{th} with the sum of the rows \( K \) and \( L \) and then replace the K\textsuperscript{th} row with the constraint equation of the voltage source, \( E_2 \). The later one is similar to the normal addition of any voltage source. In any case, a simple "book-keeping" procedure for checking those nodes to which the voltage sources and op. amps. are connected (and not other nodes) before adding the contribution of a voltage source (or an amp.) is necessary. This is the reason for entering the contributions of all other elements first. The storage of these nodes is for checking purposes only and we do not require renumbering the nodes as in the two-graph method. Therefore, this "book-keeping" procedure is simple to implement with less additional programming effort.

The above simple rules, developed in this section, can be used for not including most currents while formulating the circuit equations. We provide two examples in the next section to show the effectiveness of the modifications suggested above for adding the contributions of voltage sources and op. amps.

### III. EXAMPLES AND COMPARISON

A computer program was written to formulate the circuit equations using MNA. In this program, the rules described in the previous section were implemented as a part of the program. Several example circuits were tested using this program to find the effectiveness of the modifications. Even though the effectiveness will be considerable for large size networks, here we provide only two examples of small size networks to save space. Furthermore, for the same reason, we do not provide the matrices obtained using the original MNA in both examples.

Consider the Tow-Thomas biquad shown in Fig. 7. This circuit has 7 nonreference nodes. Assume that all the op.amps. are ideal for simplicity. The number of equations obtained using the conventional MNA would have been 11. The G, C, and W matrices, using the new modified method, are

\[
G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 7 & -2 & 0 & 0 & 0 & -4 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -10 & 20 & -10 & 0 & 0 \\
0 & 0 & 0 & -5 & 5 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & -2 \\
0 & 0 & 0 & 0 & 0 & 0 
\end{bmatrix}
\]

and

\[
W^T = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

where the superscript, T indicates the transpose of the matrix.

The sizes of G and C matrices are only 7x7 as opposed to 11x11 that would have resulted with the direct application of MNA. Thus the memory requirement to store these matrices is only 40.5% compared to the original MNA matrices. It should be noted that such a reduction is achieved during the formulation stage. For large size networks, the application of the modified method may lead to a large reduction in the memory requirement.
Our next example circuit is shown in Fig. 8. In this circuit, a VVT and a voltage source have a common node. This circuit has 6 non-reference nodes. The $G$, $C$, and $W$ matrices using the new method are

$$G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & -3 \\
0 & -2 & 1 & 0 & 0 & 0 \\
0 & 0 & -5 & 5 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & -4 & 4
\end{bmatrix},$$

$$C = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

and

$$WT = [1 \; 0 \; 0 \; 0 \; 0 \; 0].$$

The sizes of the matrices are only 6x6. It would have been 10x10 if original MNA had been used. Thus the memory requirement to store these matrices is only 36% compared to the original MNA matrices.

**IV. CONCLUSIONS**

In this paper, we have suggested a set of simple rules to modify the formulation method of circuit equations using MNA. These modifications involve the implementation of a few simple additional rules without requiring a lot of bookkeeping procedures. This method reduces the memory requirement for storing the circuit equations as evidenced by the examples. Therefore in micro-computers where memory requirement is critical, this new formulation method enables one to simulate large size networks with less additional programming effort. An additional advantage of this method, the run-time will also be reduced because of the reduced number of equations to be solved for finding frequency response and/or time domain responses.

**REFERENCES**


Fig. 5: Two independent voltage sources having a common node, J

Fig. 6: Two independent voltage sources having a common node, K

Fig. 7: Example circuit-I

Fig. 8: Example circuit-II