A FAST ALGORITHM FOR THE GENERATION OF FAULT DICTIONARY OF LINEAR ANALOG CIRCUITS USING ADJOIN NETWORK APPROACH

V.C. PRASAD
S.N. RAO PINGALA

Deptt. of Electrical, Engg., Indian Institute of Technology, Hauz Khas, New Delhi - 110 016, India.

ABSTRACT

A new method for the generation of fault dictionary of a linear analog circuit is presented. The method is based on the well known adjoint network concept. For an \((n+3)\) node network and for an order of \(n^r\) faults, the new method is of \(O(n')\) time complexity. Currently, a minimum of \(O(n^r)\) is required showing that the new method is extremely fast. Using \(n\) processors, the new method takes \(O(1)\) time on a SIMD shared memory parallel computer.

1. INTRODUCTION

Currently, analog fault diagnosis is popular among researchers. There are broadly two methods for this purpose. They are (i) fault analysis approach or situtation-adjust-first-test (ii) fault dictionary approach or simulation-before-test. In fault analysis approach, the equations of the network are obtained in terms of the unknown parameters to be determined. Substituting the measurements of the faulty network at the accessible nodes into these equations and solving them, the unknown parameters are determined. This gives us the fault. There are several ways of obtaining the equations of the faulty network. The literature is very vast, on this subject. The reader is referred to the references 1-4, 6-11, 13 and other literature for details.

In the fault, dictionary approach, the nominal network and the faulty network for each assumed fault is analysed using a circuit simulator. The information is stored in what is called a fault dictionary. Later, measurements are made on the actual faulty network and then they are matched with the entries in the fault dictionary to determine the fault. Building a comprehensive fault dictionary is highly computation intensive. Hochwald and Bastian [13] used a circuit simulator to obtain the voltages of the faulty network. For an \((n+1)\) node network, each run of the circuit simulator takes \(O(n')\) time. Therefore, for \(k\) faults, it requires \(O(n'^k)\) which is prohibitively large. Lin and El'echrif [12] also used a circuit simulator to build the fault dictionary taut, suggested a different technique for fault isolation. Pahwa and Rohref [5] also presented a technique for developing fault dictionary for a linear analog circuit, taking tolerances also into account. They used Householder's formula to do this. Use of this formula reduces the computation to \(O(-fn)\). Schreiber 13 also studied fault dictionary approach from stimulus design point of view.

In this communication, we present a method for fault dictionary of a linear analog circuit in an entirely different way. We make use of adjoint network approach for this purpose. It turns out to be the fastest method so far known (currently, adjoint network concept is used only in stimulation-after-test approach). Our method requires computation of the order of \(O(fn)\). This is evidently substantial for large networks.

2. ADJOIN NETWORK APPROACH

Consider the linear network \(N\) in Fig.1 which has \(b\) admittances and \((n+i)\) nodes, where \(n + 5\) is the reference node. Let \(N^*\) shown in Fig. 2 denote its adjoint network. Let \(N^*\) be the faulty network whose node voltages are to be determined. Controlled sources of all kinds are allowed in the network. Assume one port between each node and the reference node in all the three networks. Let port \(j\) denote the port between the \(j^{th}\) node and the reference node. For such a network, it can be shown [23] that,

\[
E' (V_{Alj}, -AV_{i}. I_{j} .- S (V_{pj} A_{pj} - V_{pj} I_{pj}) \quad (1)
\]

where \(V_{Ai}\) corresponds to \(k^{th}\) branch voltage and current respectively of \(N(N)\), \(V_{pj}\) \((pj)\) correspond to \(j^{th}\) port voltage and current respectively of \(N(N)\) with the current, flowing out of the positive terminal, \(V_{Bi}\) \((Bi)\) correspond to \(j^{th}\) branch admittance \(AV_{pj}\) etc. Similarly defined. Further, \(1_{k} + \Delta 1_{k} = (V_{k} + \Delta V_{k}) (V_{k} - \Delta V_{k})\)

\[
I_{j} = V_{j} I_{j} + \Delta V_{j} \quad \text{let} \quad I_{j} = V_{j} I_{j}
\]

Substituting these in (1), we have

\[
E' (V_{j} + \Delta V_{j}, -AV_{i} I_{j} . S (V_{pj} A_{pj} - V_{pj} I_{pj}) \quad (2)
\]

We will now use this equation, voltages of adjoint as well as nominal networks and changes in parameter values of faulty components to determine the node voltages of the faulty network. For this purpose do the following : a) Leave all parts open in \(N\) and \(N^*\), i.e., \(I_{j} = 0\) and \(A_{pj} = 0\) for \(j = 1, 2, ..., n\) b) Connect, a one ampere ideal current source to only one port, say \(j\), at a time in the adjoint network such that the current enters node \(j\). Leave the remaining ports open, i.e., \(I_{m} = 0\) for \(m \neq j\) and \(I_{m} = -I_{j}\) for \(m = j\).

3. COMPUTATION OF NODE VOLTAGES OF THE FAULTY NETWORK.

For a single fault, \(\Delta Y_{j} = 0\) except for...
one element. Thus equation (2) becomes

\[(V_+ + AV_+) V_j = AV_+ p_j = AV_p_j \quad (p_j = 1) \quad \ldots (3)\]

where \(V_k\) is \(V_k\) when the excitation is at port \(j\).

**SINGLE ELEMENT OPEN CIRCUIT FAULT**

Let \(k^{th}\) element be open, i.e., \(V_k + AV_k = 0\). Assume that the other elements are at their nominal values. Let \(k1\) and \(k2\) be the nodes to which \(k^{th}\) admittance is connected, 
\[V_{k1} + \Delta V = V_{nk1} - V_{nk2} = AV_{p_k} \quad (k = 1, 2)\]

Further all ports in \(N\) and \(W\) are open and hence we get
\[V_{nk2} = V_j - V_{nk2} = V_p_j - V_{nk2}\]
where \(n\) stands for node. Equation (3) can then be written as

\[V_{nk1} V_{nk2} (V_{nk1} - V_{nk2}) V_k = \sum_{j = 1, 2, n} (4)\]

Putting \(j = k1\) and \(j = k2\), we obtain

\[\Delta V_{nk1} = \Delta V_{nk2} = \Delta V_k = 0(1)\]

Thus this also takes 0(1) time on a SIMD parallel computer of \(n\) processors™.

**DOUBLE FAULT (OPEN CIRCUIT)**

For double fault \(V_{nk1} = 0\), except for two elements, say \(r\) and \(s\), the nodes of \(r, s\) elements are \(r_1, r_2, s_1, s_2\) respectively. Equation (3) now gets modified as

\[V_{nk1} V_{nk2} (V_{nk1} - V_{nk2}) V_k = \sum_{j = 1, 2, n} (5)\]

and

\[AV_{p_k} \quad (k = 3)\]

Proceeding as in the case of single open circuit fault, we get

\[V_{nk1} V_{nk2} (V_{nk1} - V_{nk2}) V_k = \sum_{j = 1, 2, n} (6)\]

Where \(V_{nk1}\) and \(V_{nk2}\) are known, all node voltages of the faulty network are computed using equation (13). Even in this case, it takes \(O(n)\) computations for each double open circuit fault. Thus it takes a total computation of \(O(n^2 + \pi n)\) for a double fault dictionary of \(f\) double open circuit faults.

This also takes 0(1) time on a SIMD parallel computer of \(n\) processors™.

**DOUBLE FAULT (SHORT CIRCUIT)**

Let \(r, s\) elements be short circuited and the nodes of \(r, s\) elements are \(r_1, r_2, s_1, s_2\) respectively. For double fault \(V_{nk1} = 0\).
except for \( r, s \) elements. Equation (7) now gets modified as
\[
(I_{i} + A_{i}) \bar{V}_{ij} + (I_{i} + A_{i}) \bar{V}_{ij} = AV_{j}
\]
for \( j = 1, 2, ..., n \).

Proceeding as in the case of single short circuit and substituting
\[
\bar{V}_{i} - \bar{V}_{j} = (I_{i} + A_{i}) \bar{V}_{ij} + (I_{i} + A_{i}) \bar{V}_{ij}
\]
for \( j = 1, 2, ..., n \), we get
\[
\Delta \bar{V}_{i} = (I_{i} + A_{i}) \bar{V}_{ij} + (I_{i} + A_{i}) \bar{V}_{ij}
\]
and
\[
V_{i} - V_{j} = V_{n}j + AV_{n}j
\]
Further
\[
V_{n}j + AV_{n}j = V_{n}j + AV_{n}j
\]
(16)

As \( f \) and \( a \) elements are short circuited in the faulty network. Putting \( j=r, s \), \( a \) and \( a \) in equation (14) and substituting (16) and (17) and solving the resulting two equations, \( (I_{i} + A_{i}) \) and \( (I_{i} + A_{i}) \) are obtained. Then all node voltages of 'faulty' network are computed using equation (14). This also takes \( O(n) \) computations for each double short circuit fault. Thus the total computation required is \( Q(n^2 + n) \) for \( f \) faults. From equations (14) and (15) it can be seen that once \( (I_{i} + A_{i}) \) and \( (I_{i} + A_{i}) \) are known, all node voltages can be computed in parallel time only if \( n \) processors are available. Thus this also takes \( O(n) \) time on a SIMD parallel computer of \( n \) processors.

Similarly, the method can easily be extended to any number of simultaneous faults of various types.

**COMPUTATION OF ADJOINT NETWORK: VOLTAGES!**

For the adjoint network, we have
\[
\bar{Y}_{i} \bar{V}_{i} = \bar{V}_{i}
\]
where \( \bar{Y}_{i} \) is the node admittance matrix of \( N \)
\( \bar{V}_{i} \) is the node voltage vector of \( N \)
\( j \) is the vector of port currents in \( M \) with one port between each node and the reference-"node. Then
\[
\bar{V}_{i} = \bar{Y}_{i} \bar{V}_{i}
\]

The \( i \)-th column of \( \bar{Y}_{i} \) gives \( \bar{V}_{i} \) when one excitation \( x \) is at port \( i \) and other porb \( w, v, \), etc. Then, sol vino nominal network is enough to get adjoint network voltages also.

**4. EXAMPLE**

Then, the method is illustrated with an example. Consider the nominal network shown in fig. 3 having 3 nodes (excluding reference node), 6 conductances of 1 mho each and a current source of 8 amps, i.e. \( n=3, \), \( b=6, \), \( G=10 \), mho, \( i=1 \). Then the node admittance matrix
\[
\bar{Y}_{i} = \begin{bmatrix}
3 & -1 & -1 \\
-1 & 3 & -1 \\
-1 & -1 & 3 \\
\end{bmatrix}
\]
is symmetric and \( \bar{Y}_{i} \) is symmetric.

\[
\bar{V}_{i} = \begin{bmatrix}
2 \\
1 \\
1 \\
\end{bmatrix}
\]

The node voltages of nominal network can be computed using \( \bar{V}_{i}, \) which are
\[
\bar{V}_{1}=4 \text{ volts, } \bar{V}_{2}=2 \text{ volts and } \bar{V}_{3}=0 \text{ volts.}
\]

Assume that \( G_0 \) is open in the faulty network and nodes of \( N_k \), are 1 and 3, i.e., \( k=6, \), \( k=1 \) and \( k_2=3 \). The adjoint network voltages of \( a \) are detemined from its node voltages. For example,
\[
\bar{V}_{1}(1)=1/4, \bar{V}_{2}(1)=1/4, \bar{V}_{3}(1)=1/4 \text{ volts.}
\]

Similarly, \( \bar{V}^{*}(2) \) and \( \bar{V}^{*}(3) \) can be obtained as 0 and -1/4 volts respectively. Using equation (5), we get
\[
\bar{V}^{*}(1)_{n}=\bar{v}^{*}(1)_{n}+\bar{v}^{*}(1)_{n} \text{ volts.}
\]

Thus this also takes \( O(n) \) time on a SIMD parallel computer of \( n \) processors.

**REFERENCES**


