AN EFFICIENT DC FAULT SIMULATION ALGORITHM FOR NONLINEAR CIRCUITS

Lin Hua (林华) Ling Xieting (凌贤亭) Pan Mingde (潘明德)
(Electronic Engineering Department, Fudan University)

Abstract

An efficient DC fault simulation algorithm for nonlinear circuits is proposed in which Householder's transform and the method of partitioned matrices are used to minimize the number of operations for fault simulation. The algorithm is also applicable to DC analysis for nonlinear circuits.

I. Introduction

In recent years the subject of fault location in analog circuits has been of great interest to researchers, and a number of methods have emerged\[1,2\]. Dictionary technique\[3\] is efficient in some real application in which on-line test is undertaken by using a microcomputer. One reason is that the large amounts of calculation are not needed after test, and the only need is the matching simply between the measurements and the "words" in fault dictionary. The other is that it is suitable to either linear or nonlinear circuits. In order to create the fault dictionary, time consuming calculations are required for fault simulation. With Householder's transform, G. O. Temes\[4\] proposed an efficient algorithm for fault simulation of linear circuits. The idea is that fault simulation is to solve equations in which only few parameters are changed, and simulation results at fault-free can be used in calculation to minimize the number of operations for fault simulation.

For the sake of understanding, we will briefly introduce G.C. Temes' algorithm in the next section. Our contribution in this paper is to apply Householder's formula further in direct current fault simulation. These kind of simulation in fact is to solve nonlinear equations because DC faults in circuits often cause the change of the operating points. Solving a lot of nonlinear equations will take excessive computer time. This drive us to find an efficient algorithm for nonlinear fault simulation. In section III linearized iterative equations for nonlinear circuits are set up and show the similarity between these equations and the equations in section II. In section IV, an efficient algorithm for fault simulation is given and its computational complexity is compared with other algorithm's.

II. Fault Simulation for Linear Circuits

To illustrate the ideas more simply, we use the nodal analysis here. In general, a set of nodal admittance equations takes the form of
\[ Y \cdot V = I, \] (1)

where \( Y \) is the nodal admittance matrix, \( V \) the voltage vector, and \( I \) the equivalent current vector. The faulted nodal admittance relation has the same form

\[ Y_f \cdot V_f = I_f, \] (2)

Where the subscript \( f \) represents the faulted cases. We simply suppose

\[ I_f = I, \] (3)

and get

\[ Y_f \cdot V_f = I. \] (4)

Rewriting the faulted nodal admittance \( Y_f \) as the sum of two terms in which one has no relation with faults while the other has, we obtain

\[ Y_f = Y + A_f \cdot \Delta Y_b \cdot A_f^T, \] (5)

where \( \Delta Y_b \) is a diagonal matrix whose elements is the changes of the faulted components

\[ \Delta Y_b = \text{diag}(\Delta y_1, \Delta y_2, \ldots, \Delta y_m), \] (6)

in which \( m \) is the number of faulted components. \( A_f \) is an incidence matrix composed of faulted branches

\[ A_f = \text{col}(a^1, a^2, \ldots, a^m), \] (7)

where

\[ a^i(j) = \begin{cases} 1, & \text{if the current at the } i \text{-th faulted branch flows out of node } j; \\ -1, & \text{if the current at the } i \text{-th faulted branch flows into the node } j; \\ 0, & \text{the other cases}. \end{cases} \]

From Eqs. (4) and (5), we get

\[ V_f = (Y + A_f \cdot \Delta Y_b \cdot A_f^T)^{-1} I. \] (8)

Of course, the Gaussian elimination algorithm and the LU-factorization algorithm are good algorithms to solve equations such as Eq. (8). However, noticing the feature of Eq. (8), G. C. Temes applied Householder's transform to reduce the order of equations and get an efficient algorithm for linear circuit simulation.

The general form of Householder's transform is

\[ (Y + U \cdot S \cdot W)^{-1} = Y^{-1} - Y^{-1} \cdot U \cdot (S^{-1} + W \cdot Y^{-1} \cdot U)^{-1} \cdot W \cdot Y^{-1}, \] (9)

where \( Y \) has order \( n \times n \), \( U \) and \( W \) have order \( n \times m \), \( S \) has order \( m \times m \), \( n \) is the number of nodes. Using Householder's transform Eq. (8) can be written as

\[ V_f = [1 - Y^{-1} \cdot A_f \cdot (\Delta Y_b^{-1} + A_f^{-T} \cdot Y^{-1} \cdot A)^{-1} \cdot A_f^T] \cdot V_o, \] (10)

where

\[ V_o = Y^{-1} \cdot I. \] (11)

To solve Eq. (10) is much easier than to solve Eq. (8) because

(1) in general, the number of fault branches will small, that is \( m \ll n \);

(2) \( V_o \), the nodal voltage vector at fault-free which never changes in fault simulation, can be stored in memory;

(3) \( Y^{-1} \), the inverse of nodal admittance matrix, \( Y \), at fault-free, can also be stored in memory;