ZESZYTY
NAUKOWE
POLITECHNIKI
ŚLĄSKIEJ

P. 4474/07

Janusz A. STARZYK

TOPOLOGICAL ANALYSIS AND DIAGNOSIS OF ANALOG CIRCUITS

GLIWICE 2007

POLITECHNIKA ŚLĄSKA
ZESZYTY NAUKOWE
NR 1771

P. 4474/07

Janusz A. STARZYK

TOPOLOGICAL ANALYSIS AND DIAGNOSIS OF ANALOG CIRCUITS

GLIWICE 2007

Opiniodawcy
Dr hab. inż. Leon LASEK, prof. nzw. w Politechnice Śląskiej
Dr hab. Jerzy SZABATIN, prof. Politechniki Warszawskiej

Kolegium redakcyjne
REDAKTOR NACZELNY – Prof. dr hab. inż. Andrzej BUCHACZ
REDAKTOR DZIAŁU – Doc. dr inż. Zdzisław POGODA
SEKRETARZ REDAKCJI – Mgr Elżbieta LEŚKO

Wydano za zgodą
Rektora Politechniki Śląskiej

PL ISSN 1231-1596

© Copyright by
Wydawnictwo Politechniki Śląskiej
Gliwice 2007

P 17108

Contents

List of Selected Symbols and Abbreviations 7
1. Introduction 9
2. Graphs and Network Topology 12
2.1. Graph Representations of Electronic Circuits 12
2.2. Graph Decomposition 17
2.2.1. Types of Graph Decomposition 18
2.2.2. Algorithms of Graph Decomposition 19
3. Topological Analysis 21
3.1. Foundations of Topological Analysis 21
3.2. Signal-Flow Graph Analysis of Electronic Circuits 25
3.2.1. Direct Signal-Flow Graph Analysis 29
3.2.2. Descending Hierarchical Signal-Flow Graph Analysis 30
3.2.3. Ascending Hierarchical Signal-Flow Graph Analysis 33
3.3. Directed Graph Analysis of Electronic Circuits 35
3.3.1. Direct Directed Graph Analysis 37
3.3.2. Descending Hierarchical Directed Graph Analysis 38
3.3.3. Ascending Hierarchical Direct Graph Analysis 40
3.4. Conjugated Graph Analysis of Electronic Circuits 41
3.4.1. Direct Conjugated Graph Analysis 42
3.4.2. Descending Hierarchical Conjugated Graph Analysis 43
3.5. Algorithms for Topological Analysis Methods 45
3.5.1. Multiconnections of a Signal-Flow Graph 46
3.5.2. Multitrees of a Directed Graph 48
3.6. Other Advances in Topological Analysis Methods 49
3.6.1. Hierarchical Analysis of High Frequency Interconnect Networks 50
3.6.2. Large Change Sensitivity Based Diakoptic of Large Analog Networks 57
3.7 Summary of Topological Analysis Methods 65
4. Topological Diagnosis 68
4.1. Fault Location in Nonlinear Networks 69
4.1.1. Network Decomposition 70
4.1.2. Faulty Regions 74
4.2. Fault Location in Linear Networks 77
4.2.1. Location of Faulty Elements 78
4.2.2. Fault Location by Nodal Analysis 79
4.2.3. Topological Conditions for Node Fault Diagnosis 81
4.2.4. Parameter Tolerances 83
4.2.5. Multiport Approach to Fault Location 84
4.3. Sensitivity Approach 87

4.3.1. Decomposition Approach to Sensitivity Based Testing 89
4.4. Fault Verification in Multiple-Fault Diagnosis of Linear Networks 94
4.4.1. Fault Diagnosis Equations 95
4.4.2. Fault Diagnosis Process 98
4.4.3. Large Change Sensitivity in Fault Diagnosis 104
4.5. Low Testability Analog Circuits 108
4.5.1. Fault Verification by Locating Ambiguity Groups 112
4.5.2. Fault Diagnosis in Mixed-Signal Low Testability System 114
4.6. Entropy Based Optimum Test Points Selection 116
4.6.1. Integer Coded Dictionary 117
4.6.2. Entropy Test Point Selection 118
4.6.3. Statistical Experiments 120
4.7. Summary of Topological Diagnosis Methods 121
5. Concluding Remarks 123
References 125
Summary 140

Spis Treści

Wykaz Wybranych Symboli i Skrótów 7
1. Wstęp 9
2. Grafy i Topologia Układu 12
   2.1. Grafy Układów Elektronicznych 12
   2.2. Dekompozycja Grafu 17
      2.2.1. Rodzaje Dekompozycji Grafu 18
      2.2.2. Algorytmy Dekompozycji Grafu 19
3. Analiza Topologiczna 21
   3.1. Podstawy Analizy Topologicznej 21
   3.2. Analiza Układów Elektronicznych przy Reprezentacji Grafem Przepływu Sygnału. 25
      3.2.1. Analiza Bezpośrednia przy Reprezentacji Grafem Przepływu Sygnału 29
      3.2.2. Analiza Hierarchiczna Zstępująca przy Reprezentacji Grafem Przepływu Sygnału 30
      3.2.3. Analiza Hierarchiczna Wstępująca przy Reprezentacji Grafem Przepływu Sygnału 33
   3.3. Analiza Układów Elektronicznych przy Reprezentacji Grafem Skierowanym 35
      3.3.1. Analiza Bezpośrednia przy Reprezentacji Grafem Skierowanym 37
      3.3.2. Analiza Hierarchiczna Zstępująca przy Reprezentacji Grafem Skierowanym 38
      3.3.3. Analiza Hierarchiczna Wstępująca przy Reprezentacji Grafem Skierowanym 40
   3.4. Analiza Układów Elektronicznych przy Reprezentacji Parą Grafów Sprzężonych 41
      3.4.1. Analiza Bezpośrednia przy Reprezentacji Parą Grafów Sprzężonych 42
      3.4.2. Analiza Hierarchiczna Zstępująca przy Reprezentacji Parą Grafów Sprzężonych 43
   3.5. Algorytmy Metod Analizy Topologicznej 45
      3.5.1. Wielo-połączenia Grafu Przepływu Sygnału 46
      3.5.2. Wielo-drzewa Grafu Skierowanego 48
   3.6. Inne Usprawnienia Metod Analizy Topologicznej 49
      3.6.1. Analiza Hierarchiczna Sieci Połączeń Wysokiej Częstotliwości 50
      3.6.2. Diakoptyka Dużych Układów Analogowych w Oparciu o Metodę Wrażliwości Wielkoprzyrostowej 57
   3.7 Podsumowanie Metod Analizy Topologicznej 65
4. Diagostyka Topologicza 67
   4.1. Lokalizacja Uszkodzeń w Układach Nieliniowych 68
      4.1.1. Dekompozycja Układu 69
      4.1.2. Obszary Uszkodzeń 73
   4.2. Lokalizacja Uszkodzeń w Układach Liniowych 76
      4.2.1. Lokalizacja Uszkodzonych Elementów 77
      4.2.2. Lokalizacja Uszkodzeń przy Pomocy Analizy Węzłowej 78

4.2.3. Warunki Topologiczne Diagnostyki przy Pomocy Analizy Węzłowej 80
4.2.4. Tolerancje Parametrów 82
4.2.5. Metody Wielo-wrotnikowe Lokalizacji Uszkodzeń 83
4.3. Metody Wrażliwościowe 86
4.3.1. Testowanie Wrażliwościowe z Dekompozycją 88
4.4. Weryfikacja Uszkodzeń w Diagnostyce Układów Liniowych z Wieloma Uszkodzeniami 93
4.4.1. Równania Diagnostyczne 94
4.4.2. Proces Diagnozowania Uszkodzeń 97
4.4.3. Diagnozowanie Uszkodzeń w Oparciu o Metodę Wrażliwości Wielkoprzyrostowej 103
4.5. Układy Analogowe o Niskiej Testowalności 107
4.5.1. Weryfikacja Uszkodzeń poprzez Lokalizację Grup Wieloznacznych 111
4.5.2. Diagnozowanie Uszkodzeń w Układach Mieszanych o Niskiej Testowalności ..113
4.6. Optymalny Wybór Punktów Pomiarowych w Oparciu o Entropię 115
4.6.1. Słownik Kodowany Liczbami Całkowitymi 116
4.6.2. Wybór Punktów Pomiarowych w Oparciu o Entropię 117
4.6.3. Eksperymenty Statystyczne 119
4.7. Podsumowanie Metod Diagnostyki Topologicznej 120
5. Wnioski Końcowe 122
Bibliografia 124
Streszczenie 139

List of Selected Symbols and Abbreviations

A	coefficient matrix
$\|A\|$	determinant of coefficient matrix
B	test verification matrix
BDF	backwards differentiation formula
BiMOS	bipolar MOS
$\underline{C}$	set of connections
$\overline{\overline{C}}$	cardinality of the set C
$\|C\|$	weight of the set C
c^p	proper multiconnection
c_W	multiconnection
CPU	central processing unit
CUT	circuit under test
CAD	computer aided design
CMOS	complementary MOS
d	symbolic denominator terms
DFT	design for test
E	set of graph edges, entropy
e	unit vector
F	transfer function, set of faults
FET	field effect transistor
f	number of faults
$G(V,E)$, G	graph
G_i^s	substitute graph of subgraph G_i
$G_1 \oplus G_2$	direct sum of subgraphs
$G_1 \cap G_2$	intersection of subgraphs
$G_1 \cup G_2$	union of subgraphs
$G_1 - G_2$	difference of subgraphs
GMTC	generalized mutual-testing condition
H	hybrid matrix
HOS	hierarchically organized structure
h	characteristic function
I	current vector
IC	integrated circuit
J	independent current excitations
K_{vv}	voltage gain
KCL	Kirchhoff current law
KVL	Kirchhoff voltage law
L	symbolic numerator terms
l	decomposition level

M	measurement set
m	number of measurements
MOS	metal-oxide semiconductor
MTC	mutual-testing condition
n	number of nodes
O	order of complexity
P	set of path, current graph incidence matrix
Q	voltage graph incidence matrix
R	upper triangular matrix
S	set of subgraphs, subnetworks
s	complex frequency variable
SAT	simulation after test
SBT	simulation before test
SC	switched capacitor
SOC	system on chip
STC	self-testing condition
T	set of trees, transmission matrix
t	time variable
t^p	proper multitree
t_V	multitree
V	set of graph vertices (nodes), voltage vector
VLSI	very large scale integrated circuits
W	set of pairs of vertices, excitation vector
X	solution vector
Y	admittance
Z	impedance
δ	parameter deviation
ϕ	vector of network parameters
λ	indefinite incidence matrix
$\varnothing$	empty set

1. Introduction

Did you ever wonder what graphs, Kirchhoff laws, the Internet, rough sets, neural networks, and brain organization have in common? The answer may be very simple - the system topology. Whether it is a flow-graph that describes the flow of signals between the nodes of a graph, Kirchhoff laws that describe relationships between currents or voltages in an electronic network, or the Internet that uses a web of interconnected computers to move packets of data between the end users, they all rely on specific topological information about the system structure. A similar argument can be used for rough sets that describe features of the information system; neural networks that implement the connectionist concept of massively parallel interconnect structures of processing elements; or the human brain - the most complex, and still only sketchily described, system of interconnected neurons. In all of these systems topology determines how the system operates. Topology is a silent system of constraints imposed on an electronic network, governing the signal flow between its components. Thus it is used in all aspects of system design from system analysis and synthesis through diagnosis.

Over many years computer analysis of large analog circuits was an important research topic presented in many monographs and research papers [23], [55], [56], [104], [138], [174], [193], [203], [281]. The main objective of these works was to improve computational efficiency of the computer analysis methods (like accuracy of the results, analysis time, memory requirements, numerical stability and convergence, etc.) and to obtain full, accurate, and illustrative information about the analyzed circuits. These were also the objectives of the symbolic or semi symbolic network analyses [5], [83], [145], [205], [206], [215], [216]. Since it was difficult to develop effective programs of topological analysis for large networks, the development in these years was focused on numerical methods for sparse matrices [27], [55], [99], [105] or eigenvalues methods [41], [121], [139], [155], [177], [202], [218]. Since then computer aided analysis and computer aided design of electronic circuits developed into a leading industry behind the microelectronic revolution with many professional conferences, design tools, software vendors, design houses, and fabrication facilities. In this development the symbolic analysis methods played an important role.

Also for many years the analog fault diagnosis and fault location in analog circuits have been challenging tasks for both design researchers and practitioners [69], [259]. Relationships between the input and output signals in analog circuits are obscure compared to precise relationships in digital circuits. Statistical distribution of faults or their character is unpredictable. Changes in the circuit response are not linear functions of changes in the parameter values even if a circuit is linear. Modern VLSI technology integrates many thousands of analog components sharing the same silicon substrate with even more numerous digital components, with relatively few points accessible for measurements. Lack of access to the internal points for measurements and lack of good fault models are making design for test difficult, and the analog nature of parameter changes compounds the problem. Over the years importance of the analog testing grew. As the testing costs started to exceed the design costs, the integrated circuit industry needed more effective design for test techniques and good testing standards. Many

conferences devoted to design automation and test have been organized, attracting constant attention and research efforts in this area.

Topological analysis of electronic circuits relates to learning circuit properties based on the circuit components and the way they are connected (circuit topology) without using numerical methods to solve the circuit equations. It results in transfer functions of the analyzed circuits that represent ratio of the Laplace transform of the output and input signals. Network topology can also be used indirectly to aid the circuit analysis or to simplify organization of its numerical analysis. Topological diagnosis uses network topology to determine testability conditions, isolate faulty subnetworks and locate faulty parameters, determine test coverage and test point selection, and identify ambiguity conditions. It can do so without regard the amount of parameter deviations from their nominal values.

The main objective of this work is to show how topological methods can be used in the analog circuit analysis and testing techniques, how they can enrich the software tools used in computer aided analysis, and how they can enhance the design for testability process. More specific objective of this work is to address the problems of topological analysis of large analog networks, considering various topological representations of the circuit elements. These problems include the development of effective methods and algorithms of topological analysis of large electronic networks that lead to analysis time comparable with time needed for the numerical analysis. The practical need to consider various topological representations stems from the differences in the topological analyses, different treatment of such representations in the literature, and problems with identifying the optimum representation for the topological analysis. Another specific objective is to show how the network topology can be used to develop the testing tolls for analog circuits. These include the development of testability requirements, formulation of effective test equations, selection of test points, multiple fault verification techniques, fault location in large analog networks to identify faulty nodes and faulty components, and an effective treatment of the low testability circuits. Presented approaches to reach these objectives were proposed first at the beginning of nineteen eighties [226] and elaborated and extended over the years to encompass the above mentioned problems [10], [40], [77], [133], [134], [148], [182], [201], [221], [223], [224], [227]-[242], [244]-[248], [251]. Development of computer analysis programs and inclusion of the discussed topological approaches in the software tools [21], [37]-[39], [98], [132], [222], [225], [234], [243] facilitated the use of the proposed methods in practical projects. Fragments of such projects are presented in this work to illustrate principles and forms of topological network analysis and diagnosis.

The work focuses on author's research work in topological analysis and diagnosis. It tries to put a framework over two disciplines related through topological treatment of the discussed analysis and testing problems. It explores dependencies that relate a topological description to the flow of information between the processing nodes, and provides elegant, mathematically simple relationships, independent of the system size and complexity. Yet system size and complexity affect the computational effort needed to compute the results using the network topology. Therefore, special attention is paid to developing algorithmic approaches that facilitate working with large systems. This work discusses only major results that author obtained in topological analysis and diagnosis, with references to published work for further details. When needed for clarity of

presentation, these results are illustrated with sample applications. The author's most recent quest to apply topology to modeling, simulation and design issues of building brain-like intelligent machines is not discussed here. These issues, although extremely important and very challenging, still await their successful, system level solutions.

The thesis presented in this work is that topological analysis and diagnosis methods significantly contribute to modern design and test procedures. One of the objectives of this work is to bring these methods into the light.

This work extensively uses terminology and tools of graph theory [18], [44], [45], [51], [53], [65], [96], [162], [164], [212]. The second chapter relates graph theory and network topology. It introduces three topological representations of the network equations based on the flow graphs, directed graphs, and conjugated linear graphs. It also defines the stamp models of linear circuit elements. Finally it presents various types of graph decomposition and an efficient algorithm of hierarchical decomposition.

The third chapter is devoted to topological analysis. It presents general characteristics of topological analysis methods and describes development of these methods pointing the main results that facilitated their growth. It also presents methods and computational techniques used in the three types of topological representations. Specifically, it discusses methods and techniques for a direct graph analysis, as well as analysis with decomposition. Two types of hierarchical analysis – descending and ascending methods are described. Subsequently, efficient methods and related algorithms to generate multitrees and multiconnections required in network analysis are presented. The last section of this chapter discusses topological methods used in analysis of distributed interconnection networks, and explains the network diakoptics using a hierarchical decomposition of its graph and the large change sensitivity method.

Chapter four focuses on the methods of topological diagnosis. It starts from discussing a fault location in nonlinear networks based on the network decomposition and location of faulty regions. This is followed by specialized methods of fault location in linearized networks. They contain such issues as the location of faulty elements, fault location by nodal analysis, topological conditions for fault diagnosis in nodal analysis, consideration of parameter tolerances in fault diagnosis, and fault location using the multiport representation. Subsequently, a sensitivity approach to time domain testing is presented. It uses the network topology and a hierarchical decomposition of the test equations to simplify the test equations and to improve accuracy of the test results. The next section in this chapter develops fault verification in multiple-fault diagnosis. It presents a fault diagnosis process based on the verification method and uses the large change sensitivity approach to develop a new test verification method for the analog fault diagnosis. This is followed by the presentation of ambiguity groups finding technique and a fault diagnosis approach to low testability circuits. The method uses the network topology to identify fault-free subnetworks and proceeds to identification of faulty components using fault location techniques. The last section in this chapter presents entropy based test point selection.

Concluding remarks are presented in Chapter five.

2. Graphs and Network Topology

Topological methods of circuit analysis and diagnosis relate to the study of electronic circuits. Study of these methods was initiated by Kirchhoff [123] at the end of the 19th century and intensified in the sixties and seventies [71], [119], [48], [173], [24], [25], [63], [254], to a large degree due to the development of computer technology and related devices requiring advanced methods of electronic circuit analysis and design.

Concurrently, algebraic methods that represent network topology and can be used for its analysis were developed, most notably by Wang [272], [258], [261], [68] and Bellert [14], [15], [16], [17], [87]. The most attractive feature of topological methods at this early stage of their development was their ability to obtain transfer functions directly from the circuit netlist or from its graph description.

Graph based methods used signal flow graphs (Coates [57], [58] and Mason's [157], [158] graphs), linear graphs (current-voltage [163], [212] and nullator-norator [62], [63] graphs), and directed graphs (unistor [159], [45] and distor [47], [45] graphs) to describe network topology. Specialized analysis methods were developed for each of these graph representations and there was no unifying method that would handle these various representations or reuse results from one form of graph representation to another. The presented work illustrates a unifying approach to circuit analysis using network topology and its various graph representations. First, basic notations from graph theory are presented to provide a tool for network analysis and diagnosis.

2.1. Graph Representations of Electronic Circuits

Three major types of graph representations are used to describe electronic circuit topology. These three major types and some of their better know subtypes are as follows:

1. Flow graphs
    a. Coates' graph
    b. Mason's graph
2. Directed graphs
    a. Unistor graph
    b. Dispersor graph
3. Conjugated graphs
    a. Current-voltage graph
    b. Nullator-norator graph

These graphs represent both the interconnection structure of an electronic circuit and its element values. Since topological analysis and diagnosis is performed on a linear system, it is assumed that the circuit elements represented by the graph are linearized around their DC operating points. Thus, in general, graphs represent circuit interconnect structures and associated linearized element values.

Without loss of generality we may assume that, at a given operating point, an electronic circuit is described by the modified nodal equations with the coefficient matrix

$$A=PYQ. \tag{2.1}$$

The coefficient matrix A can be directly obtained from the element equations using the "stamp approach" as described in [267]. In this approach, linearized element models are as described in Table 2.1.

These element models are directly inserted and their symbolic values are added to other element values at the corresponding locations of the modified nodal matrix. P and Q are the topological matrices that indicate location of the element's parameter value (element value) in the modified nodal coefficient matrix A. Except for the reference node (ground), typically all element values are placed at most at 4 locations in A using the stamp approach. More specifically, for each element its symbolic value Y is placed on the intersection of rows i and j and columns k and l as in the following stamp matrix (all remaining elements of the stamp matrix are zero):

$$S_Y = \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{bmatrix} . & \overset{k}{.} & . & . & \overset{l}{.} & . & . \\ . & Y & . & . & -Y & . & . \\ . & . & . & . & . & . & . \\ . & -Y & . & . & Y & . & . \\ . & . & . & . & . & . & . \end{bmatrix}, \tag{2.2}$$

Each circuit element has a single column in topological matrices P and Q that represent information about element's interconnections. More specifically, for an element described by the stamp (2.2), matrix P contains 1 and -1 in rows i and j and matrix Q contains 1 and -1 in rows k and l with all other elements equal to 0. Thus, if a circuit model has b elements, and it is described by $n \times n$ modified nodal matrix T, then both P and Q matrices are $n \times b$ matrices.

Some stamps may contain not only the element value (represented in (2.1) by Y) but constant values as well. These constant values do not affect matrices P and Q. In a circuit model with passive two-terminal circuit components (R, L, and C) only, matrix P is equal to matrix Q, and it is known as the incidence matrix.

The three types of graphs (signal-flow graphs, directed graphs, and conjugated graphs) can be easily obtained from the modified nodal stamps, and their major types are clearly established based on how they appear in the modified nodal matrix.

Each edge of a graph may describe one, two, or four elements of the coefficient matrix depending on whether it is in a signal-flow graph, directed graph or pair of conjugated graphs. Flow graphs represent each element of the coefficient matrix as an independent edge of the graph, resulting in relatively complex graphs that contain as many edges as the number of nonzero entries in the coefficient matrix. Directed graphs may represent two elements of the coefficient matrix, either in a single row (dispersor graph) or a single column (unistor graph) as an independent edge of the graph. Finally, the conjugated graphs represent four elements of the coefficient matrix as an independent edge of the graph, resulting in graphs that contain the minimum number of edges.

Table 2.1

Stamp models of linear circuit elements

Element	Symbol	Stamp	Equations
Current Source	j, J, j'	$\begin{matrix} j \\ j' \end{matrix}\begin{bmatrix} -J \\ J \end{bmatrix}$ SOURCE VECTOR	$I_j = J$ $I_{j'} = -J$
Voltage Source	j, +, I, E, j', -	$\begin{matrix} & V_j & V_{j'} & I \\ j & & & 1 \\ j' & & & -1 \\ m+1 & 1 & -1 & \end{matrix}$ SOURCE VECTOR $\begin{bmatrix} \\ \\ E \end{bmatrix}$	$V_j - V_{j'} = E$ $I_j = I$ $I_{j'} = -I$
Admittance	j, +, I, Y, V, j', -	$\begin{matrix} & V_j & V_{j'} \\ j & Y & -Y \\ j' & -Y & Y \end{matrix}$	$I_j = Y(V_j - V_{j'})$ $I_{j'} = -Y(V_j - V_{j'})$
Impedance	j, +, I, Z, V, j', -	$\begin{matrix} & V_j & V_{j'} & I \\ j & & & 1 \\ j' & & & -1 \\ m+1 & 1 & -1 & -Z \end{matrix}$	$V_j - V_{j'} - ZI = 0$ $I_j = -I_j' = I$
Open Circuit	j, +, V, j', -		$V = V_j - V_{j'}$
Short Circuit	j, I, j'	$\begin{matrix} & V_j & V_{j'} & I \\ j & & & 1 \\ j' & & & -1 \\ m+1 & 1 & -1 & \end{matrix}$	$V_j - V_{j'} = 0$ $I_j = I$ $I_{j'} = -I$
Nullator	j, j'	$\begin{matrix} & V_j & V_{j'} \\ & & \\ m+1 & 1 & -1 \end{matrix}$	$V_j - V_{j'} = 0$ $I_j = I_j' = 0$
Norator	j, j'	$\begin{matrix} & I \\ j & 1 \\ j' & -1 \end{matrix}$	V and I are arbitrary

Element	Symbol	Stamp	Equations
VCT	j, j', k, k', +, -, V, gV	$\begin{array}{c\|cc} & V_j & V_{j'} \\ \hline k & g & -g \\ k' & -g & g \end{array}$	$I_j = 0$ $I_j' = 0$ $I_k = g(V_j - V_j')$ $I_{k'} = -g(V_j - V_j')$
VVT	j, j', k, k', +, -, V, μ V	$\begin{array}{c\|cccc:c} & V_j & V_{j'} & V_k & V_{k'} & I \\ \hline j & & & & & \\ j' & & & & & \\ k & & & & & 1 \\ k' & & & & & -1 \\ \hdashline m+1 & -\mu & \mu & 1 & -1 & \end{array}$	$-\mu V_j + \mu V_{j'} + V_k - V_{k'} = 0$ $I_k = I$ $I_{k'} = -I$
CCT	j, j', k, k', I, α I	$\begin{array}{c\|cccc:c} & V_j & V_{j'} & V_k & V_{k'} & I \\ \hline j & & & & & 1 \\ j' & & & & & -1 \\ k & & & & & \alpha \\ k' & & & & & -\alpha \\ \hdashline m+1 & 1 & -1 & & & \end{array}$	$V_j - V_{j'} = 0$ $I_j = -I_{j'} = I$ $I_k = -I_{k'} = \alpha I$
CVT	j, j', k, k', I_1, rI_1	$\begin{array}{c\|cccc:c:c} & V_j & V_{j'} & V_k & V_{k'} & I_1 & I_2 \\ \hline j & & & & & 1 & \\ j' & & & & & -1 & \\ k & & & & & & 1 \\ k' & & & & & & -1 \\ \hdashline m+1 & 1 & -1 & & & & \\ m+2 & & & 1 & -1 & -r & \end{array}$	$V_j - V_{j'} = 0$ $V_k - V_{k'} - rI_1 = 0$ $I_j = -I_{j'} = I_1$ $I_k = -I_{k'} = I_2$
Operational Amplifier	j, j', k, k', I, ∞	$\begin{array}{c\|cccc:c} & V_j & V_{j'} & V_k & V_{k'} & I \\ \hline j & & & & & \\ j' & & & & & \\ k & & & & & 1 \\ k' & & & & & -1 \\ \hdashline m+1 & 1 & -1 & & & \end{array}$	$V_j - V_{j'} = 0$ $I_k = -I_{k'} = I$
Converter	j, j', k, I	$\begin{array}{c\|cccc:c} & V_j & V_{j'} & V_k & V_{k'} & I \\ \hline j & & & & & 1 \\ j' & & & & & -1 \\ k & & & & & -K_2 \\ k' & & & & & K_2 \\ \hdashline m+1 & 1 & -1 & -K_1 & K_1 & \end{array}$	$V_j - V_{j'} - K_1V_k + K_1V_{k'} = 0$ $I_j = -I_{j'} = I$ $I_k = -I_{k'} = -K_2 I$ For ideal transformer $K_1 = K_2 = n$
Transformer	j, j', k, k', M, I_1, I_2, L_1, L_2, V_1, V_2	$\begin{array}{c\|cccc:c:c} & V_j & V_{j'} & V_k & V_{k'} & I_1 & I_2 \\ \hline j & & & & & 1 & \\ j' & & & & & -1 & \\ k & & & & & & 1 \\ k' & & & & & & -1 \\ \hdashline m+1 & 1 & -1 & & & -sL_1 & -sM \\ m+2 & & & 1 & -1 & -sM & -sL_2 \end{array}$	$V_j - V_{j'} - sL_1I_1 - sMI_2 = 0$ $V_k - V_{k'} - sMI_1 - sL_2I_2 = 0$ $I_j = -I_{j'} = I_1$ $I_k = -I_{k'} = I_2$

Table 2.2 contains various types of example graphs of a passive two-terminal element and a voltage controlled current source.

Table 2.2

Graphs of a two-terminal element and a voltage controlled current source

	Single graph edge	Example stamp	
		Two-terminal element (R,L, or C) $I_i = (V_i - V_j)Y$ $I_j = -(V_i - V_j)Y$	Voltage controlled curent source $I_i = (V_k - V_l)Y$ $I_j = -(V_k - V_l)Y$
Coates' graph	$I_j = V_i Y$ $I_i = 0$		
Unistor graph	$I_i = V_i Y$ $I_j = -V_i Y$		
Current-voltage graph	$I_i = (V_k - V_l)Y$ $I_j = -(V_k - V_l)Y$		

Using the stamp approach based on the modified nodal equations [267] all types of graphs can be directly obtained. A complete set of Coates' graph stamps were presented by Starzyk in [226] using so called transitor models. Chen [45] presented a subset of unistor graph models for elements with admittance description only. In [38] this description was extended to include all elements with modified nodal equations using formal unistor models. In addition, Starzyk introduced dispersor graphs and used them for topological analysis in [231]. Seshu [212] showed how to obtain current-voltage graphs and Davies [61] presented nullator-norator networks of the controlled sources. Finally, the conjugated norator-nullator graphs and rules to use them in topological

analysis were presented by Starzyk in [227]. These stamp based models facilitated topological analysis by automating the graph creation process, a critical step in computer based topological analysis.

Comment. Dispersor graphs were introduced in [231] to use topological methods for analysis of electronic circuits with ideal op-amps. Although unistor models of electronic elements were known in the literature, only the introduction of formal unistor models [37] permitted to model ideal op-amps and other active elements for which unistor models did not exist. Other forms of graphical representations based for instance on the tableau equations can be used, however, they lead to larger graphs and, in general, require more effort to analyze.

2.2. Graph Decomposition

Graph decomposition is used in many applications dealing with large systems like linear programming [59], [112] or the shortest path problem [116], [94], [85], information encoding [59], synthesis of VLSI circuits [54], partition of sparse matrices [190], job shop scheduling [282], gene assembly [72], software synthesis [126], etc. Its aim is to improve the algorithmic performance of problems represented by a system graph. Network decomposition is used in analysis of computer and communication networks [74], [81], [122], [151], [172], [180]. Decomposition plays an important role in stability analysis of large systems [30], [97], [66] or layout compaction in very large scale integrated (VLSI) circuits [243]. In circuit analysis we distinguish diakoptics [138], [175], [251], generalized hybrid analysis [55], and topological analysis with nodal decomposition [134], [135].

Efficient topological analysis and diagnosis may also require graph decomposition into smaller blocks with rules established to govern block level analysis and merging the partial results obtained on the subgraph level. Large complex systems, such as VLSI circuits, contain many subsystems in which subsystem components strongly interact with each other, while elements of different subsystems interact weakly or not at all. This subsystem structure, typical for modern hierarchical design techniques, facilitates system partitioning. However, quite often boundaries between subsystems are not explicitly defined, or the subsystem boundaries may not yield the best decomposition quality. Decomposition quality may be evaluated based on the number of partition nodes [97], [203], number of partition edges [81], [55], or other decomposition parameters [95], [122]. Many algorithms of graph decomposition try to provide good decomposition quality as it affects the efficiency of circuit analysis and diagnosis. However, finding the optimum decomposition is computationally expensive. Instead heuristic algorithms are used to find acceptable (suboptimum) decompositions of the network graph.

Major results in graph decomposition that are applicable to topological analysis of complex electronic networks are briefly discussed here.

2.2.1. Types of Graph Decomposition

There are three types of graph decompositions:

1. Nodal decomposition.

Nodal decomposition illustrated in Fig. 2.1 is the most popular decomposition applicable to all types of topological analysis. The graph is partitioned into edge disjoint subgraphs (known as blocks) connected by common nodes (known as block vertices). Graph nodes and vertices will be used in this work interchangeably as both are used in graph theory and related topological methods.

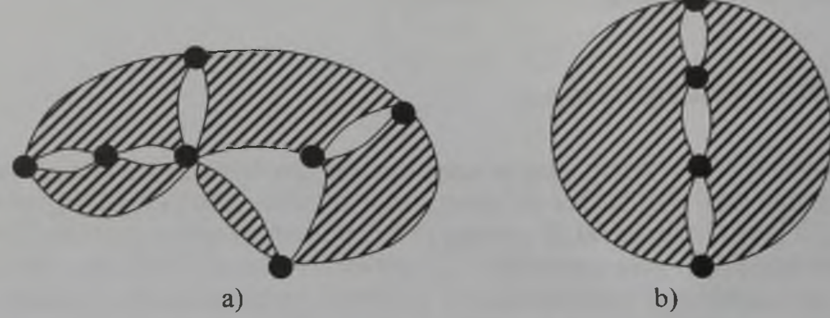


Fig. 2.1. a) nodal decomposition, b) bisection

2. Edge decomposition.

In edge decomposition a graph is partitioned into edge disjoint subgraphs connected by common edges (know as partition edges). Partition edges are cutsets as defined in [242]. Block vertices are incident to the partition edges. This decomposition, illustrated in Fig. 2.2a), is not very useful for the conjugated graph representation since the corresponding conjugated edges may belong to different subgraphs. However, edge decomposition may have some advantages over nodal decomposition in topological analysis of signal-flow graphs [230].

Fig. 2.2. a) edge decomposition, b) mixed decomposition

3. Mixed decomposition.

Mixed decomposition, shown in Fig. 2.2 b), is a combination of nodal and edge decompositions and may benefit from their particular advantages. Block vertices in this partition are the nodes incident to two or more blocks and the vertices incident to the partition edges. A particular case of the graph decomposition is bisection (partition into two subgraphs). Bisection plays an important role in topological analysis, as an arbitrary partition can be presented as a sequence of bisections. In addition, it can be shown that this is an optimum partition in the hierarchical analysis. Data structure, and memory organization is particularly simple for the sequence of bisections.

In topological analysis a graph G partitioned into subgraphs G_i is represented using a substitute graphs defined as follows:

Definition 2.1. A substitute graph G_i^s of a subgraph G_i is a complete graph 65 spanned over all block nodes incident to this subgraph. Examples of substitute graphs are shown in Fig. 2.3.

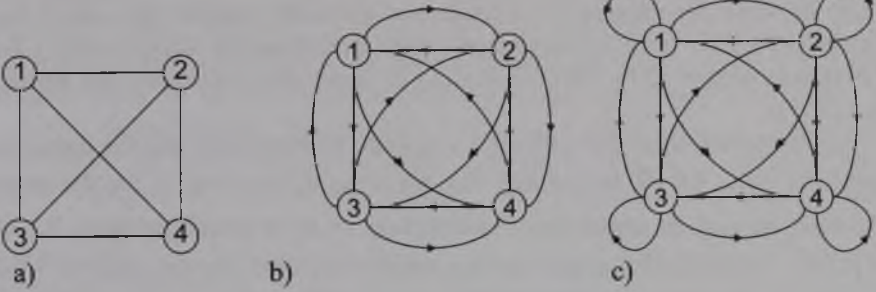


Fig. 2.3. Examples of substitute graphs for
a) flow-graph, b) directed graph, c) conjugated graphs

In a similar way, a substitute graph of the decomposed graph G is a graph composed of the substitute graphs of all its subgraphs. The substitute graph cannot be too complex as the complexity of its analysis quickly increases with the graph size. Thus, there is a need to limit the number of blocks and block nodes, which limits the size of networks that can be analyzed using **direct decomposition**. To alleviate this limitation, **hierarchical decomposition** was introduced in [223], [230] and used in topological analysis of electronic circuits using signal-flow graph and directed graph respectively. Hierarchical decomposition is illustrated by a **decomposition tree**. Vertices of the decomposition tree correspond to subgraphs of hierarchical decomposition. If graph G_k was obtained by decomposing graph G_j , then the decomposition tree has an edge directed from the vertex G_j to the vertex G_k .

2.2.2. Algorithms of Graph Decomposition

The main role of graph decomposition is to partition the network graph and to present it in a convenient for hierarchical analysis form using the decomposition tree. It is highly desirable to perform graph decomposition automatically. There are several

reasons for this. The main reason is that the graph structure is not known before the circuit data is entered into the computer program. Quite often the circuit netlist is automatically extracted from its layout by the extraction program, provided by synthesis tools, or generated from hardware description languages. Partitioning the network graph based on the subcircuit structure could lead to a suboptimum decomposition. Entering decomposed graph organization would be cumbersome and would require user's familiarity with decomposition methods. Finally, determining if the graph partition is beneficial to analysis would require complex calculations.

Graph decomposition methods use several approaches as follows:

1. Clique finding [75], [114].
2. Vertex swapping [122].
3. Solving associated equations [95].
4. Subgraph contour finding [180], [204].

Graph decomposition is NP (Non-deterministic Polynomial) complete [67], thus it is hard to expect that a fast algorithm for optimum graph decomposition can be found. Only some of these decomposition methods had efficient heuristic algorithms that find suboptimum partitions. One of these algorithms, developed by Sangiovanni, Chen and Chua [204] was improved by Starzyk leading to faster decomposition and better graph partitions [225].

This improved algorithm creates a sequence of bisections and the corresponding decomposition tree. Let G_k be a graph that corresponds to a leaf of the decomposition tree. At each step of the algorithm, the bisection of G_k is found based on the contour method [180]. If the divided graph requires smaller effort to analyze than the undivided one, then the graph G_k is subdivided into two graphs G_l and G_r, and the decomposition tree is appended with two new leafs that correspond to these two subgraphs. Details of this algorithm are presented in [225] with discussion of its application to topological analysis presented in [226]. Using the developed approach, graph decomposition can be efficiently automated with heuristic methods that yield near optimum solutions. Decomposition methods are independent of the type of the graph representation used.

3. Topological Analysis

3.1. Foundations of Topological Analysis

Topological analysis uses network topology to determine transfer functions, sensitivities and other circuit properties, yielding results in a symbolic form. For a number of years, efficient implementation of topological analysis was a subject of intense research [2], [196], [181], [178], [120], [124], [170], [109], [168], [125]. Early attempts to improve the efficiency of topological analysis through graph reduction [43], [70], [29] and decomposition [93], [49], [214] did not yield efficient computer programs and were critically evaluated by Alderson and Lin [5].

In spite of this relatively modest progress, research on implementable topological methods continued [276]. Chen used bi-section of a graph to improve analysis of directed graphs [49]. This was generalized by Konczykowska and Starzyk to a case of direct graph decomposition using structural numbers [134], leading to an efficient method and computer program for topological analysis of conjugated graphs [135],[136].

Direct decomposition significantly improved analysis time for small networks (10-50 nodes); however larger networks still could not be efficiently analyzed. Significant progress in topological analysis was made only after introduction of hierarchical decomposition methods. Starzyk introduced hierarchical analysis of signal-flow graph [230], and a similar approach was developed by Starzyk and Sliwa for directed graphs [223],[224]. Using these methods computer programs for analysis of large linear circuits were developed [37], [133], and [132]. Further progress was accomplished by means of upwards hierarchical analysis [40] that reused the same lower order terms in hierarchical analysis to improve the processing speed several times over downwards hierarchical analysis [39].

Topological analysis provides symbolic expressions for the characteristic functions of an analog circuit. It complements numerical simulation in an essential way, providing the analog circuit designer with more control over the design and testing process. Numerical simulation provides numerical results, but gives no indication where these numbers come from and how they change if parameter values are altered. In numerical analysis there is no indication how to improve the circuit performance or modify its parameters if the design specifications are not met. Multiple simulations are required at various parameter values to obtain this kind of information. And yet the results may be difficult to synthesize into a coherent and clear relation to parameter values. Optimization programs need to be employed to find better operation points.

Alternatively, by observing results in a symbolic form, much richer information about the analyzed circuit can be extracted. A symbolic simulator provides analytic formulas directly and for more complex circuits than would be possible by using hand analysis. No longer do circuit designers have to struggle to discover often complex associations between numerical results and the parameter changes that caused them. The symbolic results can be used repeatedly with varying circuit parameters, giving a different and clearer view of the circuit analysis. Properly conducted symbolic analysis

helps to reduce the analog circuit design time and reduce the design cost, complementing numerical analysis and expanding the spectrum of tools a designer may use.

The first part of this work discusses the major results of the author's work on topological analysis of large electronic networks and points to the renewed interest in symbolic methods that followed. All three major topological representations are discussed. All three major graph representations were studied to provide a uniform treatment for all topological approaches, as well as to compare them on the same grounds. The study was driven by the incommensurability of treatments of various topological representations found in the literature.

All three representations were applied to analyze an electronic circuit described by the system of linear equations:

$$AX=W \tag{3.1}$$

where $X=[x_1, x_2, \ldots, x_n]^T$, and $W=[w_1, w_2, \ldots, w_n]^T$. Typically, these equations describe a lumped, stationary system that has been linearized around the operating point and X and W are Laplace transforms of time domain variables used in the network analysis.

Other forms of equations that describe analyzed systems can also be considered. For instance, a piecewise linear analysis introduced by Katzenelson [120] may uses linear equations of similar form and requires only a change of the right hand side and coefficient matrix values between iterations. In a similar way, (3.1) can be used in a time-domain analysis of nonlinear circuits that use companion models [187] for reactive components and Newton-Raphson iterations performed on a system of linear equations. The important assumption is that the system topology does not change between iterations, so that the same symbolic results can be reused.

Even the changes that appear to change circuit's topology, such as shorts and opens, can be handled by this approach, provided the changes are described by the ideal switch elements, and treated simply as changes in switch parameter values. In particular, such analysis can be extended to handle the switch-capacitor networks or catastrophic faults in the nominal circuit.

A direct objective of topological analysis is to obtain the transfer function of the analyzed network in the following form:

$$F(s)=\frac{N(s)}{D(s)}=\frac{\sum_{n}\prod_{i} y_i}{\sum_{d}\prod_{j} y_j} \tag{3.2}$$

where the numerator $N(s)$ and the denominator $D(s)$ are the sum of products of graph edge weights and are independent of the topological representation used.

It is easy to illustrate the relation between a selected topological representation and the coefficient matrix A. For instance, if A is a nodal admittance matrix described by (2.1), then Y is a diagonal matrix of component admittance values while P and Q are the incidence matrices of graphs that describe this network. For different types of graphs P and Q are as follows:

1) In conjugated graphs, P is the incidence matrix of the current graph, while Q is the incidence matrix of the voltage graph. P and Q play similar roles in norator-nullator graphs.
2) In directed graphs, P is the incidence matrix of the unistor graph P, while Q is obtained from P by replacing all -1 values with 0. In a similar fashion, Q is the incidence matrix of the dispersor graph, while P is obtained from Q by replacing all -1 values with 0.
3) In signal-flow graphs, P is obtained from the incidence matrix of Coates' graph by replacing all 1 values with 0 and changing all -1 to 1, while Q is obtained from the incidence matrix of Coates' graph by replacing all -1 values with 0.

F(s) can be obtained using Cramer's rule as a ratio of cofactors and determinant of the coefficient matrix T. Subsequently such cofactors and determinant are expressed using the Binet-Cauchy theorem 65.

Theorem 3.1: If Q and R are rectangular $k \times m$ and $m \times k$ matrices, respectively with $k \leq m$, then the determinant of their product can be determined as follows:

$$|QR| = \sum_{major\ cofactors} |Q_i||R_i| \tag{3.3}$$

where the major cofactors are selected from the corresponding columns and rows of the rectangular matrices Q and R.

It is also known that a major cofactor of the incidence matrix is nonzero only if it corresponds to a graph tree [256]. Thus, from the Binet-Cauchy theorem, the determinant of the coefficient matrix A can be expressed through one of the following methods (depending on the topological description used):

1) Sum of the weights of all connections in Coates' graph. A connection in Coates' graph is defined as a set of disjoint cycles that contains all the graph nodes. The connection weight is the product of weights of all the edges included in the connection.
2) Sum of the weights of all directed trees in the directed (unistor or dispersor) graph. A directed tree has only a single edge outgoing from each node with the exception of the reference node. The weight of a directed tree is the product of all the edges included in the tree.
3) Sum of the weights of all complete trees in the conjugated (current-voltage or norator-nullator) graph. A complete tree is a tree in both conjugated graphs. The weight of a complete tree is the product of all the edges included in the tree.

Thus a topological analysis requires finding all the trees or all the connections in a network graph. Computational effort is proportional to this number of trees or connections and depends on the type of topological representation used.

For the signal-flow graph the number of required connections can be estimated using the adjacency matrix $D(G)=[d_{ij}]$ of Coates' graph G, where $d_{ij} = m$ if and only if there are m edges directed from node i to node j. The number of connections of the graph G is equal to the permanent of the matrix $D(G)$ [48]. A permanent of a matrix $A=[a_{ij}]$ is computed in a similar way to the matrix determinant, as a sum of products over all permutations of elements from different rows

$$perA = \sum_J a_{1j_1} a_{1j_2} \ldots a_{1j_n}, \tag{3.4}$$

where $J = (j_1, j_2, \ldots, j_n)$ is a permutation of 1,2,…,n. As shown in [230] the permanent of D(G) can be estimated as follows:

$$per\ D(G) \le \left(\frac{k+1}{n} - 1\right)^n, \tag{3.5}$$

where n is the number of graph nodes and k is the number of graph edges.

In a similar way the number of trees of a directed graph can be estimated from its adjacency matrix $D_I(G)=[d_{ij}]$, where $d_{ij} = m$ if and only if there are m edges directed from node i to j for $i \neq j$, and $d_{ii} = m$ if there are m edges directed away from node i. (Here we assume that the reference node was removed from the directed graph). The number of trees directed towards the reference node can be estimated as follows:

$$\bar{\bar{T}} = \det[D_I(G)] \le \frac{1}{n}\left(\frac{k}{n-1}\right)^{n-1}. \tag{3.6}$$

Finally, the number of complete trees of the conjugated graphs can be estimated in a similar way and it is related to determinants of their corresponding adjacency matrices as discussed in [226]. From these results it is obvious that a direct topological analysis requires computational effort that grows very fast with the circuit size. These estimates are confirmed in practical analyses where the numbers of generated symbolic terms grow exponentially. This makes direct topological analysis impractical even for medium size circuits (with more than 30 nodes).

Topological formulas define the transfer functions of the analyzed circuit using connections or trees depending on the graph representation used. Basic functions can be determined by using cofactors of the indefinite admittance matrix. For instance, a two-port voltage gain can be determined as follows:

$$K_{vv} = \left.\frac{v_0}{v_i}\right|_{i_0=0} = \frac{Y_{rp,sq}}{Y_{rr,ss}}. \tag{3.7}$$

See 45 for formulas of other network functions. Since transfer functions can be expressed through characteristic polynomials $n_{oo}, n_{os}, n_{so}, n_{ss}$ and m [185], it is sufficient to express the topological formulas' relationships to characteristic polynomials and to cofactors of the system coefficient matrix using

$$\begin{aligned} n_{oo} &= Y_{uv}, \\ n_{os} &= Y_{pp,qq}, \\ n_{so} &= Y_{rr,ss}, \end{aligned} \tag{3.8}$$

$$n_{ss} = Y_{rr,ss,pp} - Y_{rr,ss,pq} - Y_{rr,ss,qp} + Y_{rr,ss,qq},$$
$$m = Y_{rp,sq}.$$

where $Y_{rp,sq}$ denotes a cofactor of the matrix Y obtained by removing rows r and s and columns p and q. Other cofactors have similar interpretations.

The role of topological analysis is to obtain symbolic formulas for these cofactors using graphs and their topological properties. The following sections discuss topological analysis methods for different graph representations of electronic circuits.

3.2. Signal-Flow Graph Analysis of Electronic Circuits

Since signal-flow graph topological analyses using Coates' and Mason's graphs are very similar, we will illustrate this type of analysis using Coates' graph. Coates' graph of the coefficient matrix A is denoted as $G_C(A)$. So, in general, it may describe not just an electronic circuit but any linear system (3.1). An edge in Coates' graph that corresponds to a coefficient $a_{ij} \in A$ is directed from node x_j to node x_i and has the weight equal to a_{ij}.

Example 3.1. Consider the following equation

$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 0 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 0 \\ 9 \end{bmatrix}. \tag{3.9}$$

Coates' graph of the coefficient matrix associated with this equation is shown in Fig. 3.1.

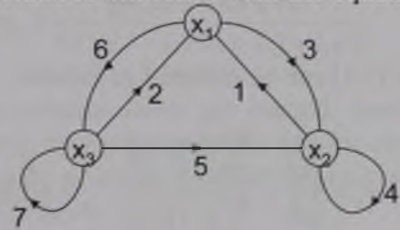


Fig. 3.1. Coates' graph of the coefficient matrix

Theorem 3.2 [45]. The determinant of the coefficient matrix A can be computed based on its Coates' graph $G_C(A)$ as follows:

$$|A| = (-1)^n \sum_{c \in C} (-1)^{l_c} |c|, \tag{3.10}$$

where C is the set of all connections in $G_C(A)$

$$|c| = \prod_{e \in c} y_e, \tag{3.11}$$

y_e is the edge weight, and l_c is the number of cycles in the connection c.

Solving (3.1) requires finding the determinant of the augmented coefficient matrix

$$A_a = \begin{vmatrix} d & L \\ -W & A \end{vmatrix}, \tag{3.12}$$

where d is a symbol for all terms in the denominator, and $L=[l_1, l_2, ..., l_n]$ contains symbols for all terms in numerators of variables $x_1, x_2, ..., x_n$, respectively. Matrix Coates' graph associated with A_a is shown in Fig. 3.2

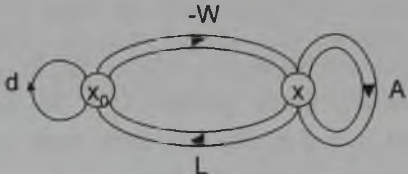


Fig. 3.2. Matrix Coates' graph of the system of equations

Vertex x_0 is identical for all independent sources. The following lemma is used to solve equation (3.1) topologically

Lemma 3.1 [226]. The determinant of the augmented matrix A_a has the following expansion

$$|A_a| = d\,|A| + \sum_{i=1}^{n} l_i\,|A_i|, \tag{3.13}$$

and the solution vector in (3.1) can be obtained as follows:

$$x_i = \frac{|A_i|}{|A|} = \frac{\sum_{c \in C_i} (-1)^{l_c} |c|}{\sum_{c \in C} (-1)^{l_c} |c|} \qquad i = 1,2,...,n, \tag{3.14}$$

where C and C_i are the connection sets of $G_C(A_a)$ that contain edges d and l_i respectively.

Example 3.2. Fig. 3.3 shows Coates' graph $G_C(A_a)$ of the augmented matrix A_a for the system of linear equations (3.9).

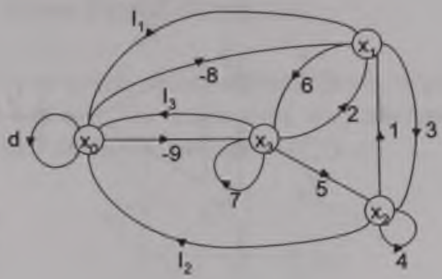


Fig. 3.3. Coates' graph of the augmented matrix A_a

The augmented matrix A_a is as follows:

$$A_a = \begin{bmatrix} d & l_1 & l_2 & l_3 \\ -8 & 0 & 1 & 2 \\ 0 & 3 & 4 & 5 \\ -9 & 6 & 0 & 7 \end{bmatrix}. \tag{3.15}$$

Using Theorem 3.2 we can find the determinant of matrix A_a

$$\begin{aligned} |A_a| &= -d \bullet 1 \bullet 3 \bullet 7 + d \bullet 1 \bullet 5 \bullet 6 - d \bullet 2 \bullet 4 \bullet 6 + l_1 \bullet 8 \bullet 4 \bullet 7 + l_1 \bullet 1 \bullet 5 \bullet 9 \\ &- l_1 \bullet 2 \bullet 4 \bullet 9 - l_2 \bullet 8 \bullet 3 \bullet 7 + l_2 \bullet 8 \bullet 5 \bullet 6 + l_2 \bullet 2 \bullet 3 \bullet 9 - l_3 \bullet 1 \bullet 3 \bullet 9 - l_2 \bullet 8 \bullet 4 \bullet 6 \end{aligned} \tag{3.16}$$

By selecting the corresponding terms from this expansion and using (3.13), we can determine any variable in (3.14). For instance x_3 can be determined as

$$x_3 = \frac{1 \bullet 3 \bullet 9 + 8 \bullet 4 \bullet 6}{1 \bullet 3 \bullet 7 - 1 \bullet 5 \bullet 6 + 2 \bullet 4 \bullet 6} \tag{3.17}$$

If only some elements of the solution vector are needed, then only the corresponding symbols in the L vector are used, with other symbols and the corresponding edges in the Coates' graph removed. Typically, an output function depends on one or two elements of the solution vector and L is very sparse.

Definition 3.1. Coates' graph of a linear circuit described by the modified admittance matrix is obtained by combining transitor models [226] of all circuit elements (including sources) with additional edges d and l_1, l_2, ..., l_n and removing the reference node with its incident edges.

Example 3.3. Fig. 3.4 shows a circuit with an ideal op-amp. Its Coates' graph with added independent voltage source V_1 and a single symbolic link l_4 to evaluate the output voltage is shown in Fig. 3.5.

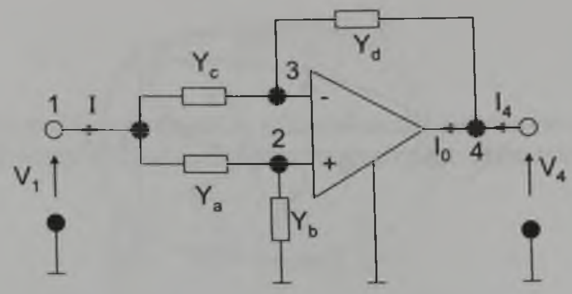


Fig. 3.4. A circuit with an ideal op-amp

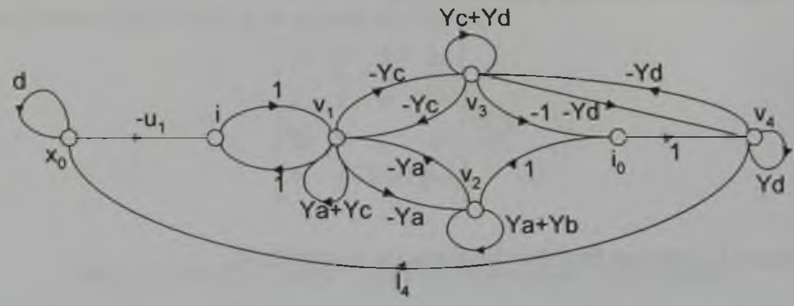


Fig. 3.5. Coates' graph of the circuit from Fig. 3.4

The determinant of the augmented matrix (3.12) is found using (3.13) and Theorem 3.2 as follows:

$$|A_a| = d\,|A| + \sum_{i=1}^{n} I_i\,|A_i| = (-1)^7[(-1)^4 d(Y_a+Y_b)(-1)(-Y_d) + (-1)^2 I_4(-V_1)(-Y_a)(Y_c+Y_d) + (-1)^2 I_4(-V_1)(-Y_c)(-1)(Y_a+Y_b)] \tag{3.18}$$

Using (3.14) we can get the output voltage as

$$V_4 = V_1 \frac{Y_a(Y_c+Y_d) - Y_c(Y_a+Y_b)}{Y_d(Y_a+Y_b)}, \tag{3.19}$$

and the voltage transfer function as

$$T = \left.\frac{V_4}{V_1}\right|_{I_4=0} = \frac{Y_aY_d - Y_cY_b}{Y_d(Y_a+Y_b)}. \tag{3.20}$$

Similar topological analysis can be conducted using Mason's graph. Mason's graph can be obtained if instead of (3.1) we will use $AX-W=X$.

3.2.1. Direct Signal-Flow Graph Analysis

The direct signal-flow graph circuit analysis can be obtained by using connections of the augmented graph $G_C(A_a)$. The augmented graph contains all the edges of graph $G_C(A_a)$ plus additional edges as shown in Fig. 3.6

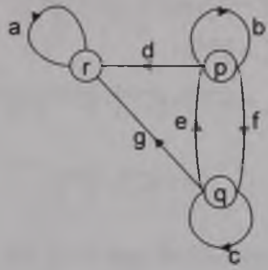


Fig. 3.6. Additional edges in the augmented graph $G_C(A_a)$

For analysis with decomposition we define multiconnections in the Coates' graph as follows. Let W denote the set of k pairs of the Coates' graph $G_c(V_c, E_c)$ vertices

$$W = \{(v_1, r_1), (v_2, r_2), \ldots, (v_k, r_k)\}, \quad v_i \neq v_j, v_i \neq r_j, r_i \neq r_j, \quad for \quad i \neq j. \tag{3.21}$$

Definition 3.2 [230]. A multiconnection of graph G_c is a subgraph c_W that contains k disjoint directed paths [65] and disjoint cycles. Each path must start at v_i and end at r_i for the corresponding pairs in W, and c_W must be incident to all the vertices V_c.

Example 3.4. Fig. 3.7 shows the 2-connection $c_W, W = \{(5,2),(1,3)\}$ of Coates' graph shown in Fig. 3.8

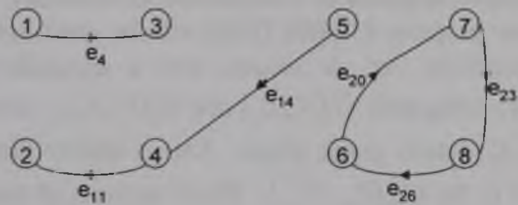


Fig. 3.7. 2-connection $c_W, W = \{(5,2),(1,3)\}$

0-connection has $W = \varnothing$ so it is denoted by c and is simply a connection, and if $v_i = r_i$ then a multiconnection has an isolated node v_i.

Definition 3.3 [230]. The weight of multiconnection set C_W of Coates' graph G_c with n nodes is defined as follows:

$$|C_W| = (-1)^n \sum_{c \in C_W} (-1)^{l_c} |c| \tag{3.22}$$

Lemma 3.2 [132]. Let $C^{a,b}$ denote the set of all connections of the augmented graph that contain edges a and b. Then the characteristic polynomials can be determined as follows:

$$\begin{aligned} n_{oo} &= |C|, \\ n_{os} &= |C^b| + |C^c| - |C^f| - |C^e|, \\ n_{so} &= |C^a|, \\ n_{ss} &= |C^{a,b}| + |C^{a,c}| - |C^{a,f}| - |C^{a,e}|, \end{aligned} \tag{3.23}$$

and

$$m = |C^d| - |C^g|. \tag{3.24}$$

Computer analysis programs SNAP and NASAP [56] were based on direct signal-flow graph analysis. They were limited to circuits with less than 30 nodes (in practice 10 nodes and 30 edges) due to the large number of generated symbolic terms required by (3.23) and (3.24).

3.2.2. Descending Hierarchical Signal-Flow Graph Analysis

A hierarchical signal-flow graph analysis uses hierarchical decomposition of its signal-flow graph. At each stage of the graph decomposition, the analysis is governed by using the results of the signal-flow graph analysis on the higher level. Substitute graphs on higher hierarchical levels have a larger effect on the analysis efficiency than the graphs on the lower levels, so it is desirable that they have a simple easy to analyze structure. This, in turn, requires near optimum decomposition of the underlying signal-flow graphs. This top-down approach is called the descending hierarchical analysis.

The basic idea of a hierarchical analysis can be explained on the case of a direct decomposition, which corresponds to a single level of hierarchy. An edge decomposition that was used in the program FANES [132] will be used here for illustration of this approach. For simplicity, let us assume that a signal-flow graph $G_1(V_1, E_1)$ was decomposed into two subgraphs $G_2(V_2, E_2)$ and $G_3(V_3, E_3)$, where V_1, V_2, V_3 denote graph vertices and E_1, E_2, E_3 denote graph edges. Let us denote the graph that contains the cutset edges E_{cut} of G_1 by $G_{cut}(V_{cut}, E_{cut})$. Block vertices of the substitute graph of the decomposed graph G_1 are the input-output nodes and the vertices V_{cut}.

Example 3.5. Fig. 3.8 shows an example of Coates' graph G_1 and the substitute graph G_1^d of its direct decomposition through the cutset with edges $E_{cut} = \{e_{13}, e_{14}, e_{15}\}$. G_i^s denotes the substitute graph of the subgraph G_i, $(i = 1,2)$.

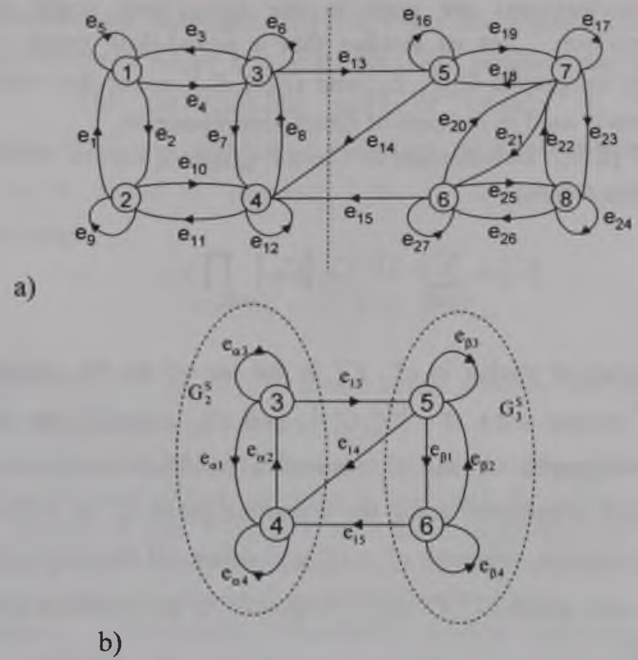


Fig. 3.8. a) Coates' graph G_1, b) the substitute graph G_1^d

Hierarchical analysis uses proper multiconnections in Coates' graph defined as follows.

Definition 3.4 [226]. A multiconnection c_W is **proper** w.r.t. the set of subgraphs $S = \{G_1^s, G_2^s, ..., G_m^s\}$, and is denoted c_W^P, if each subgraph $c_w^P \cap G_i^s$ has no path or cycle of the length greater than one.

Example 3.6. Fig. 3.9 shows examples of proper and improper connections of the graph G_1^d shown in Fig. 3.8 b) w.r.t. $S = \{G_2^s, G_3^s\}$.

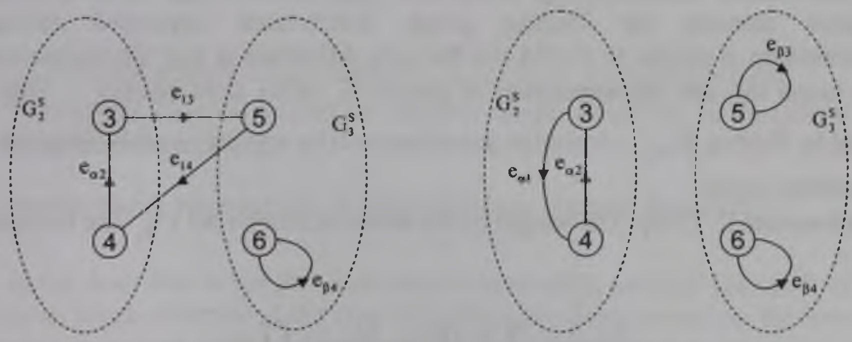


Fig. 3.9. Proper and improper connections of the graph G_1^d

Proper multiconnections are used in the signal-flow graph analysis with hierarchical decomposition. Let us assume that a signal-flow graph $G_1(V_1, E_1)$ was decomposed into two subgraphs $G_2(V_2, E_2)$ and $G_3(V_3, E_3)$, using the cutset edges E_{cut}. The following theorem is used in the case of direct decomposition.

Theorem 3.3 [230]. Determinant of Coates' graph G_1 can be obtained using the direct decomposition as follows:

$$|C_1| = \sum_{c^p \in C_1^p} (-1)^{l_p} |C_{W_2}||C_{W_3}| \prod_{e \in E_{cut} \cap c^p} y_e, \tag{3.25}$$

where l_p is the number of cycles in c^p, C_1^p is the set of all the connections of the substitute graph G_1^d proper w.r.t. $S = \{G_2^s, G_3^s,\}$, and C_{W_2} and C_{W_3} are the sets of k-connections in the subgraphs G_2 and G_3 respectively. Their corresponding pairs of vertices W_2 and W_3 are determined using the substitute graph G_i^s as follows. For each substitute graph G_i^s compute subgraph $c^p \cap G_i^s$ and delete all the loops in the resulting subgraph defining a new graph $G_i^p(V_i^p, E_i^p)$. Edges E_i^p of the obtained graph G_i^p define pairs of vertices in W_i.

Using the direct decomposition to topological analysis significantly increases its efficiency. The estimated time of analysis is reduced R_d times where

$$R_d \approx \left(\frac{k+1}{n} - 1\right)^{\frac{n}{2}} \tau_0, \tag{3.26}$$

where τ_0 depends on the complexity of the cutset graph G_{cut}, k and n are the number of edges and vertices of the original signal-flow graph G_1, respectively.

Hierarchical analysis brings even more significant savings of the analysis time. Topological formula for finding graph determinant expressed through its multiconnections is similar to (3.25) and the only difference is that the multiconnections C_{W_i} are found through decomposition of graphs G_i rather than directly. This can be illustrated by finding $|C_{W_2}|$. A similar procedure will be applied to other subgraphs on all decomposition levels.

Theorem 3. [230]. The weight of the multiconnection set C_{W_2} can be obtained as follows:

$$|C_{W_2}| = \sum_{c^p \in C_{W_2}^p} (-1)^{l_p} |C_{W_4}||C_{W_5}| \prod_{e \in E_{cut2} \cap c^p} y_e, \tag{3.27}$$

where it is assumed that subgraph $G_2(V_2, E_2)$ was decomposed into two subgraphs G_4 and G_5, using the cutset edges E_{cut2} with the entire notation similar to the one used in Theorem 3.3. The corresponding pairs of vertices W_i *(i=4,5)* that define multiconnections in G_i are determined using the substitute graph G_i^s and following a similar process as stated in Theorem 3.3. $C_{W_2}^p$ are the multiconnections of the decomposition graph

$$G_2^d = G_4^s \oplus G_5^s \oplus G_{cut2} \tag{3.28}$$

that are proper w.r.t. $S = \{G_4^s, G_5^s,\}$. Block nodes used in this analysis contain block nodes from the higher level and vertices V_{cut2}.

Theorem 3.4 can be used to further decompose subgraphs on the lower levels if they are still too big and if their partition is beneficial to reduce the analysis time. A characteristic feature of this method is that multiconnections on the lower level are obtained by using proper multiconnections from the higher level. This feature justifies the name of such approach - descending hierarchical analysis.

Comment. Descending hierarchical analysis formulas were obtained in Theorems 3.3 and 3.4 for the case of bisection using the edge decomposition. Based on Theorems 3.3 and 3.4 it is easy to obtain results for other types of decompositions. For instance, in the nodal decomposition $E_{cut2} = \varnothing$ and (3.27) reduces to

$$|C_{W_2}| = \sum_{c^p \in C_{W_2}^p} (-1)^{l_p} |C_{W_4}| |C_{W_5}|, \tag{3.29}$$

and if the graph is decomposed into more than two subgraphs then (3.27) is replaced by

$$|C_{W_2}| = \sum_{c^p \in C_{W_2}^p} (-1)^{l_p} \prod_{i \in I} |C_{W_i}| \prod_{e \in E_{cut2} \cap c^p} y_e, \tag{3.30}$$

where I is the set of indexes of all subgraphs of $G_2(V_2, E_2)$ decomposition.

Topological analysis based on Mason's graph is similar to described analysis based on Coates' graph [230].

3.2.3. Ascending Hierarchical Signal-Flow Graph Analysis

In the described in the previous section descending analysis the result of analysis at the higher levels determined the type of multiconnections needed on the lower levels. The ascending analysis reverses the direction of information flow in analyzing subgraphs of the decomposition tree. In this analysis, the results of analysis of the subgraphs on the lower levels are used to determine functions of the graphs on the higher levels.

Consequently, the ascending approach avoids repetitive analysis of the same subgraphs observable in the descending approach. This results in a significant reduction of the analysis time, on the expense of the increased memory requirement.

The main idea of the ascending analysis is to obtain and store all multiconnections that may be useful in the hierarchical analysis ahead of time, and use them when they are needed. In particular, in an upward signal-flow graph analysis, multiconnections of subgraphs on the lower level are connected together to create multiconnections of the graph on the higher level.

To make the analysis more efficient, multiconnections should be generated in groups and the whole groups of multiconnections should be combined together. To facilitate this task a wider set of multiconnections is defined.

Definition 3.5 [221]. $C(B,E)$ is a set of multiconnections in which B represents initial nodes and E represents terminal nodes of multiconnections paths, where $B=\{b_1,b_2,...,b_m\}$, $E=\{e_1,e_2,...,e_m\}$, and $B\cup E\subset NB$ - the set of block nodes.

Notice that this definition does not specify pairs of B and E nodes – just the sets. Thus a single $C(B,E)$ set contains many C_W sets, were W contains pairs of vertices B and E in various combinations. This grouping of multiconnection types simplifies the hierarchical analysis and improves its efficiency.

Definition 3.6 [221]. The sign of a multiconnection $c\in C(B,E)$ is defined as

$$sign(c)=(-1)^{n+k+l_p}\, ord(B)\, ord(E) \qquad (3.31)$$

where n is the number of graph nodes, k is the *card(B)*, l_p is the number of loops in c, and

$$ord(x_1,x_2,...,x_m)=\begin{cases}1, & \textit{when the number of permutations} \\ & \textit{ordering the set are even} \\ -1, & \textit{otherwise}\end{cases} \qquad (3.32)$$

The analysis on a higher level of decomposition consists of evaluation of multiconnections of a subgraph that results from the association of two (or more) subgraphs on the lower level. Let us denote the sets of block nodes for both subgraphs and the resulting subgraph by NB_1, NB_2, and NB, respectively. When the two subgraphs are connected, some of their block nodes become internal nodes, and no other subgraphs are connected to these nodes on the higher levels. These nodes are called reducible nodes.

Let us denote $COM=NB_1\cap NB_2$, the set of common nodes; $RED = COM - NB$, the set of reducible nodes; $C_1(B_1,E_1)$, $C_2(B_2,E_2)$, and $C(B,E)$ - sets of multiconnections (as defined in Definition 3.5) for both subgraphs and the resulting subgraph, respectively. The following theorem describes how these sets of multiconnections are related.

Theorem 3.5 [221]. Any set of multiconnections $C(B, E)$ can be obtained according to the following rule:

$$C(B,E)=\bigcup C_1(B_1,E_1)\times C_2(B_2,E_2), \qquad (3.33)$$

where the summation is performed over all sets of multiconnections $C_1(B_1, E_1)$ and $C_2(B_2, E_2)$ for which $B_1 \cap B_2 = \varnothing$, $E_1 \cap E_2 = \varnothing$, $RED = (B_1 \cup B_2) \cap (E_1 \cup E_2)$ and × is a Cartesian product 45 of sets $C_1(B_1, E_1)$ and $C_2(B_2, E_2)$. Sets B and E are obtained as $B = (B_1 \cup B_2) - RED$, $E = (E_1 \cup E_2) - RED$.

If all element weights are different, there are no duplicate terms in the formula (3.33). The sign of multiconnection $c \in C(B, E)$ can be calculated as follows:

$$sign(c) = sign(c_1)\, sign(c_2)(-1)^k\, \Delta \tag{3.34}$$

where

$c = c_1 \cup c_2,\quad c_1 \in C_1(B_1, E_1),\quad c_2 \in C_2(B_2, E_2),$

$k = \min(card(E_1 \cap B_2 \cap COM), card(E_2 \cap B_1 \cap COM)) + card(COM),$

$\Delta = ord(B_1)\, ord(E_1)\, ord(B_2)\, ord(E_2).$

Comment. By using Theorem 3.5 all the multiconnections *C(B, E)* can be obtained by combining whole groups of multiconnections from the lower level. In addition, by using (3.34) the sign modification that results from joining together multiconnections on the lower level is constant for the whole group of connections $C_1(B_1, E_1) \times C_2(B_2, E_2)$, as $(-1)^k \Delta$ depend only on the sets B_1, E_1, B_2, E_2. These features greatly simplify computer realization of hierarchical analysis because we do not have to deal with each multiconnection separately.

3.3. Directed Graph Analysis of Electronic Circuits

A linear system can be equally easily represented by a directed graph and by a signal-flow graph. However, the signal-flow graphs are typically used to represent an arbitrary linear system, while directed graphs are almost exclusively used to represent electronic circuits. This difference results from higher efficiency of representing an electronic circuit by a directed graph rather than by a signal-flow graph. Models of two-terminal elements (*R,L*, and *C*) are simpler using directed graph representation, resulting in more efficient analysis. On the other hand, signal-flow graph's structure in a natural way represents an arbitrary linear system and the resulting graphs are simpler in systems with asymmetrical coefficient matrices.

Theorem 3.6 [226]. The determinant of the coefficient matrix A can be determined based on its unistor graph $G_U(A)$ as follows:

$$|A| = \sum_{t \in T} |t|, \tag{3.35}$$

where T is a set of all trees of $G_U(A)$ directed to the reference node

$$|t| = \prod_{e \in t} y_e \,, \tag{3.36}$$

y_e is the edge weight. The reference node is a selected node in the unistor graph associated with the coefficient matrix *A* as discussed in [226].

Similar to signal-flow graph analysis, the unistor graph $G_U(A)$ can be appended with the symbolic excitation and the circuit response edges. Fig. 3.10 shows the resulting graph in the case of a single excitation and a single response.

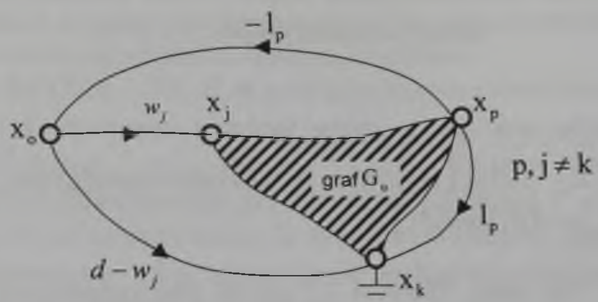


Fig. 3.10. Unistor graph of the system of equations

The following lemma is used to solve equation (3.1) topologically

Lemma 3.3 [226]. The determinant of the augmented matrix A_a has the following expansion

$$|A_a| = d\,|A| + \sum_{i=1}^{n} l_i\,|A_i|, \tag{3.37}$$

and the solution vector in (3.1) can be obtained as follows:

$$x_i = \frac{|A_i|}{|A|} = \frac{\sum_{j=1}^{n} w_j \sum_{t \in T_{ij,k}} |t|}{\sum_{t \in T} |t|} \qquad i = 1,2,\ldots,n, \tag{3.38}$$

where *T* is a set of all trees of $G_U(A)$ directed to the reference node k, and $T_{ij,k}$ is a set of two-trees with the reference nodes *k* and *i*, and node *j* in the same subtree as node *i* (for details see [226]).

There is no significant difference between topological analysis of the unistor and dispersor graphs, except that the coefficient matrix associated with a dispersor graph equals to the transposed matrix of the corresponding unistor graph. This results in symmetry of the topological formulas for circuit analysis using both representations.

3.3.1. Direct Directed Graph Analysis

The direct directed graph circuit analysis can be obtained by using the directed trees of the augmented unistor graph $G_U(A_a)$. The augmented unistor graph contains all the edges of graph $G_U(A_a)$ plus additional edges as shown on Fig. 3.11. Results for the analysis based on a dispersor graph representation are symmetrical to those obtained from the unistor graphs as presented in [226]

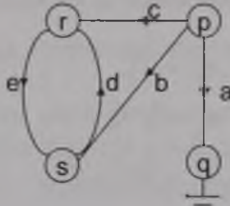


Fig. 3.11. Additional edges in the augmented unistor graph $G_U(A_a)$

For analysis with decomposition we define multitrees in the unistor graph as follows. Let V denote the family of k sets of unistor graph $G_u(V_u, E_u)$ vertices

$$V = \{(r_1, v_{11}, \ldots, v_{1m_1}), \ldots, (r_k, v_{k1}, \ldots, v_{km_k})\}, \tag{3.39}$$

Definition 3.7. A directed multitree t_V of graph $G_u(V_u, E_u)$ is a subset of its edges E_u that contains all vertices V_u and no cycles, such that each set of the family V is in different subtree and has r_i as its reference node.

Example 3.7. Fig. 3.12 shows the 3-tree $t_V, V = \{(1,2,3),(4),(5,6)\}$ of a unistor graph with 8 vertices.

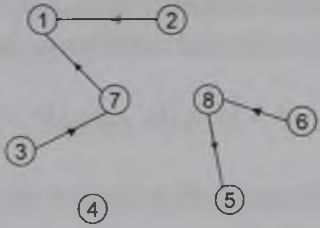


Fig. 3.12. 3-tree $t_V, V = \{(1,2,3),(4),(5,6)\}$

Definition 3.8 [226]. The weight of multitree set T_V of unistor graph G_u with n nodes is defined as follows:

$$|T_V| = \sum_{t_V \in T_V} |t_V| \tag{3.40}$$

where

$$|t| = \prod_{e \in t} y_e , \tag{3.41}$$

y_e is the edge weight.

Lemma 3.4. Let $T^{a,b}$ denote the set of all trees of the augmented unistor graph that contain edges a and b, and are directed to the reference node q. Then the characteristic polynomials can be determined as follows:

$$\begin{aligned} n_{oo} &= |T|, \\ n_{os} &= |T^a|, \\ n_{so} &= |T^d| + |T^e|, \\ n_{ss} &= |T^{a,d}| + |T^{a,e}|, \\ m &= |T^b| - |T^c|. \end{aligned} \tag{3.42}$$

Computer analysis program DISTOR [217] was based on the direct analysis of the directed graph of an analog circuit. This program obtains full symbolic analysis of the analyzed circuit and has similar limitations as programs based on direct signal-flow graph analysis.

3.3.2. Descending Hierarchical Directed Graph Analysis

The hierarchical analysis of a direct graph implements topological formulas with hierarchical decomposition of its directed graph. As in the case of the signal-flow graph representation, the basic idea of the hierarchical directed graph analysis is explained on the case of direct decomposition, which corresponds to a single level of hierarchy.

Definition 3.9 [226]. A directed multitree t_D defined by the set of paths

$$P = \{(v_1, r_1), (v_2, r_2), \dots, (v_p, r_p)\}, \tag{3.43}$$

is a multitree that contains directed path defined by the pairs of vertices $(v_i, r_i) \in P$. Each pair (v_i, r_i) defines a path from v_i to r_i.

Example 3.8. 3-tree t_V, $V = \{(1,2,3),(4),(5,6)\}$ from Fig. 3.12 is also an example of 3-tree t_P, $P = \{(2,1),(3,1),(4,4),(6,5)\}$.

Definition 3.10 [226]. A multitree t is **proper** w.r.t. the set of subgraphs $S = \{G_1, G_2, \dots, G_m\}$, and is denoted t^P, if each subgraph $t^P \cap G_i$ has no path of the length greater than one.

Example 3.9. Fig. 3.13 shows examples of proper and improper trees w.r.t $S = \{G_L, G_R\}$.

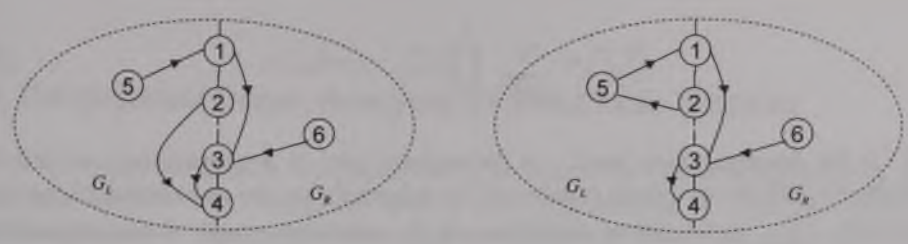


Fig. 3.13. Proper and improper trees

Proper multiconnections are used in directed graph analysis with hierarchical decomposition. Let us assume that a directed graph $G_1(V_1, E_1)$ was decomposed into two subgraphs $G_2(V_2, E_2)$ and $G_3(V_3, E_3)$. The following theorem is used in the case of direct decomposition.

Theorem 3.7 [223]. The weight of the multitree set T_P of the unistor graph G_1 can be obtained using the direct decomposition as follows:

$$|T_P| = \sum_{t \in T_P} |t| = \sum_{t_j \in T_P^d} \prod_{j=2}^{h} |T_{P_i}|, \tag{3.44}$$

where

$$|T_{P_i}| = \sum_{t \in T_{P_i}} |t|, \tag{3.45}$$

$T_{P_i}^d$ is a set of multitrees of the substitute graph G_1^d of direct decomposition, composed of the set of substitute graphs G_i^s, i=2,3,...,h to which the original graph G_1 was divided. The set of paths P_i is defined by the edges of the subgraph $G_P^i(V_P^i, E_P^i)$ defined as

$$G_P^i = t \cap G_i^s \tag{3.46}$$

with t taken from the set of multitrees T_P^d of the decomposition graph.

By combining multitrees in the substitute graph defined on the set of paths with those defined on the set of vertices, further improvement in directed graph analysis by direct decomposition is observed as reported in [226] (Theorem 5.4).

Hierarchical analysis brings even more significant savings of analysis time. Topological formula for finding graph determinant expressed through its multitrees is similar to (3.44) with this exception that the multitrees T_{P_i} are found through decomposition of graphs G_i rather than directly.

Theorem 3.8 [226]. The weight of the set of multitrees on *(l-1)*st decomposition level T_P^{l-1} can be obtained as follows:

$$\left|T_P^{l-1}\right| = \sum_{t_j \in T^d_{P(l-1)}} \prod_{i=m_l}^{h_l} \left|T^l_{P_i}\right|, \quad l = 1,\ldots,r, \tag{3.47}$$

where l is the decomposition level, r is the highest level of decomposition, m_l and h_l are the minimum and the maximum number of the subgraphs on the decomposition on level l respectively, $T^d_{P(l-1)}$ is the set of multitrees of the substitute graph of decomposition on $(l-1)$-th decomposition level, $T^l_{P_i}$ is the set of multitrees of i-th decomposition subgraph with the set of path P_i defined on l-th level of decomposition.

The set of path P_i in Theorem 3.8 is determined as in Theorem 3.7 with this exception, that the set of path P needed to determine T^0_P are given.

Comment. If a direct graph analysis is based on the edge decomposition with the cutset edges on the level l equal to E^l_{cut}, then Theorem 3.8 is modified as follows:

$$\left|T_P^{l-1}\right| = \sum_{t_j \in T^d_{P(l-1)}} \prod_{i=m_l}^{h_l} \left|T^l_{P_i}\right| \prod_{e \in t_j \cap E^l_{cut}} y_e, \quad l = 1,\ldots,r, \tag{3.48}$$

where y_e is the edge weight, and where the product is taken over all the edges that belong to the intersection of the substitute graph tree t_j and E^l_{cut}.

3.3.3. Ascending Hierarchical Direct Graph Analysis

The ascending direct graph analysis is organized in a similar way as the ascending signal-flow graph analysis. First, all the multitrees of all the substitute subgraphs are determined and are used to find the multitrees of the decomposed graphs on the lower levels. **Proper multitrees** are used to reduce the number of various multitrees considered by this algorithm.

Assume that a substitute graph of decomposition on a certain hierarchy level G^d_i was used to find its proper multitree $t_j \in T^d_P$. A product term in (3.47) that corresponds to this multitree can be easily found once its set of paths P_i is found, since the sets of multitrees T_{P_i} were already found on the lower level of decomposition. Thus it is sufficient to save the memory address where this information is stored.

Theorem 3.9 [226]. The weight of the set of multitrees on *(l-1)*st decomposition level T^{l-1}_P can be obtained in the ascending hierarchical analysis as follows:

$$\left|T_P^{l-1}\right| = \sum_{t_j \in T^d_{P(l-1)}} \prod_{i=m_l}^{h_l} \left|T^l_{P_i}\right|, \quad l = r, r-1,\ldots,1, \tag{3.49}$$

In spite of almost identical form of (3.47) and (3.49), a computer implementation efficiency of the ascending hierarchical analysis is much higher that that of the descending hierarchical analysis. Similar result applies to other forms of the direct graph decomposition.

3.4. Conjugated Graph Analysis of Electronic Circuits

A conjugated graphs representation of an electronic circuit is particularly simple, since each edge of the conjugated graphs may represent up to four elements of the coefficient matrix. This results in a smaller number of edges and nodes than for the signal-flow or the directed graph representations [31], [171].

Theorem 3.10 [212]. The determinant of the coefficient matrix A can be determined based on the pair of its conjugated graphs $(G_1(A), G_2(A))$ as follows:

$$|A| = \sum_{t \in T} |t||P_t||Q_t|, \tag{3.50}$$

where T is a set of all complete trees of $(G_1(A), G_2(A))$, P_t and Q_t are submatrices of the incidence matrices of the two graphs corresponding to the complete tree t

$$|t| = \prod_{e \in t} y_e, \tag{3.51}$$

y_e is the edge weight.

As with earlier representations, the pair of conjugated graphs $(G_1(A), G_2(A)))$ can be appended with the symbolic excitation and response edges. Fig. 3.14 shows the resulting graph in the case of a single excitation and a single response.

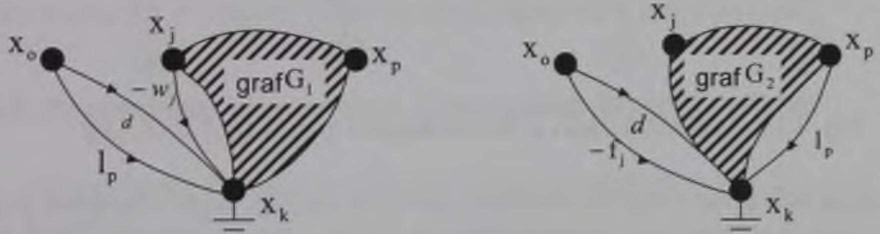


Fig. 3.14. A pair of the conjugated graphs of the system of equations

The following lemma is used to solve equation (3.1) topologically

Lemma 3.5 [226]. The determinant of the augmented matrix A_a has the following expansion

$$|A_a| = d\,|A| + \sum_{i=1}^{n} l_i\,|A_i|, \tag{3.52}$$

and the solution vector in (3.1) can be obtained as follows:

$$x_i = \frac{|A_i|}{|A|} = \frac{\sum_{j=1}^{n} w_j \sum_{t \in T_{ij,k}} |t|}{\sum_{t \in T} |t|} \qquad i = 1,2,\ldots,n, \qquad (3.53)$$

where T is a set of all complete trees of $(G_1(A), G_2(A))$ with the reference node k, and $T_{ij,k}$ is a set of complete two-trees with node j in the same subtree as node i (for details see 226).

Fundamental difficulties for using the conjugated graphs in circuit analysis stem from the requirement of simultaneous analysis of two different graphs. Thus, in spite of having a simpler graph representation, conjugated graphs do not yield as transparent and efficient topological analysis as the other two representations.

3.4.1. Direct Conjugated Graph Analysis

The direct conjugated graph circuit analysis can be obtained by using the complete trees of the augmented conjugated graphs $(G_1(A_a), G_2(A_a))$. The augmented graph $G_i(A_a)$ contains all the edges of graph $G_i(A)$ plus additional edges as shown on Fig. 3.15.

Fig. 3.15. Additional edges in the conjugated graphs $(G_1(A), G_2(A))$

Let us define the complete multitrees and their weights in the conjugated graphs for the purpose of analysis with decomposition, as follows. Let V denote the family of k sets of vertices of a conjugated graph $G(V, E)$ (Nullator, norator, current or voltage graph)

$$V = \{(v_{11}, \ldots, v_{1m_1}), \ldots, (v_{k1}, \ldots, v_{km_k})\}, \qquad (3.54)$$

Definition 3.11. A set of complete multitrees T_{V_1,V_2} of the conjugated graphs (G_1, G_2) is defined as $T_{V_1,V_2} = T_{V_1} \cap T_{V_2}$, and its weight is obtained as follows:

$$|T_{V_1,V_2}| = \sum_{t \in T_{V_1,V_2}} sign(t)|t| \qquad (3.55)$$

where

$$sign(t) = \det(P_t)\det(Q_t) \tag{3.56}$$

P_t and Q_t are the submatrices of the incidence matrices of the two graphs corresponding to the complete tree t.

Lemma 3.6. Let $T^{(a,b),(c,d)}$ denote the set of all complete trees of the augmented conjugated graphs obtained as follows:

$$T^{(a,b),(c,d)} = T_1^{(a,b)} \cap T_2^{(c,d)} \tag{3.57}$$

where $T_1^{(a,b)}$ is the set of all trees in $G_1(A_a)$ that contain edges a and b, and $T_2^{(c,d)}$ is the set of all trees in $G_2(A_a)$ that contain edges c and d. Then the characteristic polynomials can be determined as follows:

$$\begin{aligned} n_{oo} &= |T|, \\ n_{os} &= \left|T^{(b),(b)}\right|, \\ n_{so} &= \left|T^{(a),(a)}\right|, \\ n_{ss} &= \left|T^{(a,b),(a,b)}\right|, \\ m &= \left|T^{(a),(b)}\right|. \end{aligned} \tag{3.58}$$

Computer analysis programs PASTN [171] was based on direct conjugated graph analysis. This program obtains full symbolic analysis of the analyzed circuit and has similar limitations as programs based on direct signal-flow graph analysis.

3.4.2. Descending Hierarchical Conjugated Graph Analysis

The conjugated graph analysis with direct decomposition was first used in [135]. This analysis is conducted under the assumption that the pair of conjugated graphs (G_1, G_2) was divided onto subgraphs $G_{1i}(V_{1i}, E_{1i}), G_{2i}(V_{2i}, E_{2i}), i = 1,2,\dots,h$ under the following conditions:

$$1.\quad \bigcup_{i=1}^{h} E_{1i} = E_1, \quad E_{1j} \cap E_{1k} = \varnothing \quad for \quad j \neq k \tag{3.59}$$

$$2.\quad \bigcup_{i=1}^{h} V_{1i} = V_1, \tag{3.60}$$

$$3.\quad E_{1i} = E_{2i}, \quad V_{1i} = V_{2i}, \quad for \quad i = 1,2,\dots,h \tag{3.61}$$

Let T_V denote the set of multitrees of the graph G such that the vertices of different subsets of the family V are in different subtrees.

Definition 3.12. A multitree t is proper w.r.t. $S=\{G_1,G_2,...,G_m\}$, and is denoted t^P, if each subgraph $t^P \cap G_i$ contains only a star structure subtrees with the common vertices having the lowest index value in the corresponding subtrees.

Example 3.10. Fig. 3.16 shows examples of proper and improper trees w.r.t. $S=\{G_L,G_R\}$.

Fig. 3.16. Proper and improper trees

Let us consider the case of a direct nodal decomposition of the conjugated graphs (G_1,G_2). Assume that the conjugated graphs $(G_{1i},G_{2i})\, i=1,...,h$ are the decomposition graphs of (G_1,G_2). Let G^s_{mi} $(m=1,2)$ be the substitute graph of G_{mi}, and G^d_m be the substitute graph of the graph G_m decomposition, and $S_m=\{G^s_{m1},G^s_{m2},...,G^s_{mh}\}$. Let $T^P_{V_1,V_2}$, be the set of admissible pairs of multitrees of (G^d_1,G^d_2) such that if $(t_{V_1},t_{V_2})\in T^P_{V_1,V_2}$, then t_{V_m} $(m=1,2)$ is a proper multitree w.r.t. S_m and (t_{1i},t_{2i}) are admissible for all the subgraphs of decomposition $i=1,...,h$, where

$$t_{mi}=t_{V_m}\cap G^s_{mi} \quad m=1,2. \tag{3.62}$$

An **admissible** pair of multitrees of the conjugated graphs (G_1,G_2) is when they have the same number of subtrees. Using subtrees of t_{mi} $m=1,2$ let us define the family of sets of vertices

$$V_{mi}=\{(v_{11},...,v_{1m_1}),...,(v_{k1},...,v_{km_k})\}. \tag{3.63}$$

Theorem 3.11 226. The set of all complete multitrees T_{V_1,V_2} of the conjugated graphs (G_1,G_2) can be found using a direct decomposition as follows:

$$T_{V_1,V_2}=\bigcup_{(t_{V_1},t_{V_2})\in T^P_{V_1,V_2}}\prod_{i=1}^{h}T_{V_{i1},V_{2i}} \tag{3.64}$$

where $T_{V_{i1},V_{2i}}$ is the set of complete multitrees of the conjugates graphs (G_{1i},G_{2i}). The signs of complete multitrees $T_{V_{i1},V_{2i}}$ can be found using the method described in [136].

A similar theorem for a simple decomposition analysis of the conjugated graphs based on the block analysis and on the algebra of structural numbers was presented in [135]. Theorem 3.11 can be generalized to the hierarchical decomposition provided that conditions (3.59)-(3.61) are satisfied at each level of decomposition (see Theorem 5.7 in [226]). Finally, we can determine the weight of a set of complete multitrees as follows.

Theorem 3.12 [226]. The weight of a set of complete multitrees T_{V_1,V_2} in the hierarchical analysis of the conjugated graphs (G_1, G_2) can be determined using

$$T^{l-1}_{V_1,V_2} = \sum_{(t_{1_1},t_{1_2}) \in T^p_{V_1V_2}(l-1)} sign(t_1,t_2) \prod_{i=m_l}^{h_l} sign(t^1_i,t^2_i) \left| T^l_{V_{1i},V_{2i}} \right| \quad i = 1,\ldots,r \tag{3.65}$$

where l is the decomposition level, r is the highest level of decomposition, m_l and h_l are the minimum and the maximum number of the subgraphs on the level l decomposition respectively, $T^p_{V_1V_2}(l-1)$ is a set of the admissible pairs of multitrees of the substitute graphs of the graph decomposition on the l-th level of decomposition with the sets of vertices V_1 and V_2 determined on the $(l-1)$ -st level. $T^l_{V_{1i}V_{2i}}$ is the set of complete multitrees of i-th subgraph of decomposition with the set of vertices V_{1i} and V_{2i} determined on the l-th level. Signs of the complete multitrees $sign(t_1,t_2)$ and $sign(t^1_i,t^2_i)$ are computed as described in [226].

Hierarchical analysis using the conjugated graphs is much more complex than the analysis using other representations. It is a result of both restrictions imposed on the decomposition process, and more complex procedures to determine the complete multitrees and their signs. However, the major reason that this representation is inferior to representations based on signal-flow graphs or directed graphs is a significantly larger number of the admissible pairs of multitrees comparing to the number of either types of multiconnections or types of directed multitrees with the same number of the block nodes. Thus, for a given circuit topology, there are more types of subgraphs that need to be generated and verified in the conjugated graph analysis than in the case of other graph representations. This requires longer computing time and larger computer memories.

The ascending hierarchical analysis of the conjugated graphs can also be organized by generating all sets of admissible pairs of proper multitrees of all types. This analysis was not developed due to its lower efficiency than the ascending analysis of signal-flow graphs or directed graphs.

3.5. Algorithms for Topological Analysis Methods

Dedicated topological analysis algorithms are used to generate the sets of all paths, loops, multitrees, and multiconnections of various kinds depending on the selected graph representation. Most of the methods that were developed in the past (i.e. prior to author's research work) were limited to and targeted the direct analysis.

One of the most fundamental methods of generating trees in the graph is based on the elementary transformations [161], [183], [160], [162], [64]. In this method each new

tree is obtained by swapping one edge from the existing tree with one edge from the tree complement (the rest of the graph). However, this method is not the most efficient one. More effective are the recursive tree methods [176]. In these methods an edge is rejected from consideration if a closed loop is formed, otherwise it is maintained in the set of edges for further consideration by the recursive expansion algorithm [169], [165], [46], [53]. Other methods are based on finding the nonsingular major cofactors of the graph incident matrix [160], [47], [44]. Different approaches to generate trees and multitrees are presented in [26], [100], and [186]. Most of the algorithms developed for finding directed trees are derived from algorithms developed for finding trees in the linear graph [183], [211], and [88]. Large number of methods was devoted to generation of complete trees in conjugated graphs [44], [127], [160], [171], [228], [229], [232].

Multiconnections of a signal-flow graph can be reduced to generation of disjoint loops in the graph [106], [56]. Loop generation techniques first determine single loops of the graph and pairs of disjoint loops then, using these results, all the connections are generated. Such technique presented in [56] was used in the program SNAP [146]. Disadvantage of this method is a large memory required to store all the single and disjoint pairs of loops. Other methods of loop generation were presented in [89], [257], [275], and [6].

In this section developed by the author efficient algorithms for direct generation of all the multitrees and multiconnections are presented.

3.5.1. Multiconnections of a Signal-Flow Graph

All types of the multiconnections can be generated directly by using the algorithm based on the graph's structural matrix.

Definition 3.13. The structural matrix $M = [\hat{m}_{ij}]$ of a graph with n vertices is a square $n \times n$ matrix whose elements are the sets of edges that are directed from vertex i towards vertex j.

The set of connections of a flow-graph can be obtained using

$$C(M) = \bigcup_{(i_1,\dots,i_n)\in I} m_{1i_1} \times m_{2i_2} \times \dots \times m_{ni_n} \tag{3.66}$$

where I is the set of all the permutations of numbers $(i_1,\dots,i_n)$. There is no duplication in the formula (3.66). The sign of a connection $c \in m_{1i_1} \times m_{2i_2} \times \dots \times m_{ni_n}$ is equal to $(-1)^{n+h}$, where h is the number of permutations necessary to order the set $(i_1,\dots,i_n)$.

In the formulas for the hierarchical analysis, we need to use the sets of multiconnections characterized in Definition 3.5. This problem can be transformed to the generation of all the connections of the modified graph as follows.

We will assume that the signal-flow graph has $card(\hat{m}_{ij}) = 1$. Let us consider a set of multiconnections $C(B,E)$ where, according to Definition 3.5, B represents the initial nodes and E represents the terminal nodes of the multiconnections paths.

Let us renumber rows of the structural matrix in the following order: $NB - B \cup E, B - B \cap E, E - B \cap E, B \cap E$. Thus the first group of rows corresponds to the block nodes that are neither the beginning nor the end nodes of the multiconnection paths (the internal nodes). Next, there are the nodes that are the beginning of multiconnection paths, followed by the nodes that are the end nodes of the multiconnection paths. The final group of nodes are those nodes that are both the beginning and the end nodes of the multiconnection paths (the isolated nodes). Columns of the structural matrix are renumbered in a similar fashion as follows: $NB - B \cup E, E - B \cap E, B - B \cap E, B \cap E$. Such rearranged matrix is denoted as $M^1 = [m_{ij}]$.

Theorem 3.13 [226]. The set of multiconnections $C(B,E)$ can be obtained through the iterative expansion of the function $C(M^1)$ as follows:

$$C(M^i) = \bigcup_{j \in J_i} m_{1j} \times C(M_j^{i+1}) \tag{3.67}$$

where $m_{ij} \in M^i, i = 1,...,l, l = card(E)$, $J_i = \{1,...,l - i + 1\}$, and M_j^{i+1} is the matrix obtained from M^i after swapping columns 1 and j and removing its first row and column. Matrix M^i is any matrix in the expansion of $C(M^1)$ at the i-th step of the algorithm. In the last step of the algorithm we have $C(M^{l+1}) = \{\varnothing\}$, where $\{\varnothing\}$ is a unit element of Wang algebra [49]. The result is obtained with no duplicates. This result was simplified in [221] as follows

Lemma 3.7 [221]. The set of multiconnections P(B, E) of a graph with an incidence matrix M is equal to the set of connections of a graph described by the matrix M(B, E). The matrix M(B, E) is obtained from the matrix M by deleting:

all columns corresponding to nodes B;

all rows corresponding to nodes E.

Example 3.11. To generate the set of multiconnections P({ 1,2}, {3,4}) of the graph shown in Fig. 20, let us reduce the incidence matrix M, where

$$M = \begin{bmatrix} 0 & 7 & 2 & 0 & 1 \\ 0 & 9 & 3 & 4 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 5 & 6 \end{bmatrix} \tag{3.68}$$

by using lemma 3.7

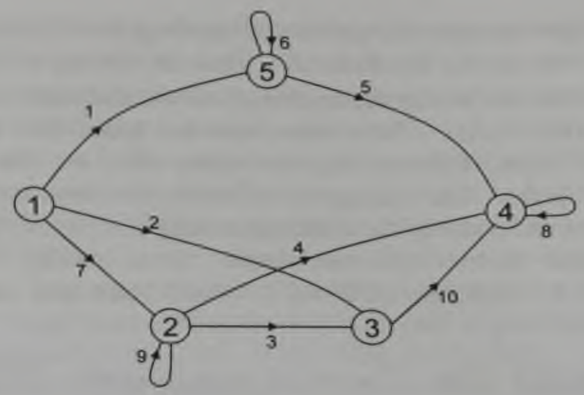


Fig. 3.17 An example of signal-flow graph

According to Lemma 3.7 M(B, E) is obtained from M by deleting columns 1 and 2, and rows 3 and 4. Hence

$$M(B,E)=\begin{bmatrix}2 & 0 & 1\\ 3 & 4 & 0\\ 0 & 5 & 6\end{bmatrix}. \tag{3.69}$$

Applying the formula (3.66) to M(B, E.) we obtain the sets of multiconnections P(B, E) so that

$$P(\{1,2\},\{3,4\})=\{\{2,4,6\},\{1,3,5\}\}. \tag{3.70}$$

3.5.2. Multitrees of a Directed Graph

Two types of directed trees are used in the hierarchical analysis of a directed graph $G_u(V_u,E_u)$. The first type (Definition 3.7) are the directed multitrees t_V defined by the set of vertices V_u, and the second type (Definition 3.9) are the directed multitrees t_D defined by the set of paths. Both types can be obtained by using a single algorithm and the efficiency of the topological analysis depends on the efficiency of such algorithm.

Tree generation is based on a simple observation that the set of trees that contains an edge *e* can be obtained by finding all the trees of the directed graph G_1 obtained from G after short-circuiting the directed edge *e*. Likewise the set of trees that does not contain an edge *e* can be obtained by finding all the trees of the directed graph G_2 obtained from G after removing the directed edge *e*. Obviously the two sets of trees obtained from graphs G_1 and G_2 are disjoint and the procedure converges to a solution since at each step the number of edges in both graphs is reduced by at least one.

To generate the directed multitrees t_D first renumber the graph vertices in the order: $h_1,h_2,...,h_\varepsilon,v_1,v_2,...,v_d,k_1,k_2,...,k_q$, where $\varepsilon=n-d-q$, and h_i $(i=1,2,...,\varepsilon)$

denote the internal vertices not included in the set D, while v_i $(i=1,2,\ldots,d)$ denote the beginning vertices of a path in D, and k_i $(i=1,2,\ldots,q)$ denote the end vertices of the path.

To generate the directed multitrees t_V we first renumber the graph vertices in the following order: $h_1, h_2, \ldots, h_\varepsilon, v_{11}, \ldots, v_{1m_1}, \ldots, v_{k1}, \ldots, v_{km_k}, r_1, \ldots, r_k$, where $\varepsilon = n-k-\sum_{i=1}^{k} m_i$, *and* h_i $(i=1,2,\ldots,\varepsilon)$ denote the internal vertices not included in V, v_{ij} *and* r_i $(i=1,\ldots,k,\ j=1,\ldots,m_i)$ denote the vertices in the same subtree with r_i selected as the reference node.

Theorem 3.14 [226].
The set of all multitrees of a directed graph G can be obtained through the iterative expansion of the function $T(M^1)$ as follows:

$$T(M^i) = \bigcup_{j \in J_i} m_{1j} \times T(M_j^{i+1}) \tag{3.71}$$

where $m_{ij} \in M^i, i=1,\ldots,n-k$, and where J_i is determined depending on the type of multitrees as follows:

1. To generate the directed multitrees t_D

$$J_i = \begin{cases} \{i+1,\ldots,n\} & for\ i \le \varepsilon \\ r_u & for\ i = v_u \end{cases} \tag{3.72}$$

where r_u is the terminal vertex of the path that begins at v_u

2. To generate the directed multitrees t_V

$$J_i = \begin{cases} \{i+1,\ldots,n\} & for\ i \le \varepsilon \\ \{i+1,\ldots,v_{um_u}, r_u\} & for\ v_{u1} \le i \le v_{um_u} \end{cases} \tag{3.73}$$

M_j^{i+1} is a matrix obtained from M^i after adding the i-th column to j-th column of the matrix M^i. In the last step of the algorithm we have $T(M^{n-k+1}) = \{\varnothing\}$, where $\{\varnothing\}$ is a unit element of Wang algebra [49]. The result is obtained with no duplicates.

Described algorithms are fast, simple to implement and require small computer memory. They can be applied to generate all types of multitrees and multiconnections required in the hierarchical topological analysis. Their implementation in the algorithms of topological analysis significantly increased the computational power of these algorithms leading to an efficient symbolic analysis of large analog circuits.

3.6. Other Advances in Topological Analysis Methods

Topological methods were also used to advance analog VLSI circuit analysis in other application areas and in development of specialized computer aided design (CAD) tools. Two such application areas that the author contributed to include analysis of

VLSI interconnects and large change sensitivity application to diakoptics. They are briefly described in this section. Full analysis of these application areas are beyond the scope of this work.

3.6.1. Hierarchical Analysis of High Frequency Interconnect Networks

A special form of topological analysis is also applicable to large regular networks typical for VLSI circuits, where identical models of network components or subnetwork macromodels can be used to simplify analysis. This is the case with large clock and power grid networks, and system level interconnect networks modeled as distributed networks or identical sections of lumped models that approximate such distributed networks.

Transmission line effects were not a serious concern when signal wavelength was significantly larger than the physical dimensions of the designed circuits. But in circuits with fine features and high clock frequencies, transmission line effects need to be considered for long interconnections [28]. This is the case in modern BiMOS/CMOS circuits, where the signal rise times are comparable to propagation delays.

Timing analyzers typically handle interconnections by using simple models based on RC trees, and either estimating the interconnect delay [73], [194], [156] or approximating the output waveforms [189], [188], [286], [179], [166], [283]. Estimation of the delay time is simple and reasonably accurate yielding reduction in the simulation time. However, as the signal frequencies increase, there is a pressing need to consider the analog behavior, due to the increasing effect of parasitics.

In [156] exact r-c-g models of distributed lines were used in order to improve timing accuracy. Another tendency in the interconnect analysis is to evaluate analog waveforms of the digital circuit response which, may cause unintended switching. This may go undetected by the network delay methods based on the RC trees. The asymptotic waveform evaluation method presented in [4], [189] satisfies many requirements for high frequency interconnect network analysis. It approximates the response of a circuit with floating capacitors, grounded resistors, inductors, and controlled sources. Its accuracy depends on the number of moments used. The method is from one to two orders of magnitude faster than HSPICE. This method was generalized in [166], where the moment polynomial nodal analysis was used to solve the interconnect networks. Different generalization of the asymptotic waveform evaluation, presented in [283], included lossy coupled transmission lines and nonlinear loads.

Another type of interconnect analysis used in [179] approximates driving point admittance of an RC tree in order to improve the simulation accuracy. A noteworthy approach was developed in [198], where the state equations in the complex frequency domain and the inverse Laplace transform were used to obtain time domain solutions of distributed line equations. The inverse Fourier transformation of the frequency domain scattering parameters was used in [276] to determine the impulse response of the transmission lines. Finally, Pade approximations of the transmission line's characteristic admittance were used in [147] to derive a recursive convolution formulation. This latest method is applicable to lossy transmission lines with nonlinear elements and is one to two orders of magnitude faster than Spice3.e.

Developed by the author method presented in [222] uses network equations in the complex frequency domain and an event-driven approach to interconnect analysis. In this analysis inverse Laplace transform is computed only when the response changes by more than a prespecified limit. In addition, a hierarchical analysis is used in order to reduce the simulation time in regular structures. As a result, a hierarchically organized interconnect network can be analyzed in logarithmic time. In [222] interconnect networks are analyzed using symbolic frequency domain analysis and exact models of distributed lines. The analyzed network must have a tree structure and may contain transmission lines as well as other linear two-ports with known transmission matrices. This method is very efficient, particularly when applied to networks with hierarchical structures, and produces accurate results. Time complexity of the method depends on the regularity of the analyzed network, and can be as low as a logarithmic function of the number of elements for a hierarchically organized structure, yielding several orders of magnitude reduction in simulation time over the leading simulators. The main features of this method are presented in the following part.

Complex structures of interconnect networks used in VLSI designs - such as clock, power grid, and control lines - show large degree of regularity in hierarchically organized designs. Identification of such hierarchically organized structures is relatively easy in a VLSI design, where the hierarchical design approach is typically used. Binary trees are used to represent hierarchical structures. Other networks are converted to such format before analysis by inserting auxiliary two-ports with unit transmission matrices. Major results of this method are presented netx for the reference.

Frequency Domain Analysis

All the two ports in the network are presented by their transmission matrices which relate input and output variables on the two-port terminals. In particular, a transmission line is represented by the matrix

$$T = \begin{bmatrix} \cosh(\gamma l) & Z_f \sinh(\gamma l) \\ \dfrac{\sinh(\gamma l)}{Z_f} & \cosh(\gamma l) \end{bmatrix} \tag{3.74}$$

where

$\gamma = \sqrt{(R+sL)(G+sC)}$, and $Z_f = \sqrt{(R+sL)/(G+sC)}$,

l is the length of the line, s is the complex frequency variable (in the Laplace transform), and R, L, C, *and* G are per unit length transmission line parameters. A single horizontal element (e.g. floating capacitor) is represented by a two-port described by

$$T = \begin{bmatrix} 1 & Z(s) \\ 0 & 1 \end{bmatrix} \tag{3.75}$$

where *Z(s)* is the impedance of this discrete element at a given complex frequency, and a single vertical element is represented by

$$T = \begin{bmatrix} 1 & 0 \\ Y(s) & 1 \end{bmatrix}. \tag{3.76}$$

Other two-ports described by transmission matrices in the complex frequency *s* domain may be used. Transmission matrices of all network tree branches are evaluated during the first step of the interconnect network analysis.

Next, the load and input admittances are computed at each tree node using tree folding procedure. The load admittance of a tree node is obtained by adding the input admittances of the branches connected to this node and the admittance of a discrete load at this node. The input admittance of a tree branch is a function of loads of the entire subtree structure connected to its output.

The procedure starts at each leaf of the tree and computes the discrete load admittance at the output of the leaf branch. Next, the input admittance of the leaf branch is computed using

$$Y_{in} = \frac{C + D\,Y_{load}}{A + B\,Y_{load}} \tag{3.77}$$

where *A, B, C, and D* are the parameters of the branch transmission matrix, and Y_{load} is its load admittance. Load admittance of a tree node is obtained by adding input admittances of the branches connected to this node and the admittance of a discrete load at this node.

Folding of the tree branches continues towards the root. The input admittance of a branch, which is not a leaf of the tree, is computed using the node load admittance as the branch load in equation (3.77). This input admittance is a function of loads and two-port parameters of the entire subtree structure connected to its output. This process continues until the root load admittance is computed.

The described algorithm for tree folding can be implemented very efficiently using a recursive procedure. The procedure starts at the root and calls itself at each branch. When a leaf of the tree is reached, which is easily identified by checking its outputs, the procedure evaluates the input admittance of the leaf. After all branches at a particular node of the tree have been analyzed, and their input admittances evaluated, the procedure evaluates node load admittance. Then the input admittance of the branch is evaluated and its value is returned to the higher level of the recursive call. The procedure stops after returning to the first level (stack of the recursive call is empty) and performs the evaluation of the root load admittance.

As the last stage of the complex frequency domain analysis, the required transfer functions from the root to the specific outputs are computed. This analysis is also straightforward and uses a property of the cascade connection of two-ports. The transmission matrix of such a connection equals to the product of transmission matrices of the constitute two-ports. Such a product can be obtained very efficiently for identical two-ports that characterize segments of the transmission lines.

Algorithm for analysis of the tree networks

Basic steps of the algorithm for tree network analysis can be summarized as follows:

1. Compute the transmission matrix of each branch.
2. Compute the discrete load admittance at each node.
3. Compute the input admittance of each branch.
4. Compute the load admittance of each node.
5. For each output find a transmission matrix

$$T = T_{i_1} T_{ld_{i1}} T_{i_2} T_{ld_{i2}} \cdots T_{i_{out}} \tag{3.78}$$

where $i_1, i_2, \ldots, i_{out}$ are the indices of branches on the unique path from the root to the output node, and where T_{ld_i} is the transmission matrix, which represents the load effect of the stray branches at the i-th node of the tree (a stray branch is a branch not included in the unique path from a given output node to the root). Matrix T_{ld_i} is computed as

$$T_{ld_i} = \begin{bmatrix} 1 & 0 \\ Y_{ld_i} & 1 \end{bmatrix} \tag{3.79}$$

and $Y_{ld_i} = Y_i - Y_{in_{i+1}}$, where (i+1) stands for the index of the next branch in the path from the input to the output.

6. For each output find the transfer function

$$\frac{V_{out}}{V_{in}} = \frac{1}{A + B Y_{out}} \tag{3.80}$$

Each of these steps takes time proportional to either the number of nodes or the number of branches in the tree. Since the number of branches in a tree is equal to the number of nodes minus one, we see that this algorithm can be completed in time which is a linear function of the number of branches. In the hierarchical analysis of an interconnect network, at least one branch is described by a hierarchically organized structure (HOS). A HOS is defined as an interconnect structure of identical cells. Each cell may contain transmission lines, discrete two-ports and HOS's. If a cell contains no HOS then it is considered a simple branch, otherwise the hierarchy level of the HOS is increased by one.

Hierarchical Analysis

The concept of hierarchical analysis of high frequency interconnects is based on two observations. The first observation relates to the existence of the identical branches in many interconnect networks. If the branches are identical, then only one of them has to be analyzed and all such branches have identical transmission matrices. The second observation relates to the existence of the identical subtrees. This is often the case in hierarchical design of VLSI circuits, where lower level functional blocks are reused to obtain higher level subsystems. The input admittances of such subtrees are equal. Using these two observations one can save considerable processing time.

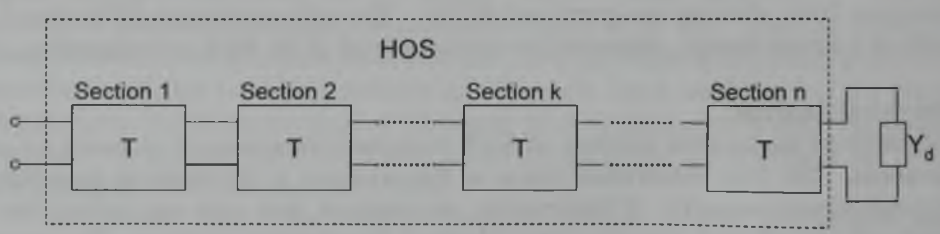
Let us compare the computing effort to analyze a single, hierarchically organized structure, with a similar effort for the flat network analysis. Assume first, that we analyze a single hierarchical level, and that the two-ports, which represent the subcircuits at this level, are identical and connected in a regular way. A typical, regular connection of two-ports on a single hierarchy level is the cascade connection. Other connections, for example serial or parallel connection of two-ports, are possible. The only requirement is that they are repeated in every substructure present on the same level of hierarchy.

These regular connections of two-ports can be analyzed very efficiently. For instance, the cascade connection of identical two ports can be analyzed in constant time, independent on the number of cascaded two-ports. This analysis can be handled efficiently using the eigenvalues and the characteristic equation of the transmission matrices describing the cascaded two-ports.

The analysis time for the same structure in the flat network is, in the best case, proportional to the number of sections. Therefore, the analysis can be up to n times faster, where n is the total number of identical two-ports in the regular substructure. Now, let us suppose that on the next level of hierarchy similar savings are obtained and the analysis time of the second level sections is reduced m times. Thus the combined savings in time for the two level hierarchy is proportional to the product nm. If the hierarchy is organized in such a way that each level has the same number of sections, for instance n, then the k level hierarchical structure can be analyzed up to n^k times faster than the flat level network. Thus the hierarchical analysis of regular interconnect network can be performed in time proportional to the logarithm of the number of nodes.

Evaluation of transfer functions in hierarchical analysis

After all hierarchical branches are analyzed, one can evaluate the transfer functions from the input to the individual outputs. Let us consider a cascade of n identical two-ports and assume that the output is inside the two-port k (see Fig. 3.18a). Depending on the two-port internal structure, we may design different ways to evaluate the transfer function from its input to an internal point $T_{i_{out}}$. The most typical structures used in the interconnect networks are Γ and T cells, built of transmission lines with discrete loads at the output of each line. For illustration let us assume that each section is a Γ section with the horizontal line being a transmission line, and the vertical line being a lower level hierarchical substructure (see Fig. 3.18b).



a)

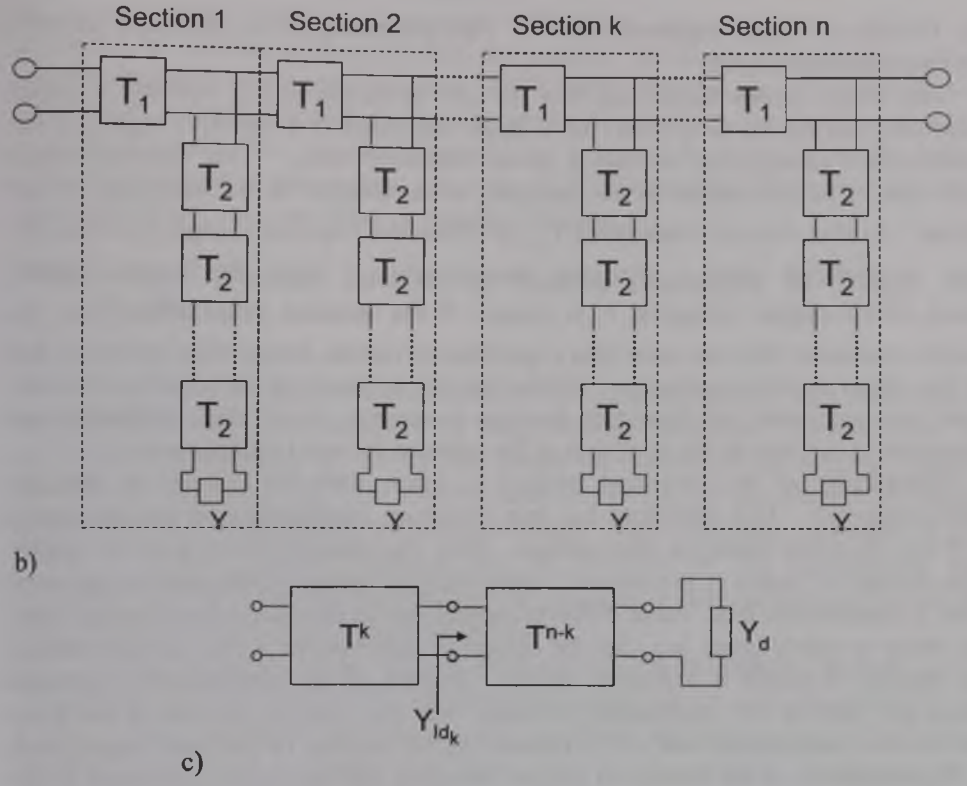


Fig. 3.18. A hierarchically organized structure (a) cascade of n sections, (b) two level HOS, (c) illustration of the intermediate stage of HOS analysis

The transfer function from the input to the output of section k is equal to T^k. At this point the cascade is broken in two parts (see Fig. 3.18(c)). The first part from the cascade's input to the output of section ***k*** (represented by its transmission matrix T^k), and the rest from the output of section ***k*** to the output of the cascade represented by the load admittance Y_{ld_k}. The load admittance Y_{ld_k} can be evaluated by using parameters of the transmission matrix T^{n-k}, and the cascade's load. Details of evaluation of Y_{ld_k}, T_{ld_k} and $T_{i_{out}}$ and the desired transfer functions are presented in [222]. In the computer program that computes $T_{i_{out}}$ a recursive procedure is used, therefore the code that implements this method is simple and efficient.

Time Domain Solution and Numerical Results

To obtain the time domain response on a specific output, first multiply the transfer function by the Laplace transform of the input signal, and then use the inverse Laplace transform. To compute the output function value in time domain v(t), from its Laplace transform V(s), the numerical Laplace transform inversion technique presented in [267] is used. Selecting M = 8 and N = 6, results in Pade approximation (see [267]) that gives

correct 15 terms of Taylor expansion of e^{st}. This selection gives us sufficient accuracy for the interconnect analysis.

Fast, event driven simulators, like the one presented in [3], evaluate a circuit response only during the time when the voltage computed in a specified region of the analyzed circuit changes by at least a given threshold value. This approach saves analysis time, since the circuit is not analyzed when changes in its responses are not significant. At first, the next time point t_i, at which the response changes by a specific amount, is predicted using a quadratic approximation. Next, the inverse Laplace transform of the output voltage at t_i is found. If the obtained value differs from the previously evaluated value by more than a specified deviation, reduce time increment and repeat the output function evaluation. Notice that the accuracy of the solution evaluated at a new time point does not depend on the time increment. So, all the computed circuit responses are as accurate as the precision of the selected inverse Laplace method.

Complexity of the presented method is linear with the number of different two-ports analyzed. This statement has two important implications on the computing time of the algorithm based on this method. First, the number of two-ports is smaller than the number of nodes in the network, particularly if several ladder sections are used to model a transmission line. Other efficient algorithms for the interconnect analysis may also be linear or nearly linear (e.g. like the algorithm presented in [189]), but they depend on the number of nodes in a discrete model. Second, if the hierarchically organized structures are parts of the interconnect network, then the effective number of two-ports needed for the interconnect analysis is reduced to the number of different transmission lines. This reduction of the number of unique two-ports that need to be considered by the method gives savings in time, which may be very significant in the regular VLSI designs.

Interconnect analysis program Connect, which realizes this method, has been written in C++ and implemented on Sun workstations at ATT Bell Laboratories. In order to demonstrate efficiency of the presented method analysis time required by Connect program was compared with time needed by Advice, a general circuit simulator used in ATT Bell Labs [184]. The results of this approximation are presented in Table 3.1. Both programs were run on Sun-3/60 and results shown in Table 3.1 represent the combined user time and system time for analysis.

Table 3.1

Performance comparisons

circuit name	number of			time in sec	
	lines	elements	nodes	Connect	Advice
net16_1	16	32	18	8	55
net16_8	16	256	130	8	98
net136_1	136	272	138	12	102
net136_8	136	2176	1090	12	477
net528_1	528	1056	530	5	298
net528_8	528	8448	4226	5	2593

In Table 3.1, the number of elements and nodes correspond to a discrete models simulated by Advice program, while the number of lines corresponds to distributed models simulated by Connect program. The number of nodes, in case of distributed lines modeled by a single Γ section, is equal to the number of lines plus 2. Thus, based on the simulation results, the presented hierarchical topological approach to the analysis of large high frequency interconnect networks in VLSI circuits, is more than two orders of magnitude faster than the industry standard simulator.

3.6.2. Large Change Sensitivity Based Diakoptic of Large Analog Networks

As a last example of applications of the network topology to circuit analysis let us consider a case of diakoptics in large analog networks. Fundamental to this approach is large change sensitivity that relates the solution of the network equations to the network topology, and uses incident matrices to specify the amount of change that result from the change in the component values. This approach was used in [251] to derive a simple, compact notation for the solution of the decomposed network equations. Through the use of ideal switches inserted in the partition nodes, the process of decomposition was conceptually simplified.

Over the years, many techniques have been developed to analyze partitioned networks in order to reduce the computational effort and to save the computer storage space. These techniques originated with diakoptics introduced by Kron [138]. The partitioning and sparse matrix techniques were later combined, producing yet more efficient methods [55], [281], [204]. Matrix modification techniques were used in [268], [12], and [102] to simplify analysis in cases when only some coefficients change in the system equations. Further development included application of the hierarchical decomposition approach to network analysis which allows analysis of very large networks with great efficiency [89], [268]. Please refer to references [99], and [101] for a discussion of the efficiency of these approaches.

A common problem of the decomposition techniques is the apparent complexity of the algorithms and the lack of transparency at the high levels of abstraction used. By using the network topology, a simple direct explanation of decomposed network analysis can be obtained as presented in [251].

Equations of a decomposed nonlinear network

Consider a large nonlinear network decomposed into s subnetworks. The separation of subnetworks can easily be achieved by inserting ideal switches between the interconnection nodes (Fig. 3.19). When the switches are open the network is decomposed and each subnetwork can be solved separately. When the switches are closed the original network is obtained. The objective is to update solutions in each subnetwork according to the rules of the large change sensitivity approach [267]. For simplicity of presentation it is assumed that each subnetwork contains a common reference node. This requirement does not restrict the applications of the proposed approach. A general partition can be implemented as discussed in [89].

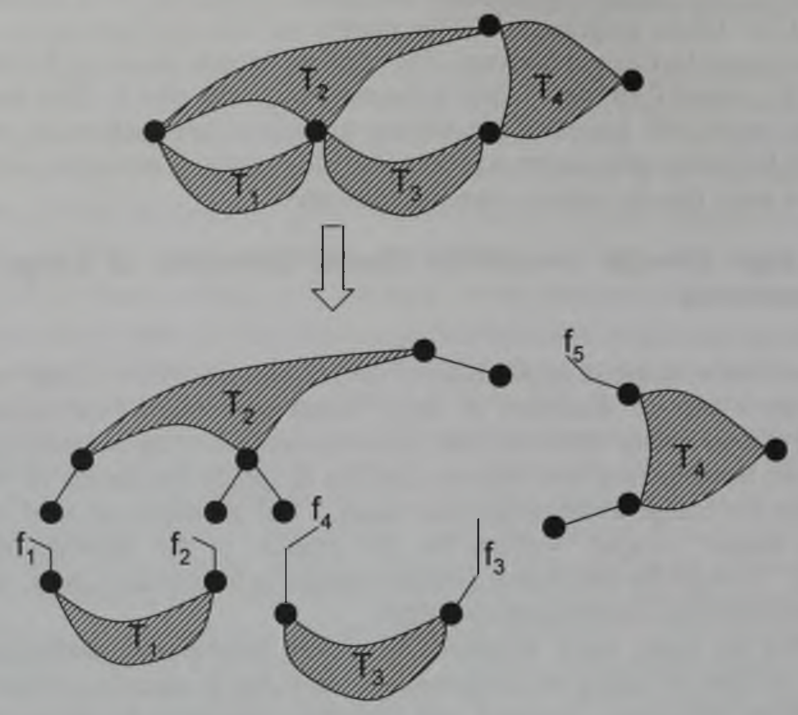


Fig. 3.19. Network T decomposed by insertion of the ideal switches

Each subnetwork can be described by a vector equation

$$g_i(y_i) = 0 \qquad i = 1,2,...,s \tag{3.81}$$

where the independent variables y_i represent either nodal voltages or branch currents. Both vectors g_i and y_i have the same dimension n_i .

If two nodes *j* and *m* are connected by an ideal switch *f*, an unknown current i_f is added to the Kirchhoff current law (KCL) equation at the node *j*, and the identical current i_f is subtracted from the KCL equation at the node *m*. Subnetwork equation (3.81) is then augmented by the additional equation

$$(v_j - v_m) F + (F - 1) i_f = 0 \tag{3.82}$$

in which v_j is an element of the vector y_1, v_m is an element of y_2 and i_f, is an additional variable. The value *F* is 0 for the open switch and 1 for the closed switch [267]. Overall, we must add t such equations where t is the number of switches used for interconnections. The system of nonlinear equations $g(y) = 0$ is solved through Newton-Raphson iterative process using the following equation:

$$\frac{\partial g(y^k)}{\partial y}\,\Delta y^k = -\,g(y^k) \tag{3.83}$$

where Δy^k is a $n \times 1$ vector of the incremental changes in the k-th iteration and n is the number of unknown currents and voltages plus the number of switches. The Jacobian of the system equations has the following form:

$$T = \frac{\partial g(y^k)}{\partial y} = \begin{bmatrix} T_1 & & & & \lambda_1 \\ & T_2 & & & \lambda_2 \\ & & \ddots & & \vdots \\ & & & T_s & \lambda_s \\ F_1 & F_2 & & F_s & F - I \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial y_1} & & & & \lambda_1 \\ & \frac{\partial g_2}{\partial y_2} & & & \lambda_2 \\ & & \ddots & & \vdots \\ & & & \frac{\partial g_s}{\partial y_s} & \lambda_s \\ F_1 & F_2 & & F_s & F - I \end{bmatrix} \tag{3.84}$$

where F_i contain switch variables (0 or 1) and λ_i are parts of the incidence matrix which correspond to switch locations in different subnetworks. In addition

$$y^k = \begin{bmatrix} y_1^k \\ y_2^k \\ \vdots \\ y_s^k \\ i_f^k \end{bmatrix}, \quad g(y^k) = \begin{bmatrix} g_1(y_1^k) + \lambda_1 i_f^k \\ g_2(y_2^k) + \lambda_2 i_f^k \\ \vdots \quad \vdots \quad \vdots \\ g_s(y_s^k) + \lambda_s i_f^k \\ \sum_{i=1}^{s} F \lambda_i^T y_i^k + (F-1)\, I\, i_f^k \end{bmatrix}, \quad i_f^k = \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i \end{bmatrix}, \tag{3.85}$$

and

$$F_i = F\,\lambda_i^T \tag{3.86}$$

When all switches are open $F_i = 0$, and the coefficient matrix becomes block triangular

$$T_0 = \begin{bmatrix} T_1 & & & & \lambda_1 \\ & T_2 & & & \lambda_2 \\ & & \ddots & & \vdots \\ & & & T_s & \lambda_s \\ & & & & -I \end{bmatrix} \tag{3.87}$$

Only the changes in the ideal switches are considered in this analysis, however, an extension to a general case with large changes in other parameters is straightforward.

Let us call T_0 a "nominal" matrix. It corresponds to the "nominal" system equations in which all the subnetworks are disconnected:

$$T_0\, X_0 = W_0 \tag{3.88}$$

where

$$X_0 = \Delta y^k\Big|_{F=0} \quad W_0 = -\, g\big(y^k\big)\Big|_{F=0} \tag{3.89}$$

Equation (3.88) can be easily solved as the inverse of T_0 is obtained explicitly as

$$T_0^{-1} = \begin{bmatrix} T_1^{-1} & & & & T_1^{-1}\lambda_1 \\ & T_2^{-1} & & & T_2^{-1}\lambda_2 \\ & & \ddots & & \\ & & & T_s^{-1} & T_s^{-1}\lambda_s \\ & & & & -I \end{bmatrix} \Rightarrow \quad X_0 = T_0^{-1} W \tag{3.90}$$

so, each block can be solved (factorized) separately and X_0 calculated with much lower effort than without decomposition. When all switches are closed, corresponding to setting variable F to one, the interconnected network is obtained and equation (3.83) can be replaced by

$$T\, X = W \tag{3.91}$$

where

$$T = \frac{\partial g\big(y^k\big)}{\partial y}\Bigg|_{F-1} = \begin{bmatrix} T_1 & & & & \lambda_1 \\ & T_2 & & & \lambda_2 \\ & & \ddots & & \vdots \\ & & & T_s & \lambda_s \\ \lambda_1^T & \lambda_2^T & & \lambda_s^T & 0 \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial y_1} & & & & \lambda_1 \\ & \frac{\partial g_2}{\partial y_2} & & & \lambda_2 \\ & & \ddots & & \vdots \\ & & & \frac{\partial g_s}{\partial y_s} & \lambda_s \\ \lambda_1^T & \lambda_2^T & & \lambda_s^T & 0 \end{bmatrix} \tag{3.92}$$

and

$$X = \Delta y^k\Big|_{F=1} \quad W = -\, g\big(y^k\big)\Big|_{F=1} \tag{3.93}$$

To calculate X use large scale sensitivity approach [267]

$$X = X_0 - T_0^{-1} P\, z \tag{3.94}$$

where z is obtained from

$$\left(\delta^{-1} + Q^T T_0^{-1} P\right) z = Q^T T_0^{-1} W \tag{3.95}$$

The only changes that need to be considered to represent connection of blocks in a single network are the changes in the interconnecting switch values from 0 to 1, therefore δ = I and the size m of the auxiliary system is equal to the number of switches.

P and Q are the $n \times t$ topological matrices which indicate the location of the switches used for the network decomposition

$$P = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{bmatrix}, \qquad Q = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_s \\ I \end{bmatrix} \tag{3.96}$$

Using (3.90) and (3.96) equation (3.95) can be simplified to

$$\left[\sum_{i=1}^{s} \lambda_i^T \left(\frac{\partial g_i}{\partial y_i} \right)^{-1} \lambda_i \right] z = Q^T X_0 \tag{3.97}$$

Example. Use the large-scale sensitivity based decomposition approach to solve the example network shown in Fig. 3.20. Assume that s = 1

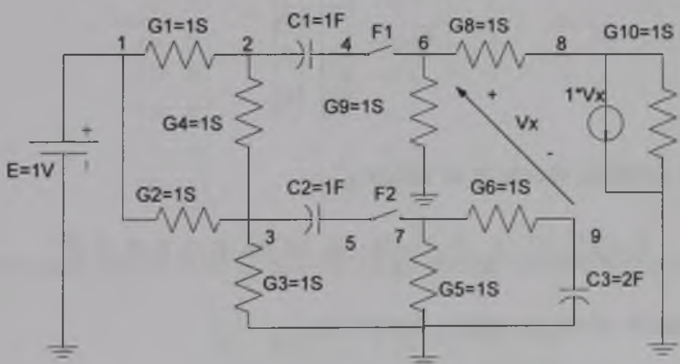


Fig. 3.20. The example network

We have two subsystems separated by two switches

$$T_1X_{10}=\begin{bmatrix} G_1+G_4+sC_1 & -G_4 & -sC_1 & 0 \\ -G_4 & G_2+G_3+G_4+sC_2 & 0 & -sC_2 \\ -sC_1 & 0 & sC_1 & 0 \\ 0 & -sC_2 & 0 & sC_2 \end{bmatrix}\begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}=\begin{bmatrix} EG_1 \\ EG_2 \\ 0 \\ 0 \end{bmatrix}$$

and

$$T_2X_{20}=\begin{bmatrix} G_8+G_9 & 0 & -G_8 & 0 \\ 0 & G_5+G_6 & 0 & -G_6 \\ 1-G_8 & 0 & G_8+G_{10} & -1 \\ 0 & -G_6 & 0 & G_6+sC_3 \end{bmatrix}\begin{bmatrix} v_6 \\ v_7 \\ v_8 \\ v_9 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The nominal solution of the first subsystem is

$$T_1X_{10}=\begin{bmatrix} 3 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}=\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \Rightarrow X_{10}=T_1^{-1}\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}=\begin{bmatrix} 0.6 & 0.2 & 0.6 & 0.2 \\ 0.2 & 0.4 & 0.2 & 0.4 \\ 0.6 & 0.2 & 1.6 & 0.2 \\ 0.2 & 0.4 & 0.2 & 1.4 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}=\begin{bmatrix} 0.8 \\ 0.6 \\ 0.8 \\ 0.6 \end{bmatrix}$$

and the nominal solution of the second subsystem is

$$X_{20}=T_2^{-1}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Thus the nominal solution vector is as follows

$$X_0=[X_{10}^T\ X_{20}^T\ I_{F1},\ I_{F2}]^T=[0.8\ \ 0.6\ \ 0.8\ \ 0.6\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]^T$$

The coefficient matrix of the complete system is

$$T = \begin{bmatrix} & T_1 & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 1 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & & & & & -1 & 0 \\ 0 & 0 & 0 & 0 & & T_2 & & & 0 & -1 \\ 0 & 0 & 0 & 0 & & & & & 0 & 0 \\ 0 & 0 & 0 & 0 & & & & & 0 & 0 \\ 0 & 0 & F_1 & 0 & -F_1 & 0 & 0 & 0 & F_1-1 & 0 \\ 0 & 0 & 0 & F_2 & 0 & -F_2 & 0 & 0 & 0 & F_2-1 \end{bmatrix}$$

where the incidence matrices are

$$\lambda_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \lambda_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

and topological matrices describing switch location and switch changes are

$$P = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ I \end{bmatrix}, \quad \delta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

To get the corrector vector z let us use the large change sensitivity approach

$$\left(\delta^{-1} + Q^T T_0^{-1} P\right) z = Q^T \underbrace{T_0^{-1} W}_{X_0}$$

or in more detail

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + Q^T \underbrace{T_0^{-1} \begin{bmatrix} 0 \\ \hline 1 & 0 \\ 0 & 1 \end{bmatrix}}_{last\ 2\ columns\ of\ T_0^{-1}} \right) \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = Q^T X_0$$

and solve the reduced size equation

$$\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \lambda_1^T & \lambda_2^T & I \end{bmatrix} \begin{bmatrix} T_1^{-1}\lambda_1 \\ T_2^{-1}\lambda_2 \\ -I \end{bmatrix} \right) \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \left(\lambda_1^T T_1^{-1}\lambda_1 + \lambda_2^T T_2^{-1}\lambda_2 \right) \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$$

where

$$\lambda_1^T T_1^{-1}\lambda_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.6 & 0.2 & 0.6 & 0.2 \\ 0.2 & 0.4 & 0.2 & 0.4 \\ 0.6 & 0.2 & 1.6 & 0.2 \\ 0.2 & 0.4 & 0.2 & 1.4 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.6 & 0.2 \\ 0.2 & 0.4 \\ 1.6 & 0.2 \\ 0.2 & 1.4 \end{bmatrix} = \begin{bmatrix} 1.6 & 0.2 \\ 0.2 & 1.4 \end{bmatrix}$$

and

$$\lambda_2^T T_2^{-1}\lambda_2 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & 0.05 & 0.25 & 0.1 \\ 0 & 0.6 & 0 & 0.2 \\ 0 & 0.1 & 0.5 & 0.2 \\ 0 & 0.2 & 0 & 0.4 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.05 \\ 0 & 0.6 \end{bmatrix}.$$

Then compute z using

$$\begin{bmatrix} 2.1 & 0.25 \\ 0.2 & 2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$$

and the corrector vector z is equal to:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0.3494 \\ 0.2651 \end{bmatrix}$$

The final solution vector is obtained by combining the nominal solution vector X_0

$$X_0 = [X_{10}^T, X_{20}^T, I_{F1}, I_{F2}]^T = [0.8 \;\; 0.6 \;\; 0.8 \;\; 0.6 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$$

with the correction term as follows

$$T_0^{-1}Pz = \begin{bmatrix} 0.6 & 0.2 \\ 0.2 & 0.4 \\ 1.6 & 0.2 \\ 0.2 & 1.4 \\ -0.5 & -0.05 \\ 0 & -0.6 \\ 0 & -0.1 \\ 0 & -0.2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0.3621 \\ 0.2638 \end{bmatrix} = \begin{bmatrix} 0.2627 \\ 0.1759 \\ 0.6120 \\ 0.4410 \\ -0.1880 \\ -0.1590 \\ -0.0265 \\ -0.0530 \\ -0.3494 \\ -0.2651 \end{bmatrix} \Rightarrow \quad X = X_0 - T_0^{-1}Pz = \begin{bmatrix} 0.5373 \\ 0.4241 \\ 0.1880 \\ 0.1590 \\ 0.1880 \\ 0.1590 \\ 0.0265 \\ 0.0530 \\ 0.3494 \\ 0.2651 \end{bmatrix}$$

The presented large-scale sensitivity based decomposed network analysis can also be used in the case of hierarchical decomposition [98], [268], [287]. The concept can be explained by analyzing equation (3.97). The coefficient matrix on the left hand side of this equation may be big enough to justify its decomposition. In the described switch based approach, this means that not all switches will be closed at the same stage of the network analysis. Instead, the switches will be closed according to the current level of hierarchical partition analyzed. As an effect of closing the first level switches, small subnetworks will be combined into larger subnetworks and the solution vector X_0 will be updated to its first level update X_1. The next group of switches will then be closed to obtain larger subnetworks (obtained on the second level of decomposition hierarchy) and the solution vector X_1 updated to X_2. This procedure is repeated until the entire network is put back together on the highest level of decomposition hierarchy.

This hierarchical decomposition approach applies to both linear and nonlinear circuits. Resultant subnetworks can be solved separately, allowing the computations to be performed in parallel increasing computing efficiency of the described method.

3.7 Summary of Topological Analysis Methods

Decomposition techniques significantly improved efficiency of topological analysis. Based on software implementations the simulation time for topological analysis of a graph with n nodes using direct decomposition can be estimated as follows

$$T_2(n) = a\, b^{\sqrt{cn}} \tag{3.98}$$

where a, b, and c depend on the network topology and type of decomposition. When a descending hierarchical analysis is used, this analysis time can be estimated as

$$T_3(n) = a\, n^b \tag{3.99}$$

When the ascending hierarchical decomposition is used the analysis time is further reduced. Assuming that the network size grow as a function of n, and also assuming that the substitute graphs have fixed size, one can observe a linear growth of the analysis time with the network size

$$T(n) = a\, n \tag{3.100}$$

To summarize, topological methods produce fully symbolic results of analog circuit analysis. Such fully symbolic form of analysis results was not achievable using other analysis methods known at time when this research was developed. This work and the following publications [242], [221] revived interests in the symbolic analysis area. Subsequently, a number of methods [110], [128], [130], [154], [207], [208], and computer programs like ISAAC [91], ASAP [80], SYNAP [209], SAPEC [154], SSPICE [274], SCAPP [110], and GASCAP [118] applicable to symbolic analysis of large electronic networks were developed. In addition, a series of international workshops devoted to symbolic methods and applications in electronic circuit design were developed in 1990-ies and continue up to this day to focus interests of researchers in this area (see SMACD International Workshops).

Topological methods were also used in computer analysis of switched-capacitor (SC) circuits [137], [50], [131], [140], [145]. One of the most prominent is a symbolic SC simulator, SCYMBAL, developed by Konczykowska [129]. This program can produce fully symbolic transfer functions based on the topological evaluation of the SC networks. Another computer-aided tool for the symbolic analysis of SC networks was presented in [84]. Its signal flow graph generation process uses rule-based techniques. Symbolic analysis of switched-capacitor circuits became a useful designer tool. In addition, determinant-based methods of symbolic analysis proved to be as efficient as the graph based methods [90]. Symbolic analysis opened new possibilities for the CAD tools, not only in the circuit simulation but also in other tasks in the design and testing cycle of modern analog integrated circuits.

Modern symbolic analysis can be applied not only to lumped, linear, time-continuous, and switched-capacitor analog circuits in the frequency domain. By using Taylor expansion around the operating point in nonlinear circuits, including both CMOS and bipolar circuits [115] and using small signal linear models, symbolic analysis was extended to nonlinear circuits. It can be used to obtain analytic expressions for transfer functions, common-mode rejection ratios, power supply rejection ratios, impedances, noise analysis, etc. Furthermore, approximation and truncation of symbolic formulas was used to improve efficiency and clarity of the obtained expressions in large circuits [208], [274], and [110]. It was further extended to include symbolic expressions for poles and zeros [8] and symbolic analysis of harmonic distortions in nonlinear circuits [270].

An approximate symbolic analysis was developed to address the problem of large complexity of the obtained symbolic results and to present these results in more understandable format [284],[285]. To perform an accurate and efficient simplification, adaptive scaling mechanism was proposed [80]. Another approach to approximate exact symbolic transfer functions was based on finding sensitivity of the magnitude of a

network function with respect to its coefficients. The algorithm that uses this approach enumerates a small number of the dominant terms for each coefficient reducing complexity of underlying symbolic expressions [269].

However, symbolic manipulation in these methods was complicated and time consuming. Therefore, an exact symbolic analysis based on hierarchical decomposition and a graphical representation of symbolic determinants called determinant decision diagrams (DDD) was introduced in [255]. DDD's take advantage of the coefficient matrix sparsity leading to exact and canonical symbolic analysis that share symbolic expressions to improve computing efficiency. Efficiency of this method exceeds efficiency of numerical analysis programs like Spice. It is also faster than other symbolic programs such as ISAAC and Maple-V, and uses less computer memory [213].

Excellent reviews of symbolic analysis methods were presented in [143] and [92], with a comparative study of symbolic analysis methods for analog integrated circuits presented in [271].

4. Topological Diagnosis

Topological diagnosis uses network topology to determine fault location, fault verification, testability conditions, test ambiguity, test coverage and test point selection, leading to results that do not depend on specific parameter values. The author was one of the pioneers to develop topological diagnosis and fault location in electronic circuits.

Fault diagnosis of analog circuits has been one of the most challenging topics for researchers and test engineers since the 1970s. Given the circuit topology and the nominal circuit parameter values, fault diagnosis is to obtain the exact information about the tested circuit based on the analysis of the limited amount of measured circuit responses. There are three dominant and distinct stages in the process of fault diagnosis: fault detection to find out if the circuit under test (CUT) is faulty comparing with the fault-free or "gold" circuit, fault identification to find out where the faulty parameters are located inside the faulty circuit, and parameter evaluation to find out how much the faulty parameters deviate from their nominal values and to obtain values of other circuit parameters such as branch and nodal voltages, transistor gains, doping levels, threshold values, e.t.c..

The bottlenecks of analog fault diagnosis primarily lie in the inherited features of analog circuits: nonlinearity, parameter tolerances, limited accessible nodes, and lack of efficient fault models. Multiple fault diagnosis techniques are even less developed than single fault diagnosis, because it is more difficult to model and detect multiple faults, particularly in the presence of tolerances or the measurement noise. In addition, in the multiple fault situations one fault's effect on the circuit could be masked by the effects of the other faults leading to low testability circuits and ambiguity groups. Generally speaking, there was no widely accepted paradigm for analog test or fault diagnosis even with the introduction of IEEE 1149.4 standard for mixed-signal test bus.

With recent development of electronic design automation tools and widespread applications of analog VLSI chips and mixed-signal systems in the area of wireless communication, networking, neural networks and real-time control, the interests in analog test and fault diagnosis have been revived. Different building blocks including analog, digital, or software components can be integrated and embedded within the System-on-Chip (SOC) technology. Consequently, new challenges are brought to test and verify those mixed-signal products, mostly caused by increased complexity and reduced die size as well as limited accessibility of individual SOC circuits and their components. Several good periodical reviews on this topic appeared in 1979 [69], 1985 [11], 1991 [149] and 1998 [265], respectively.

Author worked intensively over a number of years to develop topological methods in analog testing, fault diagnosis, and fault simulation. In this section some results based on his work in topological fault diagnosis are presented. Discussed is the use of the network graph topology and hierarchical decomposition to locate faults in large-scale analog circuits, signal-flow graph based testability conditions, measures of testability, location of faulty regions, evaluation of faulty elements, node fault diagnosis, and consideration of element tolerances. Topological methods are applied to a generalized fault diagnosis in dynamic analog circuits.

This is followed by a discussion of a multiport approach to fault location that demonstrates implications of block dependency on network testability and resulting restrictions on the network topology for a circuit to be testable. A popular method of analog testing is based on the sensitivity approach applicable alike to linear and nonlinear networks. Sensitivity based testing method was improved by considering the network topology and developing a decomposition approach in such testing.

In circuits with small number of faults, fault verification methods are very useful. They can evaluate multiple fault circuits even with large deviations in parameter values. Thus, catastrophic faults like shorts and opens can be considered by verification methods. These methods are presented together with fault diagnosis equations and large change sensitivity approach to verification based testing. Another focus point in this section is devoted to low-testability circuits and ambiguity groups. By using ambiguity groups, the circuit testability can be established and ambiguity groups that contain untestable components can be efficiently determined for a given circuit topology and the selected test points. Finally, the network topology is explored in the entropy based test point selection to optimize testing effort and the accuracy of the test results.

4.1. Fault Location in Nonlinear Networks

Over a number of years, various approaches to fault location were developed. These include two main types of fault location methods: simulation before test (SBT) [113], [263] and simulation after test (SAT) [280], [260], [20], [233], [9], [266], [103], [167], [195], [113], [200].

The SBT requires the simulation of all possible faults and formulation of a fault dictionary. The analyzed network responses are compared with the fault dictionary and the most similar match determines a most likely fault. This method is the most suitable for a single catastrophic fault location (short or open faults). The dictionary approach is not practical for multiple faults or soft faults since the size of the fault dictionary and the cost to produce the one grow fast with the number of faults. The SAT methods either identify all parameters [260] or locate only the faulty elements [280], [20] assuming that the number of faults is small. This gives much more flexibility in detecting the multiple faults, and makes the method computationally efficient. A compromise must be reached between the computational effort and the number of accessible test nodes [280].

Following this work author developed a unified approach to fault location and analog diagnosis based on graph decomposition [201]. The circuit under test is hierarchically decomposed into subnetworks to locate multiple faults within the smallest possible subnetworks. A nodal decomposition is carried out for the CUT with the requirement that the measurement nodes must include the nodes of decomposition. In the method such nodes are used to formulate test equations. The voltage measurements are employed to isolate the faulty subnetworks. Using the voltage measurements is both more precise and more convenient than using current measurements in practical testing. Necessary and almost sufficient conditions for a subnetwork or a group of subnetworks to be fault free were developed based on Kirchhoff current law and the network topology. Logical analysis of the results of these tests is carried out to identify the faulty subnetworks. The result applies both to the linear and nonlinear network testing yielding a unifying decomposition approach to analog circuit testing.

In analog integrated circuits the network elements are randomly distributed within specified tolerance intervals. In the unified decomposition approach, statistical methods are used to check testing conditions under these random changes. The method also leads to identification of faulty elements or regions within each of faulty subnetworks. Depending on the size of the subnetwork, either the fault verification is carried out directly or the network is broken into several smaller subnetworks, with subsequent fault verification test in faulty regions. The main idea is to isolate faults to smallest possible regions using decomposition approach before the specific fault in the CUT is identified.

4.1.1. Network Decomposition

It is natural to assume that the topology of the network under test, its nominal parameter values and their tolerances are known. Before the test can be completed a nodal decomposition of the network is performed (see Fig. 2.1a). This yields several subnetworks connected to each other at the decomposition nodes. No mutual coupling is permitted between subnetworks and the nodes of decomposition should be selected from the set test nodes where voltage measurements can be performed. The decomposition is either done manually or by using an efficient decomposition algorithm [204], [225].

Example 4.1 [201]. Consider an example of the video amplifier circuit shown in Fig. 4.1. This network is decomposed at the nodes number 1, 2, 5, 7, and 10. Thus the circuit is decomposed into eight subnetworks, as shown in Fig. 4.2. In the actual testing voltage measurements are performed at the nodes of decomposition indicated on the figure. The network is not physically torn but rather logically divided into block structure for the purpose of fault location.

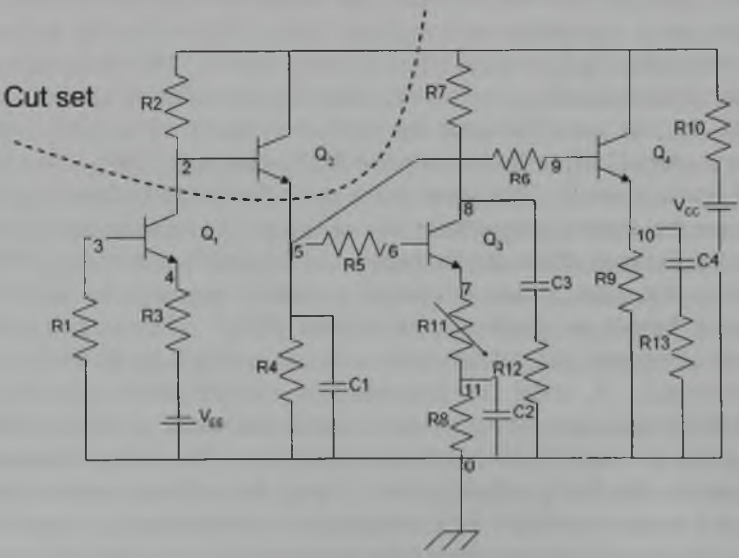


Fig. 4.1. A video amplifier circuit

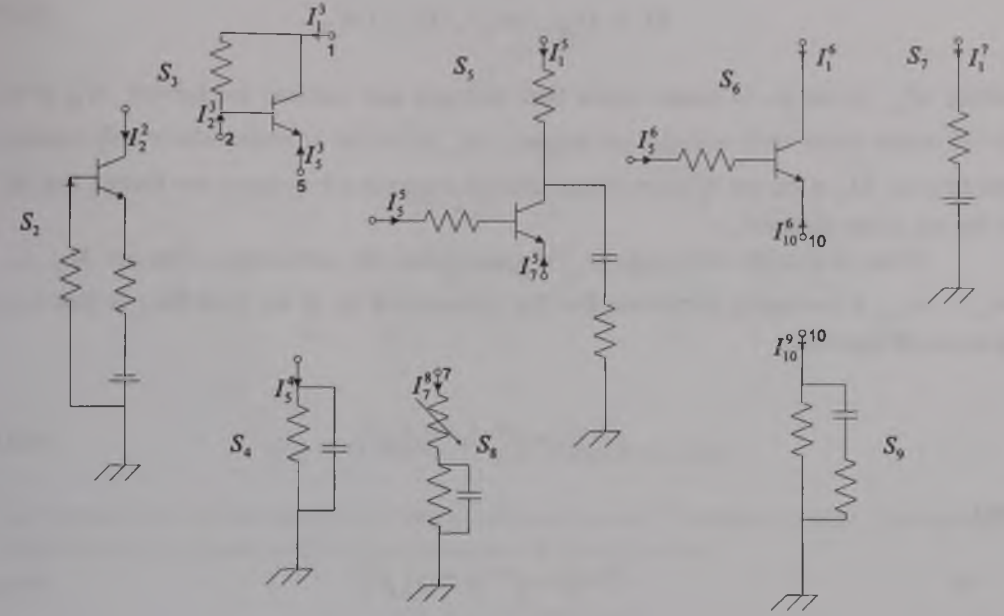


Fig. 4.2. Network decomposition of the video amplifier

The input-output relation for a subnetwork S_i, that is connected to the rest of the network by $m_i + 1$ external nodes, as shown in Fig. 4.3, with one of the nodes taken as the reference, is given by

$$i^{M_i}(t) = h^{M_i}\left(v^{M_i}(t), \phi_i\right) \tag{4.1}$$

where ϕ_i is the vector of the subnetwork parameters and the cardinality of $i^{M_i}(t)$, h^{M_i}, *and* $v^{M_i}(t)$ is m_i.

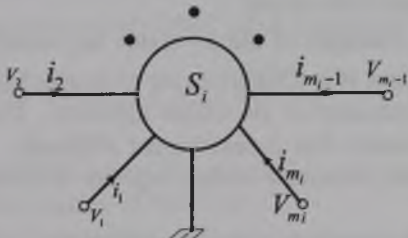


Fig. 4.3. Subnetwork S_i with $m_i + 1$ external nodes

Let us assume that the subnetwork S_j is connected, i.e., there exists a path between any two nodes of subnetwork S_j and that we cannot partition S_i into smaller uncoupled subnetworks using only the set of $m_i + 1$ external nodes. Let us divide the set of all measurement nodes into disjoint subsets

$$M_i = M_{i\alpha} \cup M_{i\beta} \cup M_{i\gamma} \cup M_{i\delta} \quad (4.2)$$

where $M_{i\alpha}$, is the set of nodes where both voltages and currents are known, $M_{i\beta}$ is the set of nodes where only voltages are known, $M_{i\gamma}$ is the set of nodes where only currents are known, $M_{i\delta}$ is the set of nodes where neither currents nor voltages are known and M_i is the set of the m_i nodes.

If the cardinality of the set $M_{i\alpha}$ is greater than the cardinality of the set $M_{i\delta}$, i.e., $m_{i\alpha} > m_{i\delta}$, a necessary condition for the subnetwork S_i to be fault-free is that both systems of equations

$$i^{M_{i\alpha}}(t) = h^{M_{i\alpha}}\left(v^{M_i}(t), \phi_i^0\right) \quad (4.3)$$

and

$$i^{M_{i\gamma}}(t) = h^{M_{i\gamma}}\left(v^{M_i}(t), \phi_i^0\right) \quad (4.4)$$

are consistent at any instance of time, where ϕ_i^0 is the vector of nominal parameter values.

When all the voltages of M_i are known and $m_{i\alpha}$ is greater than or equal to one, we can state the following lemma.

Lemma 4.1 [201]. Self-Testing Condition (STC)

A necessary and almost sufficient condition for a connected subnetwork S_i with $m_i + 1$ external nodes $m_{i\alpha} \geq 1$ and $m_{i\gamma} = m_{i\delta} = 0$ to be fault-free is that

$$i^{M_{i\alpha}}(t) - h^{M_{i\alpha}}\left(v^{M_i}(t), \phi_i^0\right) = 0 \quad (4.5)$$

Sufficiency of Lemma 4.1 is determined with probability equal to one for the tested circuit with the nominal parameter values.

Normally, only the voltages of the m_i nodes are measured. The currents $i^{M_{i\alpha}}(t)$ are not measured directly since it is difficult to do so in practice, with the exception when they represent the input excitation to the whole network. The application of KCL and topological relations overcomes this measurement difficulty. Instead, the currents are computed using the nominal parameter values together with the measured voltages, and by invoking KCL.

This result can be extended to a number of interconnected subnetworks. Let us assume that we have a set of k subnetworks $S_i, i \in J_t$, which are incident on a common node c as shown in Fig. 4.4.

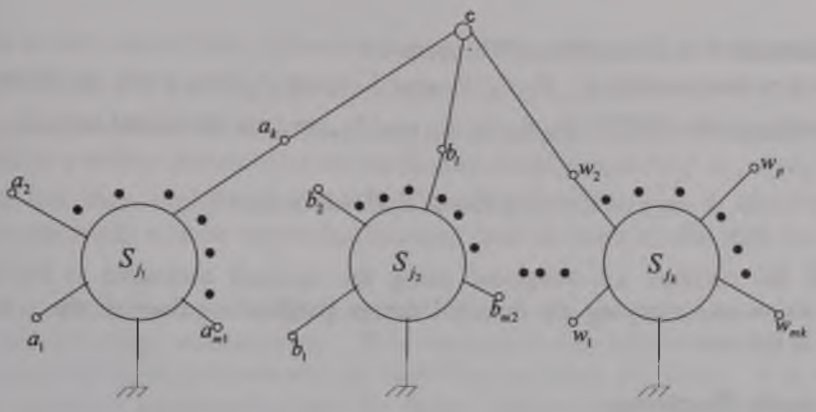


Fig. 4.4. k subnetworks incident at node c

Each subnetwork is assumed to be connected and has $m_i + 1$ external nodes. The current incident to the common node c from subnetwork S_i is given by

$$i_c^{M_i}(t) = h_c^{M_i}\left(v^{M_i}(t), \phi_i^0\right) \tag{4.6}$$

The following lemma applies to the CUT.

Lemma 4.2 [201]. Mutual-Testing Condition (MTC)

A necessary and almost sufficient condition for $S_i, i \in J_t$, to be fault-free is that

$$\sum_{i \in J_t} h_c^{M_i}\left(v^{M_i}(t), \phi_i^0\right) = 0, \qquad \forall t \tag{4.7}$$

i.e., the currents incident to the common node c computed by using the measured voltages and nominal parameter values must satisfy KCL.

Example 4.2: Illustration of Lemma 4.2.

Consider the decomposed network shown in Fig. 4.2. Subnetworks $S_3, S_5, S_6, and\ S_7$, are incident to node 1. So, according to the MTC of (4.7) they are fault-free if and only if

$$I_1^3 + I_1^5 + I_1^6 + I_1^7 = 0$$

where all currents are computed using the nominal parameter values and the measured voltages at the decomposition nodes of the CUT.

Theorem 4.1 [201]: Generalized -Mutual -Testing Condition (GMTC)

Let $E_i, i \in J_t$, denote some external nodes of the subnetwork S_i. Each subnetwork S_i is connected and has $m_i + 1$ nodes $E_i \subseteq M_i$. If the currents incident to E_i form a cut set, then a necessary and almost sufficient condition for these subnetworks to be fault-free is

$$\sum_{i \in J_t} \sum_{j \in E_i} h_j^{M_i}\left(v^{M_i}(t), \phi_i^0\right) = 0, \qquad \forall t \tag{4.8}$$

Example 4.3: Illustration of Theorem 4.1

The branches that connect $S_2, S_4, S_5, S_6,$ *and* S_7, with S_3 form a cut set shown in Fig. 4.1. According to the GMTC $S_2, S_4, S_5, S_6,$ *and* S_7, are fault free if and only if

$$I_2^2 + I_3^4 + I_1^5 + I_3^5 + I_1^6 + I_3^6 + I_1^7 = 0$$

where all the currents are computed using the voltages measured at the nodes of decomposition and applying the nominal design parameter values of the subnetworks involved in the test.

4.1.2. Faulty Regions

Network decomposition and the mutual tests can identify all faulty subnetworks. If needed, further diagnosis may be carried out to identify faulty element(s) or at least the faulty regions inside the faulty subnetworks. If a subnetwork contains only few elements, a search for the faulty elements inside the subnetwork (fault verification) is feasible, since the number of different combinations to be considered is small. The larger subnetworks need to be subdivided first to identify a smaller region that contains the faulty elements by using mutual test conditions. Subsequently, the fault verification technique is applied to this faulty region to find faulty elements.

Necessary and almost sufficient conditions for fault verification in linear and nonlinear networks have been developed in [280], [20], [244], [266], and [117].

Identification of Faulty Regions

An application of the self test condition (4.5) starts by partitioning the faulty subnetwork S_i into two smaller subnetworks S_j *and* S_k, such that $S_i = S_j \cup S_k$, as shown in Fig. 4.5 [244].

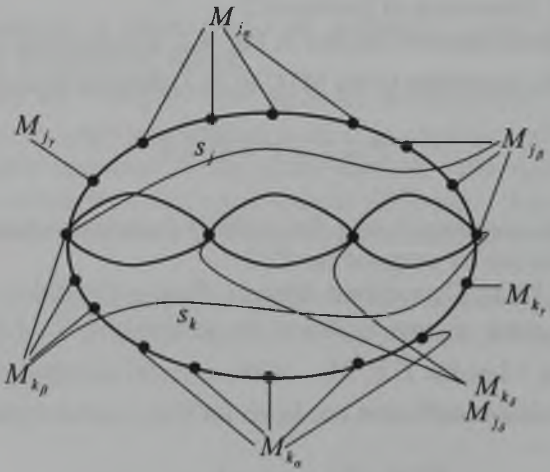


Fig. 4.5. Decomposition of network S_i into two subnetworks S_j *and* S_k

For at least one of these subnetworks, $m_{i\alpha} > m_{i\delta}$, where i= j or k, as appropriate. Utilizing condition (4.3) one can identify whether S_j *or* S_k, are fault-free or not. While checking these conditions, continue the binary partitioning process in the identified faulty region until no partition that satisfies the cardinality condition (namely $m_{i\alpha} > m_{i\delta}$) can be found. At this stage, apply one of the fault verification techniques to identify all the faulty elements inside a faulty region that in many cases is much smaller than the original subnetwork S_i.

The method developed in [201] assumed that all the decomposition nodes should be accessible to voltage measurements. This assumption was removed in [237]. For the network-under-test some subnetworks are fault-free and some are faulty. It is easier to locate the fault-free subnetworks than the faulty subnetworks. The first step in this method is to locate as many as possible fault-free subnetworks based on the following corollary. Let us define a common node as a node incident to several subnetworks in the decomposed network or a voltage measurement node.

Corollary 4.1 Suppose that a common node *c* connects *k* subnetworks S_i (i=1, 2, ..., k). If all the currents incident to the common node c computed by the measured voltages and the nominal parameter values satisfy the KCL equations, i.e.,

$$\sum_{i=1}^{k} I_c^i = 0 \quad \forall t \tag{4.9}$$

$$I_c^i = h_c^{M_i}\left(v^{M_i}(t),\ \phi_i^0\right) \tag{4.10}$$

where I_c^i is the current incident to node c from subnetwork S_i, M_i is the measurement set consisting of measurement nodes, v^{M_i} are the measured nodal voltages in subnetwork S_i, ϕ_i^0 are the nominal component values of subnetwork S_i, then all subnetworks S_i (i=1, 2, ..., k) are fault-free.

Such a common node is called a fault-free node. If equation (4.9) is not satisfied then at least one subnetwork S_i is faulty. In this corollary, all the decomposition nodes are the measurement nodes.

Suppose now that one decomposition node x in the subnetwork S_i is inaccessible, i.e., its nodal voltage V_x is unknown. Thus, the decomposed subnetwork topology remains unchanged, while the measurement set of S_i is changed by removing the node x. We can still compute I_c^i in (4.10) by changing the measurement set as above and the Corollary 4.1 remains valid to locate the fault-free subnetworks.

Corollary 4.2: Suppose a subnetwork S_i has two fault-free nodes *y* and *z* and one of the voltages V_x in this subnetwork is unknown. If the currents incident to these common nodes satisfy the KCL equations, i.e.,

$$\sum_{i=1}^{k_c} I_c^i = 0 \quad \forall t \quad c \in (y, z) \tag{4.11}$$

$$I_c^i = h_c^{M_i+X}\left(v^{M_i+X}(t),\ \phi_i^0\right) \quad (4.12)$$

where k_c is the number of subnetworks incident to common node c, then all subnetworks incident to nodes y and z are fault-free.

Here, the measurement set M_i is appended by the node x. Since there is only one unknown variable V_x in (4.12), V_x can be determined uniquely because we know that such solution exists in the network-under-test. As a generalization of Corollary 4.2 we can formulate the following lemma.

Lemma 4.3. Consider a subset of fault-free nodes in subnetworks S_i with p inaccessible decomposition nodes. All p inaccessible nodes are appended to the measurement set, leading to p unknown variables V_{x1}, V_{x2}, ..., V_{xp}. If there are m fault-free nodes and $m \geq p$, then by using m KCL equations

$$\sum_{i=1}^{k_x} I_x^i = 0 \quad x = 1, 2, ..., m \quad \forall t \quad (4.13)$$

$$I_x^i = h_x^{M_i+P}\left(v^{M_i+P}(t),\ \phi_i^0\right) \quad (4.14)$$

where k_x is the number of subnetworks incident to node x, we can determine all the voltages V_{x1}, V_{x2}, ..., V_{xp} and verify that all the subnetworks incident to fault-free nodes are fault-free. Using Corollaries 4.1 and 4.2 and Lemma 4.3 all fault free subnetworks can be verified and their internal voltages determined. The following example illustrates this process.

Example 4.4. Consider a faulty analog network decomposed at the measurement nodes into 4 subnetworks *S1, S2, S3,* and *S4* as illustrated on Fig. 4.6.

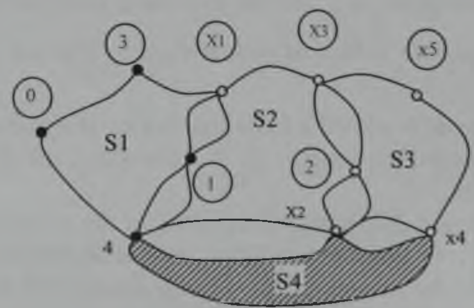


Fig. 4.6. Decomposed network for Example 4.4

Assume that *S4* (illustrated by the hashed area) is the only faulty subnetwork. Thus, nodes {*0, 1, 2, 3, x1, x3, x5*} are fault-free nodes with the node indexes circled in Fig. 4.6. Nodes 0 to 4 are the accessible nodes and nodes *x1* to *x5* are inaccessible. Apply (4.11) to nodes 0 and 3 to compute the currents I^{S1}:

$$I_{node\ 0}^{S1} = h_{node\ 0}^{M_{S1}+X1}\left(v^{M_{S1}+X1}(t),\ \phi_{S1}^0\right) \quad (4.15)$$

$$I^{S1}_{node\,3} = h^{M_{S1}+X1}_{node\,3}\left(v^{M_{S1}+X1}(t),\ \phi^{0}_{S1}\right) \tag{4.16}$$

where the measurement set is M_{s1} =[node0, node1, node3, node4]. Currents computed from (4.15) and (4.16) should equal to the external current excitations at these nodes. *S1* is concluded as fault-free by Corollary 4.2 and the internal voltage V_{x1} is calculated.

Subsequently by applying Lemma 4.3 to fault-free nodes in subnetworks S_2 and S_3 (nodes 1, 2, x1, x3, x5) with inaccessible decomposition nodes *x2, x3, x4,* and *x5,* 5 equations are obtained with 4 unknown voltages. Thus we can determine the unknown voltages V_{x2}, V_{x3}, V_{x4}, and V_{x5}, as well as verify that *S2* and *S3* are fault-free.

The results obtained by using Lemma 4.3 require knowledge (or a guess) of the location of fault-free nodes, since only KCL equations in these nodes can be used to formulate the verification equations. Instead of this ad-hoc approach, subnetwork verification can proceed efficiently by finding all ambiguity groups and solving the ambiguous equations.

4.2. Fault Location in Linear Networks

For the linear network test, the matrix description of the subnetworks greatly simplifies the computational effort needed for checking test conditions. Without loss of generality let us assume that the network is analyzed in the complex domain with voltages and currents represented by their Laplace transform variables.

Consider a subnetwork S_i which has m_i+1 external nodes, one of which is the reference node, and n_i internal nodes. The nodal equations are given by

$$\begin{bmatrix} Y_{M_iM_i} & Y_{M_iN_i} \\ Y_{N_iM_i} & Y_{N_iN_i} \end{bmatrix} \begin{bmatrix} V^{M_i} \\ V^{N_i} \end{bmatrix} = \begin{bmatrix} I^{M_{ii}} + I^{M_i}_g \\ I^{N_i}_g \end{bmatrix} \tag{4.17}$$

where

$$I^i_g = \begin{bmatrix} I^{M_i}_g \\ I^{N_i}_g \end{bmatrix} \tag{4.18}$$

defines the current sources associated with the subnetwork, V^{M_i} is the voltage vector of the external (measured) nodes, V^{N_i} is the voltage vector of the internal nodes, and I^{M_i} is the current input vector to the subnetwork from the outside environment applied through m_i external nodes. Eliminating equations at n_i internal nodes we get

$$I^{M_i} = -\left[I^{M_i}_g - Y_{M_iN_i}Y^{-1}_{N_iN_i}I^{N_i}_g\right] + \left[Y_{M_iM_i} - Y_{M_iN_i}Y^{-1}_{N_iN_i}Y_{N_iM_i}\right]V^{M_i} \tag{4.19}$$

or more compactly

$$I^{M_i} = -H_{M_i} I_g^i + Y_{M_i} V^{M_i} \tag{4.20}$$

where

$$H^{M_i} = -\left[1 - Y_{M_i N_i} Y_{N_i N_i}^{-1}\right] \tag{4.21}$$

and

$$Y_{M_i} = \left[Y_{M_i M_i} - Y_{M_i N_i} Y_{N_i N_i}^{-1} Y_{N_i M_i}\right] \tag{4.22}$$

and 1 is a unit matrix of order m_i. Equation (4.20) describes the input-output relation of the subnetwork and consistency of this relation is tested in Theorem 4.1.

Testing of a large network uses the hierarchical decomposition illustrated by a **decomposition tree** as illustrated in Fig. 4.7.

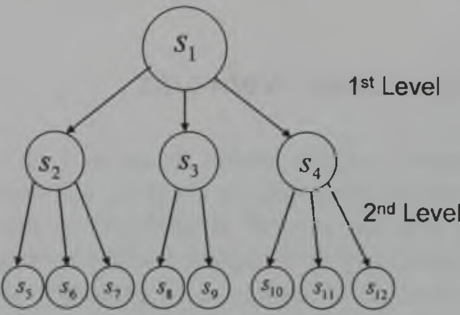


Fig. 4.7. Illustration of the hierarchical decomposition of a network

This figure shows the decomposition tree of the video amplifier circuit from Fig. 4.1.

Fault location of the CUT begins by considering self-testing and mutual testing conditions for the subnetworks at the first level of decomposition. If a subnetwork is declared fault-free no further partitioning of such subnetwork is needed. Subnetworks that did not receive a fault-free status are decomposed further using the measurement nodes. Detailed algorithm for location of faulty subnetworks is described in 201.

4.2.1. Location of Faulty Elements

Representing the change in a faulty element from its nominal value by a current source across that element, we may write (4.20) as

$$I^{M_i} = H_{M_i} I_g^i + Y_{M_i} V^{M_i} + H_{M_i F} I^F \tag{4.23}$$

where I^F represents the faulty current sources and $H_{M_i F}$ is computed using the nominal parameter values of the subnetwork and defines the transfer relation between I^{M_i} and I^F.

Normally, it is computed using the adjoint network concept as discussed in 8. Considering a subset of (4.23) that corresponds to $m_{i\alpha}$ known currents we have

$$H_{M_{i\alpha}F} I^F = I^{M_{i\alpha}} - H_{M_{i\alpha}} I_g^i - Y_{M_{i\alpha}} V^{M_i} \tag{4.24}$$

where only the rows $M_{i\alpha}$ are considered in the matrices $H_{M_{i\alpha}F}, H_{M_{i\alpha}}, and\ Y_{M_{i\alpha}}$. If $m_{i\alpha}$ is greater than or equal to $f+1$, then the system of equations (4.24) is overdetermined.

A necessary condition for F to contain the faulty set is that (4.24) is a consistent system of equations. The faulty set F is unique if [244]

$$Rank\lfloor H_{M_{i\alpha}F} \quad H_{M_{i\alpha}x} \rfloor = f+1 \qquad for \quad \forall x \notin F \tag{4.25}$$

where $H_{M_{i\alpha}x}$ represents a transfer vector from a current source across an element x in the subnetwork to the measurement nodes $M_{i\alpha}$, and (4.25) is considered for all elements x in the subnetwork other than the elements in the faulty set F.

4.2.2. Fault Location by Nodal Analysis

Nodal equations are convenient for multiple fault location in analog circuits [244]. Assume that a network S has $(n+1)$ nodes, m of them accessible for excitation and measurement, with f faulty nodes and $f < m$. The nodal equations of S are used to write an overdetermined system of equations. Consistency of this system is a necessary condition to identify the all faulty nodes. For the nominal values of the network elements the nodal equations which describe the network S have the following form

$$I_n = Y_n V_n \tag{4.26}$$

where Y_n, is the nodal admittance matrix, V_n , is the vector of nodal voltages with respect to a selected reference node r, and I_n is the vector of nodal currents. If, as a result of faults, S is perturbed to $(S + \Delta S)$ then maintaining the same excitations we will get

$$I_n = (Y_n + \Delta Y_n)(V_n + \Delta V_n). \tag{4.27}$$

Subtracting (4.26) from (4.27) yields

$$\Delta I_n = Y_n \Delta V_n = -\Delta Y_n \hat{V}_n \,. \tag{4.28}$$

from which we have

$$\Delta V_n = Y_n^{-1} \Delta I_n \tag{4.29}$$

Applying (4.29) to the measurement nodes only, we get the fault diagnosis equations as

$$\Delta V_n^M = Z_{MN} \Delta I_n \tag{4.30}$$

where

$$Y_n^{-1} = \begin{bmatrix} Z_{MN} \\ Z_{N-M,N} \end{bmatrix} \tag{4.31}$$

Under the assumption that the network has at most f faults, fault diagnosis equations are

$$\Delta V_n^M = \begin{bmatrix} Z_{MF} & Z_{M,N-F} \end{bmatrix} \begin{bmatrix} \Delta I_n^F \\ 0 \end{bmatrix} \tag{4.32}$$

from which we will have

$$\Delta V_n^M = Z_{MF} \, \Delta I_n^F \tag{4.33}$$

For $f < m$ (4.33) is an overdetermined system of equations and its solution leads to the following testability requirement

Lemma 4.4 [11]. A necessary condition for isolating the faulty nodes through single frequency nodal voltage measurements is that the following equation is satisfied

$$\left[Z_{MF} \left(Z_{MF}^T Z_{MF} \right)^{-1} Z_{MF}^T \quad -1 \right] \Delta V_n^M = 0 \tag{4.34}$$

This equation is satisfied regardless of the values of the faulty elements and depends on their location only. Considering a specific candidate for the set of faulty nodes F, the following condition must be satisfied:

$$Rank(Z_{MX}) = f + 1 \tag{4.35}$$

where X refers to the set of columns of the matrix Z_{MN} such that

$$X = F \cup \{y\}, \qquad y \in N - F \tag{4.36}$$

The above condition is equivalent to the existence of a square nonsingular submatrix Z_{MX} of order $(f+1)$. Thus an equivalent testability condition can be stated in terms of the nodal admittance matrix using the following equivalence relation

$$\det Z_{EX} \neq 0 \quad \Leftrightarrow \quad \det Y(X|E) \neq 0 \tag{4.37}$$

where $Y(X|E)$ is the matrix obtained from Y_n by removing X rows and E columns, and E is a subset of M with cardinality $f + 1$. By using Lemma 4.4 we can establish topological network testability conditions as described in the next section.

4.2.3. Topological Conditions for Node Fault Diagnosis

This section presents topological conditions to determine network testability based on the results obtained in [21], [235], [248]. Topological criterion is developed to check nonsingularity of the matrix $Y(X|E)$ with X containing specific candidates for the set F of faulty nodes. The advantage of topological conditions is that they are formulated regardless of parameter tolerances or their numerical values. However, for large-scale networks, a direct approach using these conditions could be inefficient. In such cases the decomposition technique [201] may be used and topological conditions applied to the decomposed network. Topological testability condition, which capitalizes on (4.37) uses Coates' flow graph representation of a CUT. It checks nonsingularity of the matrix $Y(X|E)$ with X containing specific candidates for the set F of faulty nodes.

Let G_c denote a directed Coates' graph of the network S, and let $G_c(X,E)$ be the graph obtained from G_c after deleting all the edges incoming to nodes X and all the edges outgoing from nodes E. Let W denote a set of node pairs of G_c as in Definition 3.2

$$W = \{(v_1,r_1),(v_2,r_2),\ldots,(v_k,r_k),\}, \quad v_i \neq v_j, v_i \neq r_j, r_i \neq r_j, \quad for \quad i \neq j$$

such that $v_i \in X$ and $r_i \in E$.

Theorem 4.2 [248]. Existence of at least one $(f+1)$-connection c_W in the graph $G_c(X,E)$ is a necessary and almost sufficient condition for $\det Y(X|E) \neq 0$

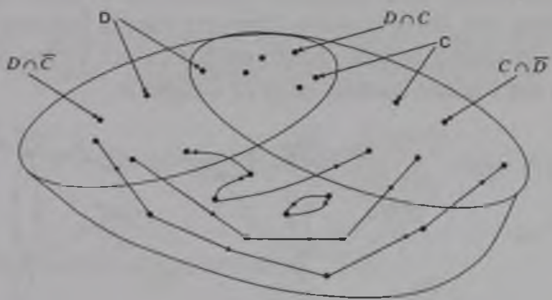


Fig. 4.8. An example of the required 3-connection

A similar topological condition was obtained on the basis of the conjugated graph representation of the given network [235].

Theorem 4.3 [235]. A necessary and almost sufficient condition for $\det Y(X|E) \neq 0$ is the existence of a common tree of both graph $G_1(X \cup \{r\})$ and $G_2(E \cup \{r\})$, where r is the reference node.

Topological testing was extended in [247] to remote subnetworks inaccessible to direct measurements. Assume that $\bar{Y}$ is an admittance matrix of the remote inaccessible subnetwork to be identified through the remote node measurements. Unknown elements of $\bar{Y}$ will be identified through identification of reduced cut-sets (see [247]).

Assume that independent excitations are applied to the subset of nodes A (called injection nodes). Nodes A can be chosen from a remote inaccessible subnetwork and they must be a subset of the external nodes of this subnetwork.

Corollary 4.3 [247]. If independent excitations which are applied to the subset of nodes $A \subset N$ are sufficient for the identification of all elements of $\overline{Y}$, then

$$\forall E_j \quad \exists B_j \subset A \quad and \quad \exists X_j : \quad \det \overline{I}^T_{B_j X_j} \neq 0 \quad and \quad \det \overline{Y}_{\overline{X}_j \overline{E}_j} \neq 0$$

where $card\ B_j = card\ E_j = card\ X_j \quad and \quad \overline{X} = N - X, \quad \overline{E} = N - E$.

Corollary 4.4 [247]. It follows from Corollary 1, that to identify all elements of $\overline{Y}$ one must find a set B_j such that, after deleting all the edges outgoing from nodes E_j, and after deleting all the edges incoming to nodes X_j, there are no isolated nodes in the $\overline{E_j \cap X_j}$.

To help testing the remote networks a special technique based on the idea of topological corner is developed in [248].

Definition 4.1 [248]. A node is said to be a corner if there exists a complete subgraph containing all the edges incoming to the node as well as the edges having the same weight as any of the incoming ones.

The order of this complete subgraph is not defined. In particular, it may be a complete graph of zero order – see Fig. 4.9(a), in which vertex v is a corner. Also, there may exist edges outgoing from a corner to other parts of the network graph which are not part of the complete subgraph – see Fig. 4.9 (b), in which both vertices labeled v are corners. The remaining two vertices are not corners simply because complete subgraphs that contain all edges incoming to these nodes do not exist. Vertex x in Fig. 4.9(c) is not a corner since it does not contain another edge of weight α.

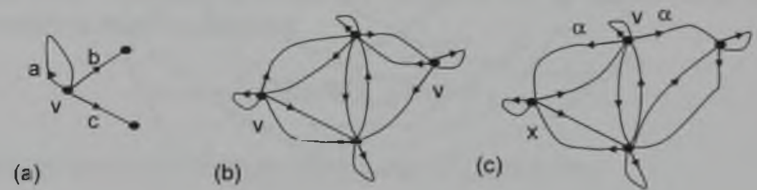


Fig. 4.9. Examples of corners are denoted by v

Theorem 4.4 [235]. All the corners must be the injection nodes. Thus if a vertex is not a corner, it follows that we do not have to provide independent excitations at this node to identify all the elements of $\overline{Y}$. The number of corners influences the minimal cardinality of A. An efficient heuristic algorithm which can be adopted to find the injection nodes was presented in [248]. The algorithm localizes injection nodes in such a way that there exists a set of separate paths from the injection nodes to the nodes of each reduced cut-set as illustrated in Fig. 4.10.

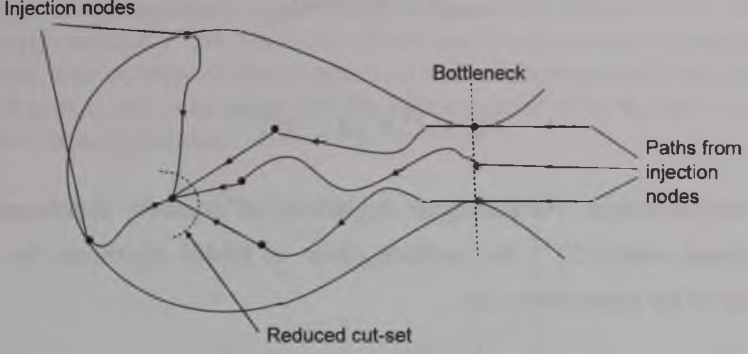


Fig. 4.10. Paths required form injection nodes to a reduced cut-set

4.2.4. Parameter Tolerances

The actual parameter values can deviate from their nominal values within prescribed tolerance bounds. Thus in practice, we face the situation that Lemmas 4.1, 4.2 and Theorem 4.1 are not satisfied to the required degree of machine accuracy. Taking the parameter tolerances into consideration we may rewrite condition (4.5) as

$$i^{M_{ia}}(t) - h^{M_{ia}}\left(v^{M_i}(t), \phi_i^0 + \Delta\phi_i\right) = 0 \tag{4.38}$$

where $\Delta\phi_i = \left[\Delta\phi_{i1}\ \Delta\phi_{i2} \ldots \Delta\phi_{ip}\right]^T$ defines the tolerance changes in the p elements of the tested subnetwork. For small tolerances the first-order approximation can be used to describe the network response. Accordingly, we may write (4.38) as

$$i^{M_{ia}}(t) - h^{M_{ia}}\left(v^{M_i}(t), \phi_i^0\right) = \sum_{j=1}^{p} \frac{\partial h^{M_{ia}}}{\partial \phi_{ij}} \Delta\phi_{ij} \tag{4.39}$$

where the partial derivatives are calculated at ϕ_i^0. Let

$$\Delta i^{M_{ia}}(t) = i^{M_{ia}}(t) - h^{M_{ia}}\left(v^{M_i}(t), \phi_i^0\right) = B_i \Delta\phi_i \tag{4.40}$$

where

$$B_i = \begin{bmatrix} \frac{\partial h^{M_{ia}}}{\partial \phi_{i1}} & \frac{\partial h^{M_{ia}}}{\partial \phi_{i2}} & \cdots & \frac{\partial h^{M_{ia}}}{\partial \phi_{ip}} \end{bmatrix}. \tag{4.41}$$

At a given time *t* equation (4.40) is an underdetermined system of linear equations in the variable $\Delta\phi_i^0$. The weighted least squares solution of (4.40) is given by [167],

$$\Delta\phi_i^0 = B_i^+ \Delta i^{M_{i\alpha}}(t_0) \tag{4.42}$$

where

$$B_i^+ = C_i B_i^T \left[B_i C_i B_i^T\right]^{-1} \tag{4.43}$$

and C_i is a weight matrix. For parameter deviations $\Delta\phi_i^0$ normally distributed with mean 0 and covariance matrix C_i^{-1}, the solution given in (4.42) represents the conditional expected value of the parameters $\Delta\phi_i$,

$$\Delta\phi_i = E\left[\Delta\phi_i \mid \Delta i^{M_{i\alpha}}(t_0)\right] \tag{4.44}$$

where E denotes the expectation. Moreover, the solution is a minimum in the weighted least squares sense. So $\Delta\phi_i$ is the solution of

$$\text{minimize} \quad \Delta\phi_i^T C_i^{-1} \Delta\phi_i \tag{4.45}$$

subject to

$$B_i \Delta\phi_i = \Delta i^{M_{i\alpha}}(t_0) \tag{4.46}$$

Using the probabilistic expectation (4.44), we can measure to what degree (4.5) is satisfied under the statistical variations caused by the parameter tolerances. If any deviation $\Delta\phi_i$ obtained from (4.42) significantly exceeds the parameter tolerance we consider that the subcircuit is faulty. The weight matrix C_i in (4.43) provides the possibility of considering existing correlations between the network parameters.

4.2.5. Multiport Approach to Fault Location

Multiport representation is frequently used in description of high-speed VLSI interconnect, and microwave systems. Fault diagnosis in systems with a hybrid multiport description requires different testing approach. Early work on this subject was done by Biernacki and Bandler who formulated necessary conditions on the set of equations used for fault identification [20]. The problem of consistency of the chosen set of equations used for multiport fault identification is discussed by the author in [236]. The paper also discusses topological restrictions of the multiport fault location. In particular implications of block dependency on the network testability were demonstrated. The paper helps to understand the limitations of using the multiport method for fault location. Some of the results obtained in [236] are presented here for reference.

Assume for simplicity that the linear network under investigation contains one-port elements and controlled sources only. Assume also that the network has $n + 1$ nodes, e elements and that f of these elements are faulty. To identify all the faults let us measure m voltages in the network, with $m > f$.

Parameter changes from their nominal values can be represented by current sources in parallel with the nominal elements (for one ports and controlled current

sources) or by voltage sources in series with the nominal elements (for one ports and controlled voltage sources) - see Fig. 4.11. Faults can be represented as loads of the *(m + f)*-port network that contain all the elements of the original network with their nominal values - see Fig. 4.11 b). Assuming that the hybrid matrix H of the *(m + f)*-port exists we obtain the following relation

$$\begin{bmatrix} V^M \\ R^F \end{bmatrix} = \begin{bmatrix} H_{MM} & H_{MF} \\ H_{FM} & H_{FF} \end{bmatrix} \begin{bmatrix} I^M \\ S^F \end{bmatrix} \tag{4.47}$$

where

$$S^F = \begin{bmatrix} I^{F_1} \\ V^{F_2} \end{bmatrix}, \qquad R^F = \begin{bmatrix} V^{F_1} \\ I^{F_2} \end{bmatrix} \tag{4.48}$$

are source and response vectors at fault ports,

$$V^M = \begin{bmatrix} V_1^M & \dots & V_m^M \end{bmatrix}^T, \qquad I^M = \begin{bmatrix} I_1^M & \dots & I_m^M \end{bmatrix}^T \tag{4.49}$$

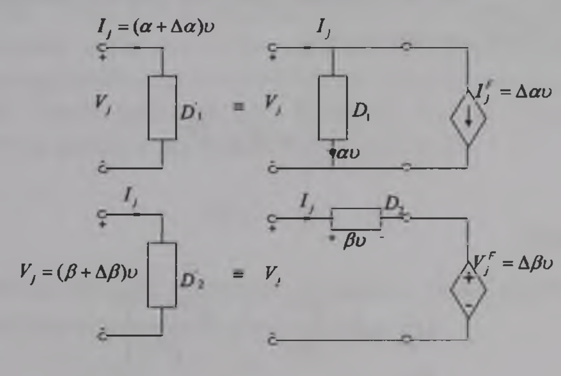
are measured voltages and currents,

$$V^{F_1} = \begin{bmatrix} V_1^F & \dots & V_{f_1}^F \end{bmatrix}^T, \qquad V^{F_2} = \begin{bmatrix} V_{f_1+1}^F & \dots & V_f^F \end{bmatrix}^T \tag{4.50}$$

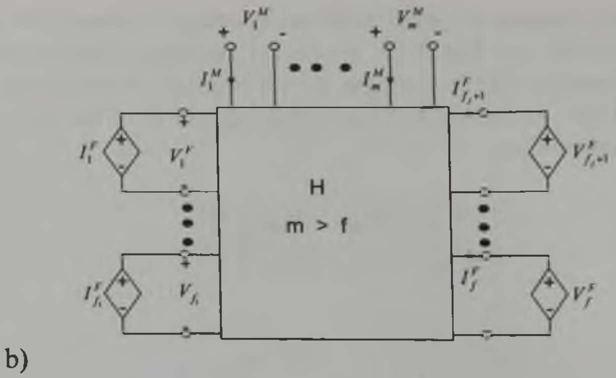
are voltages at fault ports, and

$$I^{F_1} = \begin{bmatrix} I_1^F & \dots & I_{f_1}^F \end{bmatrix}^T, \qquad I^{F_2} = \begin{bmatrix} I_{f_1+1}^F & \dots & I_f^F \end{bmatrix}^T \tag{4.51}$$

are currents flowing through the fault ports. A change in the measurement voltage vector $\Delta V^M = V^M - V^{M0}$ in the multiport testing can be expressed as $\Delta V^M = H_{MF}\, S^F$.



a)



b)

Fig. 4.11. Representing changes in parameter values. (a) D_1 denotes a one-port or controlled current source, D_2 denotes a one-port or controlled voltage source, (b) V_i^M denotes the controlling voltage or current

Assuming that H_{MF} is of full column rank, the necessary condition for the set F of network elements to contain all the faults, is given by the relation

$$\left(\overline{H}_{MF} - 1\right)\Delta V^M = 0, \tag{4.52}$$

where

$$\overline{H}_{MF} = H_{MF}\left(H_{MF}^T H_{MF}\right)^{-1} H_{MF}^T . \tag{4.53}$$

H_{MF} is called the test matrix. Equation (4.52) allows checking if the set F contains all the network faults on the basis of the measured voltage change vector ΔV^M.

The test matrix H_{MF} can be designed using the adjoint network analysis. For the adjoint network with $\hat{S}^F = 0$ we obtain [56]

$$\begin{bmatrix} \hat{V}^{F_1} \\ -\hat{I}^{F_2} \end{bmatrix} = H_{MF}^T \hat{I}^M \tag{4.54}$$

that yields the test matrix

$$H_{MF}^T = \begin{bmatrix} \hat{V}^{F_1 1} & \dots & \hat{V}^{F_1 m} \\ -\hat{I}^{F_2 1} & \dots & -\hat{I}^{F_2 m} \end{bmatrix} \tag{4.55}$$

Having calculated all the voltages and the currents in all adjoint network elements with different current excitations $\hat{I}^{M_i}$, we can obtain the test matrix H_{MF} corresponding to any set F of faulty elements.

Relation (4.52) is a necessary but not sufficient condition for F to contain all the faults in the network. One of the important indicators of the circuit testatability is the block dependency of two overdetermined systems of equations. The following lemma relates the network topology to the consistency conditions of the multiport test matrix.

Lemma 4.5 [236]. If the set of faulty elements contains a subset consisting of either

(a) a circuit formed by one-ports, controlled current sources or currents that control faulty voltage or current sources;

(b) a cutset formed by one-ports, controlled voltage sources or voltages that control faulty voltage or current sources;

then the test matrix H_{MF} of contains linearly dependent columns.

A corollary follows immediately from Lemma 4.4.

Corollary 4.5 [236]

If the set of faulty elements contains any subset defined in Lemma 4.4, then the multiport method cannot be used for fault location.

Although Corollary 4.5 is of a negative nature, it formulates precise topological restriction on the multiport testing method, thus it becomes a constructive extension of Theorem 2 given by Biernacki and Bandler [20]. Linear dependency of both rows and columns affect the efficiency of multiple fault location. It is evident that the number of independent voltage measurements should at least be equal to the number of columns of the test matrix to obtain the full column rank.

Corollary 4.6 [236]

The maximum number of faults that can be located by the multiport fault location method is equal to the number of nodes in the network minus two.

Linear dependency between the columns of the test matrix leads to the problem of ambiguity groups addressed in a separate section.

4.3. Sensitivity Approach

Sensitivity approach is a popular way to describe the effect of small deviations on the response function. Thus it was a natural choice for analog testing, under the same assumption of the limited variations of the parameter values. This section shows how the sensitivity based analog testing can be aided by the network topology.

In sensitivity based approach test functions can be defined as the vector of differences between the nominal and measured responses as

$$\phi(p) = x(p) - x_M, \tag{4.56}$$

where $x(p)$ is obtained through computer simulation based on the circuit model with element values p, and the corresponding test equation is

$$s\,\Delta p = \Delta x, \tag{4.57}$$

where Δx is the vector of voltage deviations from the nominal values, and s is the sensitivity matrix($s = \partial x / \partial p$).

We assume that the system parameters p are close to their nominal values. Each linear element is identified through a single parameter p_i. Nonlinear elements have their characteristics described through several parameters p_i (e.g., $i_b = p_0 \exp(p_1 v_b) + p_2$), so one nonlinear element may require identification of more than one parameter value.

Sensitivities can be efficiently calculated by the adjoint technique. Let us discuss sensitivities in time domain testing. A similar approach is used for DC testing, and frequency domain testing. First, let us consider a nonlinear circuit described by the set of algebraic differential equations in an implicit form:

$$f(\dot{x}, x, p, t) = 0, \tag{4.58}$$

where f is the vector of circuit functions, x is the vector of circuit variables, $\dot{x}$ is the vector of time-derivatives of x, p is the vector of circuit parameters and t is time. The system equation (4.58) can be obtained using any general formulation technique such as modified nodal formulation, sparse tableau or hybrid description.

It is assumed that at $t = 0$, the initial conditions are consistent and (4.58) has a unique solution. The time interval $(0, \tau)$ is divided into the discrete points $(0, t_1, \ldots, t_j, \ldots, \tau)$. At a certain time point t_j, equation (4.58) can be solved iteratively with the Jacobian matrix M_j equal to

$$M_j = \frac{df_j}{dx_j} = \frac{\partial f_j}{\partial \dot{x}_j}\frac{\partial \dot{x}_j}{\partial x_j} + \frac{\partial f_j}{\partial x_j}, \tag{4.59}$$

where $\frac{\partial \dot{x}_i}{\partial x_j}$ can be obtained by using any numerical integration method. In this presentation let us assume that the backwards differentiation formula (BDF) was selected as a numerical integration method. However, similar results can be derived for other numerical integration methods. Using BDF $\dot{x}_j$ can be obtained from the solution vectors in the past integration steps as follows:

$$\dot{x}_j = \sum_{l=0}^{k_j} a_l \, x_{j-l} \,. \tag{4.60}$$

Derivatives $\frac{\partial f_j}{\partial \dot{x}_j}$ and $\frac{\partial f_j}{\partial x_j}$ are calculated using current values of the solution vector x. Since these values change from the iteration to iteration, the Jacobian matrix M_j has to be evaluated and factorized at each iteration. The equation (4.58) is solved using

$$M_j \Delta x_j = -f_j, \tag{4.61}$$

To obtain sensitivities with respect to p, let us differentiate (4.58) and get

$$\frac{\partial f_j}{\partial \dot{x}_j}\dot{s}_j + \frac{\partial f_j}{\partial x_j}s_j = -\frac{\partial f_j}{\partial p}, \tag{4.62}$$

Substituting BDF formula (4.60) for $\dot{s}_j$ into (4.62), we have

$$\left[\frac{\partial f_j}{\partial \dot{x}_j}a_0 + \frac{\partial f_j}{\partial x_j}\right]s_j = -\frac{\partial f_j}{\partial p} - \frac{\partial f_j}{\partial \dot{x}_j}\sum_{l=1}^{k_j} a_l s_{j-l}, \tag{4.63}$$

where the system matrix is identical to the Jacobian matrix M_j. Rewrite (4.63) as

$$M_j s_j = B_j, \tag{4.64}$$

where B_j is a coefficient matrix which includes all terms on the right hand side of (4.63). Note that all these terms were evaluated at the previous time steps, so they are readily available for computations.

The sensitivity approach handles a broad category of networks and testing situations. Specialized formulas have been developed for this approach when applied to linear and nonlinear networks or networks with reactive elements and switches. Different test equations are derived depending upon the type of measured responses such as time domain response, frequency domain or harmonics of a periodic response.

However, the method shows some serious drawbacks when applied to large scale circuits. The first drawback is its low speed. In order to derive the sensitivity matrix, a circuit must be analyzed using simulators based on Newton's method, a sparse matrix technique, and numerical integration. Computation time limits the size of circuits that can be tested using the sensitivity approach to a few hundred elements.

Another drawback is low accuracy of the sensitivity method. In addition to errors caused by the first order approximation, the method is very sensitive to inaccuracies in the circuit model and in the numerical integration techniques, parasitics introduced by the test equipment and errors of time synchronization. Serious problems are associated with finding the rank of the sensitivity matrix, testability factors, and ambiguity groups.

Finally, the sensitivity method requires large computer memory. Transfer functions are sensitive to variations of all the network parameters. This forces full density of the sensitivity matrix and makes numerical calculations very expensive in large networks.

4.3.1. Decomposition Approach to Sensitivity Based Testing

Network decomposition is a very effective way to reduce computations in the analysis of large networks [251]. A special version of the decomposition approach proposed in [201] was used for fault location in large-scale networks, in which faulty elements were located within separable regions of the network with the remaining part of

the network fault-free. Decomposition of a network into smaller subnetworks facilitated testing by localizing the effect of faults to the specific subnetworks.

The idea of using a decomposition approach for parameter identification stemmed from the fault location approach mentioned above. However, the two approaches are quite different in the form of equations obtained and the processing steps required.

A direct decomposition of a parameter identification problem is difficult. This difficulty is caused by a high density of the test matrix based on the sensitivity approach. In order to resolve this difficulty, a different set of test functions is used. In the proposed approach, the measurements are taken at the selected nodes and the circuit is decomposed at these nodes. Usually, the selected nodes are limited to the external terminals or, in case of a large system, to the nodes between different subsystems [234]. By using circuit decomposition the complexity of the obtained test equations is reduced and the corresponding test matrix is sparse. This results in savings in evaluation of both the solution vector x, and the vector of parameter deviations Δp.

Decomposed Network Sensitivity Analysis

Let N be the network under test. The nodal decomposition is used to partition the network N into k subnetworks at accessible nodes (see Fig. 2.2). There must be no mutual coupling between any two subnetworks. Assume that all decomposition nodes can be accessed for measurements. The measured or external nodes are denoted by m and the remaining ones (internal nodes) are denoted by i. After all the voltages have been measured, the external variables x^m have known values. Using these values in a network analysis corresponds to adding voltage sources connected at the network terminals. In the iterative solution of the system equations we get

$$\Delta x^m = 0, \tag{4.65}$$

and since the measured voltages do not vary with the assumed or computed parameter values, sensitivities of the external variables w.r.t. the parameters are zero.

$$s^m = \frac{\partial x^m}{\partial p} = 0, \tag{4.66}$$

Simplifications in system analysis which follow from (4.65) and (4.66) make computations of the solution vector x^i and the sensitivities s^i of the internal variables much easier than in a direct approach. When the circuit is decomposed into a number of small subnetworks, computations of x^i and s^i can be implemented in parallel on separate processors within each subnetwork.

According to (4.65), the deviations of system variables are nonzero at the internal part only

$$\Delta x = \begin{bmatrix} \Delta x^i \\ \Delta x^m \end{bmatrix} = \begin{bmatrix} \Delta x^i \\ 0 \end{bmatrix}, \tag{4.67}$$

so that only the internal system equations f^i are used to obtain the internal variables Δx^i. As a result one can simplify (4.61) by eliminating rows and columns of M_j which

correspond to the measured circuit variables, renaming the obtained submatrix by M_j^i. Then (4.61) can be replaced by

$$M_j^i \Delta x_j^i = -f_j^i, \tag{4.68}$$

and the internal variables are evaluated using the measured variables and the nominal parameter values.

To observe the effect of this simplification on the analysis and testing process, let us consider the structure of M_j and M_j^i, for hierarchical partition at the measurement nodes. As shown on Fig. 4.12 M_j has a bordered block diagonal structure, where the block diagonal part of M_j corresponds to the internal circuit variables.

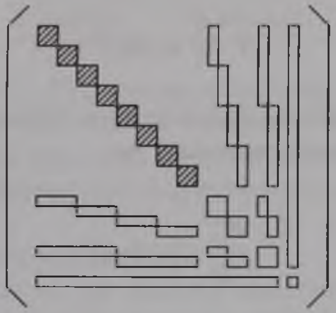


Fig. 4.12. Bordered block diagonal sensitivity matrix of three-level decomposition

As a direct result of this bordered block diagonal structure of M_j, the submatrix M_j^i is block diagonal and (4.68) becomes:

$$\begin{bmatrix} M_j^{i1} & & & & & \\ & M_j^{i2} & & 0 & & \\ & & \cdots & & & \\ & & & M_j^{i\lambda} & & \\ & 0 & & & \cdots & \\ & & & & & M_j^{ik} \end{bmatrix} \begin{bmatrix} \Delta x_j^{i1} \\ \Delta x_j^{i2} \\ \cdots \\ \Delta x_j^{i\lambda} \\ \cdots \\ \Delta x_j^{ik} \end{bmatrix} = - \begin{bmatrix} f^{i1} \\ f^{i1} \\ \cdots \\ f^{i1} \\ \cdots \\ f^{i1} \end{bmatrix}_j . \tag{4.69}$$

This equation can be solved independently in each subnetwork. Details of this method and numerical results of the decomposed network testing are presented in [245].

Subnetwork Testing

After the network analysis is completed, the system equations (4.58) will be satisfied at the internal points used in all iterations, but in general they will not be satisfied at the partition points. This results from a mismatch between the nominal parameters p^0 for which iterations (4.68) were performed, and the actual parameters p^*, for which the measurements were taken. Let us define the external system functions as the **test functions**

$$\phi(p) = f_j^m(\dot{x}_j, x_j, p) \tag{4.70}$$

The test functions are in general nonlinear functions of all design parameters p. Since the test functions are built using the Kirchhoff law at the partition points, they must satisfy $\phi(p^*) = 0$ for the actual and unknown parameters p^*. By using the Taylor expansion about the nominal point p^0 we will obtain the test equations that can be used to identify faulty parameters.

$$\phi(p^*) \cong \phi(p^0) + \left.\frac{\partial \phi}{\partial p}\right|_{p=p^0} (p^* - p^0) + \ldots = \phi(p^0) + T\,\Delta p \tag{4.71}$$

which leads to the following test equations

$$T\,\Delta p = -\phi(p^0) \tag{4.72}$$

Thus from (4.70) we have the following form of the test equations that is useful to evaluate the change in the network parameters Δp

$$\frac{df_j^m}{dp}\,\Delta p = -f_j^m \tag{4.73}$$

The test matrix $T_j = \frac{df_j^m}{dp}$ is evaluated by finding derivatives of $\phi(p)$ w.r.t. parameters p.

$$T_j = \frac{\partial f_j^m}{\partial \dot{x}_j} \sum_{l=1}^{k_j} a_l s_{j-l} + \frac{\partial f_j^m}{\partial x_j^i} s_j^i + \frac{\partial f_j^m}{\partial p} \tag{4.74}$$

Individual subcircuits will have a similar form of their corresponding test matrices as shown in (4.74).

Matrix M_j^i is block diagonal, while matrix T has the bordered block diagonal structure as shown in Fig. 4.12 . There is no overlap between any two blocks in the block diagonal matrix, so equations (4.68) can be solved independently in each block. In the bordered block diagonal matrix, blocks which correspond to different subnetworks overlap. Such equations are solved very efficiently using the sparse matrix technique.

As was demonstrated in [234] decomposition based testing method is much more efficient than the direct sensitivity based approach in network testing. The computational time in the decomposition method can be estimated as $O(n^\alpha)$, $1 < \alpha < 1.06$, while the direct sensitivity based method requires time proportional to $O(n^\beta)$, $2.7 < \beta < 2.9$, where n is the number of circuit nodes (see Fig. 4.13).

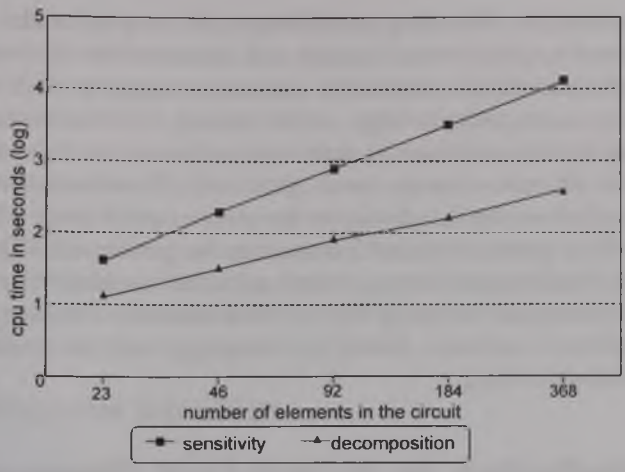


Fig. 4.13. CPU time at testing stage [234]

While similar savings were known to exist in the sparse matrix analysis of electronic circuits, this was the first result in which test equations were obtained in the sparse, block diagonal matrix format, resulting in high test efficiency.

Effect of Tolerances and Catastrophic Faults

When a model of CUT is not accurate (for example due to the presence of parasitic elements and tolerances in the circuit) or when there are changes in the circuit topology caused by catastrophic faults (open- or short-circuits), accuracy of the parameter evaluation in sensitivity approach is adversely affected.

To address this issue the effect of modeling errors on the accuracy of the parameter values obtained both by decomposition sensitivity and direct sensitivity based testing methods was analyzed in two separate studies [234]. In the first study circuit element values were randomly altered from their nominal values by adding small tolerance level deviations. While test results are sensitive to element location within the tested network and the network topology, the decomposition approach to sensitivity based testing resulted in the parameter estimation errors on the level of few percent (form the true value). However, under the same tolerances, the direct sensitivity based method errors reached several hundred percent. Thus, decomposition based approach to testing proved to be more accurate than direct sensitivity method.

In the second study, catastrophic faults were injected into the analyzed circuit and the mean square error was computed for all the estimated circuit parameters. The decomposition based testing method delivered average errors of 0.14%, while the direct sensitivity based testing method resulted in the average parameter errors reaching 1630%. While the individual parameter errors may depend on circuit testability, in general the results obtained by the decomposition method are much more accurate than those obtained by the classical sensitivity method.

The developed decomposition approach to sensitivity based testing can be applied to test mixed-mode circuits. To facilitate this test, the design may include additional circuitry like multiplexers and shift registers. This multiplexer based testing, commonly used in digital circuits, may also benefit testing of complex analog circuits such as

artificial neural networks. By using multiplexers the internal nodes of the CUT are accessible, increasing system's controllability and observability. In case of such testing of mixed-mode circuits, digital subcircuits are tested using specialized algorithms for digital testing (e.g. scan path testing), while analog subcircuits use analog testing methods. Different subnetworks can be simulated and tested on different levels of circuit representation, like discrete element level, gate level, functional level, e.t.c. Some networks can be tested on the functional or the macro model level, while others on the subcircuit level where parameter identification can be performed. This decomposition based approach to mixed-mode testing, where an additional hardware is inserted in the signal path was facilitated by adopting IEEE 1149.4 standard. The standard describes the recommended hardware, software description language, and test procedures to aid the analog and mixed-mode testing.

4.4. Fault Verification in Multiple-Fault Diagnosis of Linear Networks

A considerable problem in analog testing is a limited number of independent measurements. This leads to ambiguities in locating and evaluating parametric faults. Fault verification technique is an effective approach to address this problem. The basic requirement of fault verification methods is that a tested circuit has a limited number of faulty parameters. Thus the number of measurements required by verification approaches is greater than the number of faults in a faulty circuit. The circuit topology and the nominal circuit parameters are assumed known. The test requires only the voltage measurements at the selected test nodes. This requirement reduces the measurement errors introduced by the test instrumentation and it is easily satisfied in practice. The basic idea behind the fault verification techniques is checking consistency of certain equations that are invariant to the changes in faulty parameters. Consistency is checked within the tolerance limits of the circuit components.

In addition, fault verification techniques are capable to deal with multiple-faults and accurately diagnose even large deviations of the parameter values. Dictionary techniques are other widely used approaches in fault diagnosis, but they require significant simulation efforts before the test [141], [7]. That is why a dictionary technique usually diagnoses only single catastrophic faults. Although neural networks can help to reduce costly sampling of the fault space [220], [34], the accuracy of fault location and parameter evaluation is limited in these approaches.

In [197], a fault verification technique was developed for piecewise linear analog circuits based on homotopy approach and on bilinear transformation, but it is only applicable for single-fault case. Another fault verification approach was proposed in [252] for both linear and non-linear circuits, but it is not reliable when the values of deviations are large. Large change sensitivity analysis was utilized in [278], [238] to handle large deviations of parametric faults. In addition, a symbolic analysis was utilized in [76], [82] to address the problems of computation round-off errors and large amount of computations required. Finally, to efficiently recognize the ambiguity groups hidden within the fault diagnosis equations, different ambiguity group locating techniques were proposed for fault verification methods [244], [77], [222], [239], [241], [253], and [153].

In what follows, a general and efficient fault verification technique is proposed. In [148], a fault verification method was proposed for multiple fault diagnosis of linear analog circuits in the frequency domain. The method needs additional equations to describe the catastrophic faults and uses either multiple-excitations or multiple-measurements but not both. This approach is generalized here to present a new fault verification technique where both multiple-excitations and multiple-measurements are utilized to obtain independent measurements. Both catastrophic faults and parametric faults can be diagnosed. The ambiguity group locating technique developed in [148] is significantly simplified here, since the first stage (Gaussian elimination) and the third stage (swapping operation) required in [148] are completely removed. Hence, computational efforts are greatly reduced.

4.4.1. Fault Diagnosis Equations

Assume that the linear circuit under test has $n+1$ nodes and p parameters in the impedance form $Z_v, (v=1,2,...,p)$. Each circuit parameter Z_v can be described by two-port like model: its controlled port is located between nodes i_v and j_v while its controlling port is located between nodes k_v and l_v. Thus, the circuit topology can be described by two nxp structural matrices P and Q which are defined as follows [267]:

$$P=[e_{i_1}-e_{j_1} \quad e_{i_2}-e_{j_2} \quad \dots \quad e_{i_p}-e_{j_p}], \textit{ and}$$
$$Q=[e_{k_1}-e_{l_1} \quad e_{k_2}-e_{l_2} \quad \dots \quad e_{k_p}-e_{l_p}], \tag{4.75}$$

where $e_v\,(v=1,2,...,p)$ represents an $nx1$ vector of zeros except for the v-th entry, which is equal to one. Suppose that there are e different excitations applied to both the fault-free and faulty circuits. KCL is first applied to the fault-free circuit to yield:

$$PI_{b,0}=J, \tag{4.76}$$

where $I_{b,0}$ is a pxe matrix of branch currents (i.e. current sources located at individual parameters positions) and J is an nxe matrix of independent branch current excitations.

Thus (4.76) represents KCL equations in the CUT resulting from independent current excitations. Likewise by using KVL in the same fault-free circuit, one gets:

$$Q^T V_{n,0}-Z_0 I_{b,0}=0, \tag{4.77}$$

where the superscript T denotes transposition of a matrix, $V_{n,0}$ is an nxe matrix of nodal voltages which correspond to independent current excitations at all accessible nodes, Z_0 is a pxp diagonal matrix of nominal parameter impedances:

$$Z_0=diag(Z_{v,0}). \tag{4.78}$$

Notice that each column in matrix $V_{n,0}$ corresponds to a full vector of nodal voltages obtained under the excitation vector from a corresponding column in matrix J. Replace the nominal parameter impedance $Z_{v,0}$ by a new variable $F_{v,0}$ as follows:

$$F_{v,0} = \frac{1}{Z_{v,0} + 1}, \quad Z_{v,0} \in [0, +\infty). \tag{4.79}$$

The benefit for replacing $Z_{v,0}$ by $F_{v,0}$ is to eliminate the extreme values in $Z_{v,0}$: (0 and ∞), which correspond to short-circuit and open-circuit faults. Subsequently, the equation (4.78) can be transformed to:

$$Z_0 = diag\left(\frac{1 - F_{v,0}}{F_{v,0}}\right) = F_0^{-1}(1 - F_0), \tag{4.80}$$

where

$$F_0 = diag(F_{v,0}). \tag{4.81}$$

Similarly, for a network with faulty parameters we have:

$$Z = F^{-1}(1 - F). \tag{4.82}$$

Equation (4.77) can be transformed to:

$$F_0 Q^T V_{n,0} + (F_0 - 1) I_{b,0} = 0. \tag{4.83}$$

Combining (4.76) and (4.83), we have equation which describes the system under nominal condition

$$\begin{bmatrix} 0 & P \\ F_0 Q^T & F_0 - 1 \end{bmatrix} \begin{bmatrix} V_{n,0} \\ I_{b,0} \end{bmatrix} = \begin{bmatrix} J \\ 0 \end{bmatrix}. \tag{4.84}$$

Now, consider the faulty condition. Assume that there are f of p faulty parameters in the faulty circuit with $f \le e$. The faulty parameters are changed from theirs nominal values $Z_{v,0}$ to $Z_v = Z_{v,0} + \Delta Z_v$, where ΔZ_v are the deviations from the nominal values. So that the corresponding diagonal matrix can be written as:

$$Z = Z_0 + \Delta Z. \tag{4.85}$$

Similarly, the nominal values of $F_{v,0}$ are changed to $F_v = F_{v,0} + \Delta F_v$ and the diagonal matrix F_0 becomes

$$F = F_0 + \Delta F. \tag{4.86}$$

Correspondingly, the system equation for the faulty circuit is as follows:

$$\left(\begin{bmatrix} 0 & P \\ F_0 Q^T & F_0 - 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \Delta F\, Q^T & \Delta F \end{bmatrix}\right)\left(\begin{bmatrix} V_{n,0} \\ I_{b,0} \end{bmatrix} + \begin{bmatrix} \Delta V_n \\ \Delta I_b \end{bmatrix}\right) = \begin{bmatrix} J \\ 0 \end{bmatrix}, \tag{4.87}$$

where ΔF is a *pxp* diagonal deviation matrix, ΔV_n is an *nxe* matrix of nodal voltage deviations, and ΔI_b is a *pxe* matrix of branch current deviations. Let us denote

$$\begin{bmatrix} V_n \\ I_b \end{bmatrix} = \begin{bmatrix} V_{n,0} \\ I_{b,0} \end{bmatrix} + \begin{bmatrix} \Delta V_n \\ \Delta I_b \end{bmatrix}. \tag{4.88}$$

Equation (4.87) can be simplified by subtracting (4.84) to obtain:

$$\begin{bmatrix} 0 & P \\ F_0 Q^T & F_0 - 1 \end{bmatrix}\begin{bmatrix} \Delta V_n \\ \Delta I_b \end{bmatrix} = \begin{bmatrix} 0 \\ -\Delta F\, Q^T\, V_n - \Delta F\, I_b \end{bmatrix}. \tag{4.89}$$

Suppose that *m* nodal voltages are measured with *e<m*, then (4.89) can be rearranged as follows:

$$\begin{bmatrix} 0 & P \\ F_0\, Q_2^T & F_0 - 1 \end{bmatrix}\begin{bmatrix} \Delta V_n^{N-M} \\ \Delta I_b \end{bmatrix} = \begin{bmatrix} 0 \\ -\Delta F\left(Q^T V_n + I_b\right) \end{bmatrix} + \begin{bmatrix} 0 \\ -F_0\, Q_1^T\, \Delta V_n^{M} \end{bmatrix}. \tag{4.90}$$

where M denotes the set of measured nodes and N denotes the set of all nodes, Q_1 is an *mxp* structural matrix and Q_2 is a *(n-m)xp* structural matrix.

Equation (4.90) which is called fault diagnosis equation relates the measured responses deviations ΔV_n^{M} with the faulty parameter deviations ΔF. The left-hand side (n+p)x(n+p-m) coefficient matrix of (4.90) can be constructed from the knowledge of the circuit topology and the nominal values of circuit parameters, thus it is known before test. The unknown deviation matrix of (4.90) (solution matrix) has a size of (n+p-m)xe. The right-hand side of (4.90) is a (n+p)xe matrix with fxe unknown entries due to faults. Thus, (n+p-f)xe linear equations with (n+p-m)xe variables can be obtained from (4.90). Since m>f, the number of equations is more than the number of unknown variables, satisfying a necessary condition for solvability of the fault diagnosis equation.

4.4.2. Fault Diagnosis Process

Suppose that the locations of all the faulty components are known, (4.90) can be decomposed by permuting the matrix rows according to positions of faulty parameters:

$$\begin{bmatrix} 0 & P_f & P_{p-f} \\ F_{0,1}Q_{21}^T & F_{0,1}-1_{fxf} & 0 \\ F_{0,2}Q_{22}^T & 0 & F_{0,2}-1_{(p-f)x(p-f)} \end{bmatrix} \begin{bmatrix} \Delta V_n^{N-M} \\ \Delta I_b \end{bmatrix} = \begin{bmatrix} 0 \\ -[\Delta F_1 \;\; 0]\left(Q^T V_n + I_b\right) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -F_{0,1}Q_{11}^T \Delta V_n^M \\ -F_{0,2}Q_{12}^T \Delta V_n^M \end{bmatrix} \quad (4.91)$$

where

$$F_0 = \begin{bmatrix} F_{0,1} & 0 \\ \hline 0 & F_{0,2} \end{bmatrix}, \quad (4.92)$$

and

$$\Delta F = \begin{bmatrix} \Delta F_1 & 0 \\ \hline 0 & \Delta F_2 \end{bmatrix} = \begin{bmatrix} \Delta F_1 & 0 \\ \hline 0 & 0_{(p-f)xp} \end{bmatrix}. \quad (4.93)$$

In a similar fashion, we define $Q_1 = [Q_{11} \;\; Q_{12}]$ and $Q_2 = [Q_{21} \;\; Q_{22}]$. Here, the matrices P, F_0, ΔF, Q_1 and Q_2 are decomposed into two parts: the first part corresponds to faulty parameters, while the second one corresponds to fault-free parameters. Hence, $\Delta F_2 = 0_{(p-f)xp}$.

By appending the right-hand side of (4.91) to the coefficient matrix of (4.91) and transposing the result, we construct a *(n+p+e-m)x(n+p)* matrix B_S

$$B_S = \begin{bmatrix} 0 & P_f & P_{p-f} & 0 \\ F_{0,1}Q_{21}^T & F_{0,1}-1_{fxf} & 0 & -[\Delta F_1 \;\; 0]\left(Q^T V_n + I_b\right) - F_{0,1}Q_{11}^T \Delta V_n^M \\ F_{0,2}Q_{22}^T & 0 & F_{0,2}-1_{(p-f)x(p-f)} & -F_{0,2}Q_{12}^T \Delta V_n^M \end{bmatrix}^T$$

$$= \begin{bmatrix} 0 & Q_{21}F_{0,1} & Q_{22}F_{0,2} \\ P_f^T & F_{0,1}-1_{fxf} & 0 \\ P_{p-f}^T & 0 & F_{0,2}-1_{(p-f)x(p-f)} \\ 0 & \left(-[\Delta F_1 \;\; 0]\left(Q^T V_n + I_b\right)\right)^T - \left(\Delta V_n^M\right)^T Q_{11}F_{0,1} & -\left(\Delta V_n^M\right)^T Q_{12}F_{0,2} \end{bmatrix}_{(n+p+e-m)x(n+p)} \quad (4.94)$$

Similarly, we can construct a *(n+p+e-m)x(n+p)* new matrix, called **verification matrix *B***, by appending the second term of the right-hand side of (4.90) to the coefficient matrix of (4.90) and transposing the result:

$$B=\begin{bmatrix} 0 & P & 0 \\ F_0 Q_2^T & F_0 - 1 & -F_0 Q_1^T \Delta V_n^M \end{bmatrix}^T = \begin{bmatrix} 0 & Q_2 F_0 \\ P^T & F_0 - 1 \\ 0 & -\left(\Delta V_n^M\right)^T Q_1 F_0 \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & Q_{21} F_{0,1} & Q_{22} F_{0,2} \\ P_f^T & F_{0,1} - 1_{fxf} & 0 \\ P_{p-f}^T & 0 & F_{0,2} - 1_{(p-f)x(p-f)} \\ 0 & -\left(\Delta V_n^M\right)^T Q_{11} F_{0,1} & -(\Delta V_n^M)^T Q_{12} F_{0,2} \end{bmatrix}_{(n+p+e-m)\times(n+p)} \tag{4.95}$$

There are two differences between B_S and B: the first term of the right-hand side of (4.90) is omitted in matrix B, and the faulty components in matrix B are not identified. Hence, matrix B can be obtained from circuit topology and parameter nominal values.

If there is at least one faulty component ($f \geq 1$) and all the current excitations are independent, the unknown solution matrix of (4.91) is not a zero matrix. Thus the right hand side of (4.91) can be expressed as a linear combination of columns of the left hand side matrix. Appending these columns to the coefficient matrix does not increase its rank, which is less or equal to $n+p-m$. So, the rank of $(n+p-m+e)x(n+p)$ matrix B_S is also less or equal to $n+p-m$. The rank of matrix B is less or equal to $n+p-m+f$, and thus can be larger by f than the rank of the coefficient matrix. The expected increase of the rank of B over the rank of B_S is due to f faulty components.

One can start from the known matrix B to locate the faulty components. If ΔV_n^M is not zero, it means that at least one faulty parameter is detected. Thus the position of faulty parameters in the faulty circuit can be located by checking the dependency relationship between columns of the verification matrix B.

The columns of B correspond to the circuit nodes and parameters. The matrices B and B_S differ only in these columns that contain faulty parameters. These altered columns are the only reason for the rank increase. These columns are independent from the remaining columns in matrix B, therefore, locating the faulty parameters is equivalent to finding these independent columns. One approach is an exhaustive search that requires on the order of $O\left(\binom{n+p}{f}\right)$ operations. More efficient approaches will reduce the testing cost. A search technique developed in [148] which finds the independent columns in time proportional to $O(p^3)$ will be modified here to improve the method efficiency.

The rank of matrix B determines the maximum number of faulty parameters that can be uniquely identified by solving (4.90). Matrix B can be written as

$$B = B_1 [I \ C] \tag{4.96}$$

where $(n+p+e-m)xr$ matrix B_1 has the full column rank equal to the rank r of the matrix B, and $rx(n+p-r)$ matrix C expands the dependent columns of B into a set of the basis columns B_1. Performing QR factorization of B, we can formulate the following equation:

$$BE = \hat{Q}R, \tag{4.97}$$

where E is $(n+p)x(n+p)$ permutation matrix with only a single nonzero element equal to one in each column, $\hat{Q}$ is $(n+p+e-m)x(n+p+e-m)$ orthogonal matrix, and R is $(n+p+e-m)x(n+p)$ upper triangular matrix. Upper triangular matrix R can be written as

$$R = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}, \tag{4.98}$$

where R_1 is rxr upper triangular matrix and R_2 is $rx(n+p-r)$ matrix.

Theorem 4.5 [241]. *A* ***linear combination matrix*** *C can be numerically obtained from the QR factorization of the matrix B using*

$$C = R_1^{-1} R_2 . \tag{4.99}$$

Thus C is $rx(n+p-r)$ matrix. Since the number of measurements m is larger, by at most the number of independent excitations, matrix C has a very limited number of columns due to $n+p-r \geq m-e$. The number of rows of matrix C is relatively large comparing with its number of columns due to $r \leq n+p+e-m$.

Fault diagnosis equation (4.90) contains unknown matrix of voltage and current deviations on the left-hand side and partly unknown vector on the right-hand side. The verification matrix B has the rank as high as $n+p+f-m$ (where $f \leq e$), however, the rank of $(n+p-m)x(n+p)$ matrix S

$$S = \begin{bmatrix} 0 & Q_2 F_0 \\ P^T & F_0 - 1 \end{bmatrix} \tag{4.100}$$

is less or equal to $n+p-m$. So, the increase in the rank of matrix B over the rank of matrix S may be as high as $(n+p+f-m)-(n+p-m)=f$. This rank increase is due to the presence of faults, which make part of the right-hand side of (4.82) independent on rows of matrix S.

Lemma 4.6. *All columns of matrix B which correspond to faulty parameters are forced to the basis and columns of matrix B which are not in the basis are independent from these columns.*

The independency relation in Lemma 4.6 is reflected in the linear combination matrix C by the presence of a row with all zero entries. Hence the following Lemma holds based on Lemma 4.5.

Lemma 4.7. *If all of the faulty parameters are included in the basis, then all the circuit parameters corresponding to zero rows in the matrix C are faulty.*

Since $f \leq e$ and faulty parameters are independent from each other, all of the faulty parameters are guaranteed to be included in the basis. Therefore, by applying

Lemma 4.7 to linear combination matrix C, the faulty elements can be identified directly (no search for a minimum size ambiguity group by column swapping is needed, unlike in the procedure presented in [148] which requires such search). In fact, similar to columns of matrix B, rows of the linear combination matrix C correspond to a combination of circuit parameters and nodes. Some nodes associated with faulty components will be represented by zero-entry-rows in the linear combination matrix C and may be also located by Lemma 4.7. The proposed verification technique is to directly locate faulty parameters, so that these identified faulty excitation nodes could be used either for additional verification of faulty component location or just can be discarded.

After the location of faulty parameters, (4.99) can be decomposed according to positions of faulty parameters as in (4.100). Decompose (4.100) into two smaller size sub-matrices as follows:

$$\begin{bmatrix} 0 & P \\ F_{0,2}\,Q_{22}^T & \begin{bmatrix} 0_{(p-f)xp} & F_{0,2} - 1_{(p-f)x(p-f)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta V_n^{N-M} \\ \Delta I_b \end{bmatrix} = \begin{bmatrix} 0 \\ -F_{0,2}\,Q_{12}^T\,\Delta V_n^M \end{bmatrix}, \qquad (4.101)$$

and

$$\begin{bmatrix} F_{0,1}\,Q_{21}^T & \begin{bmatrix} F_{0,1} - 1_{fxf} & 0_{fx(p-f)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta V_n^{N-M} \\ \Delta I_b \end{bmatrix} = \begin{bmatrix} -\begin{bmatrix} \Delta F_1 & 0 \end{bmatrix} \left(Q^T V_n + I_b \right) \end{bmatrix} + \begin{bmatrix} -F_{0,1}\,Q_{11}^T\,\Delta V_n^M \end{bmatrix} \qquad (4.102)$$

The solution to (4.101) can be uniquely determined by

$$\begin{bmatrix} \Delta V_n^{N-M} \\ \Delta I_b \end{bmatrix} = \left((S_1)^t S_1 \right)^{-1} (S_1)^t \begin{bmatrix} 0 \\ -F_{0,1}\,Q_1^T\,\Delta V_n^M \end{bmatrix}, \qquad (4.103)$$

where

$$S_1 = \begin{bmatrix} 0 & P \\ F_{0,1}\,Q_2^T & F_{0,1} - 1_{(p-f)xp} \end{bmatrix}. \qquad (4.104)$$

Subsequently, all the values of nodal voltages V_n and branch currents I_b in the faulty circuit can be obtained by using (4.88). Re-arranging (4.102), we have

$$\begin{bmatrix} \Delta F_1 & 0 \end{bmatrix} H = \begin{bmatrix} F_{0,1} - 1_{fxf} & 0_{fx(p-f)} \end{bmatrix} \Delta I_b - \mathrm{F}_{0,1} \begin{bmatrix} Q_{21}^T & Q_{11}{}^T \end{bmatrix} \Delta V_n = S_2, \qquad (4.105)$$

where

$$H = Q^T V_n + I_b\,. \qquad (4.106)$$

The size of the coefficient matrix H is pxe. To reduce the computational effort, select only one column H_1 from matrix H and only one column S_{21} from matrix S_2. The faulty parameter deviations can be exactly computed by solving (4.105):

$$\Delta F_1 = S_{21} \ rdivide \ H_1 \tag{4.107}$$

where *rdivide* is an element-by-element division of two vectors. To reduce effects of the round-off errors or element tolerances, equation (4.107) can be solved by using element by element division of S_2 by H and averaging the result to obtain ΔF_1.

The deviation values of the circuit impedances can be derived from (4.80), (4.82), (4.85) and (4.86)

$$\Delta Z = -\frac{\Delta F}{F_0(F_0 + \Delta F)}. \tag{4.108}$$

Comparing with the ambiguity group locating technique presented in [148], this verification technique does not search for the minimum size ambiguity group, but locates faulty parameters directly without Gaussian elimination step and without swapping operations, thus reducing the computational cost.

Example 4.5. An example circuit is shown in Fig. 4.14 and contains 8 nodes and 21 parameters. The nominal values of the cascaded FET amplifier are indicated in Fig. 4.14. The simplified model of FET (Fig. 4.15 has the following nominal values: $g_m = 5\times10^{-3}\,\Omega^{-1}$, $g_D = 10^5\,\Omega^{-1}$, $C_{GS} = 2\,pF$, $C_{GD} = 2\,pF$, and C_{DS} is neglected. The current excitation source of two different values $J = 0.1\,A$ and $J = 0.1\sin\left(2\pi\,x\,10^5 \times t\right)A$ is applied alternatively between the pairs of nodes {0, 1} and {0, 4} respectively, and two sets of voltage measurements are made with these two excitations. Note that Fig. 4.14 shows only the case of the current excitation source applied between nodes {0,1}.

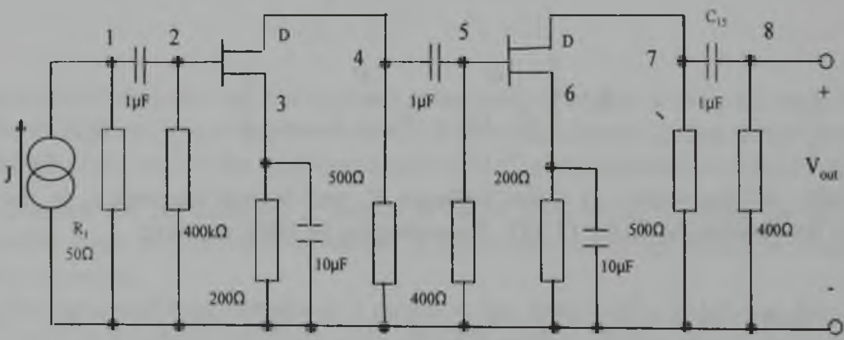


Fig. 4.14. Cascaded FET amplifier

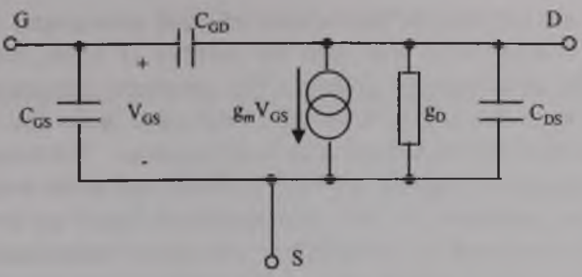


Fig. 4.15. FET model

Assume that there are two faulty parameters in this circuit: R_1 that is changed from 50 Ω to 500 Ω and C_{15} changed from 1 µF to 5 µF. The corresponding impedance deviations are $\Delta Z_1 = 450\,\Omega$ and $s\Delta C_{15} = 2.51\,j\,\Omega$. The nodal voltages at nodes {1, 4, 6, 8} are measured. Thus $n=8$, $p=21$, $e=2$, $f=2$, $m=4$ and $f \le e < m$. The measured nodal voltage deviations under two distinct excitations are:

$$\Delta V_n^M = \begin{bmatrix} 4.50*10^{+1} - 6.20*10^{-2}\,j & 1.57*10e^{-5} + 4.48*10^{-9}\,j \\ 4.50*10^{+1} - 6.19*10^{-2}\,j & 1.57*10^{-5} + 4.54*10^{-9}\,j \\ 1.80*10^{-5} + 3.00*10^{-9}\,j & 6.38*10^{-12} + 1.18*10^{-14}\,j \\ 7.17*10^{-12} + 3.83*10^{-13}\,j & 2.50*10^{-10} + 1.38*10^{-11}\,j \end{bmatrix},$$

which indicates the presence of fault(s) in the circuit.

By applying QR factorization to the fault diagnosis equation, *27x2* matrix C with rank of $r=2$ is obtained. Subsequently, by analyzing the permutation matrix E, the co-basis includes only two circuit nodes {4, 8} and the remaining 6 nodes and 21 parameters are included in the basis.

By analyzing this *27x2* matrix C, the only zero rows found correspond to parameters $\{R_1, C_{15}\}$. According to Lemma 4.7, these parameters are judged as faults since all the circuit parameters are included in the basis. This conclusion correctly identifies faults inserted in the given FET amplifier circuit.

Similarly, the deviations of faulty parameters are exactly calculated as:

$$\begin{bmatrix} \Delta Z_1 \\ s\Delta C_{15} \end{bmatrix} = \begin{bmatrix} 4.5000\times 10^{2} + 8.7393\times 10^{-14}\,j \\ 2.5133\,j \end{bmatrix},$$

where the small imaginary term of Z_1 can be omitted.

In the proposed fault verification method, fault diagnosis equation is established by analyzing the circuit topology and utilizing the KCL and KVL to relate the measured response deviations to faulty parameter deviations. The coefficient matrix is only determined by the circuit topology and the nominal values of the circuit parameters. Catastrophic faults are represented by only one variable with finite numerical values, so that they can be treated like parametric faults. In this method, multiple-excitations and

multiple-measurements are required for the location of faulty parameters. In addition, the number of excitations should be no less than the number of faults, but less than the number of the selected measurement nodes. The proposed approach is extremely effective for large parameter deviations including catastrophic faults and limited number of accessible test nodes used for excitations and measurements. The computational cost for the ambiguity group location required in [148], is eliminated in this method.

Another effective approach to fault verification is based on the large change sensitivity analysis. This approach is both effective and easy to understand, thus it can be used to illustrate both the concept of fault verification and to present a novel way of using large change sensitivity analysis. It is discussed in the following section.

4.4.3. Large Change Sensitivity in Fault Diagnosis

Large change sensitivity analysis is one of the approaches applied to develop test verification methods for analog fault diagnosis [12], [251], [273], [278]. A significant advantage of the large change sensitivity analysis is that both the parametric and catastrophic faults (open or short circuits) can be exactly solved. In the SBT methods [11], large change sensitivity methods are typically utilized to construct the fault dictionary. This approach is characterized by large computational requirements. The method proposed in [238] combined the large change sensitivity approach with the ambiguity group locating technique, expanding application of the large change sensitivity method into the Simulation-After-Test category and greatly reducing the computation cost. In this method, fault location is based on the QR factorization approach used to locate the ambiguity groups. The method yields a minimum size test equation. It is briefly characterized in this section.

Assume that the topology of the CUT as well as the nominal values of the circuit parameters are known, and the CUT has $n+1$ nodes and p parameters. Let us start with a typical modified nodal equation [267] to describe the circuit with nominal parameters:

$$T_0 X_0 = W_0 \tag{4.109}$$

where T_0 is an nxn coefficient matrix, X_0 is an $nx1$ vector of node voltage and/or parameter currents, and W_0 is an $nx1$ excitation vector.

The nominal solution vector X_0 is obtained by inverting non-singular coefficient matrix T_0:

$$X_0 = T_0^{-1} W_0 \tag{4.110}$$

Suppose that f of p parameters are faulty and deviate from their nominal values $h_{10}, h_{20}, \ldots, h_{f0}$ to the new values $h_{10} + \delta_1, h_{20} + \delta_2, \ldots, h_{f0} + \delta_f$, where $\delta_1, \delta_2, \ldots, \delta_f$ are the amounts of parameter changes. The new values of circuit parameters are:

$$h_\nu = h_{\nu 0} + \delta_\nu \qquad \nu = 1,2,\ldots,f \tag{4.111}$$

The corresponding changes in the coefficient matrix are in the form $p_\nu \delta_\nu q_\nu{}^t$ with

$$\begin{aligned} p_\nu &= e_{\nu_i} - e_{\nu_j} \\ q_\nu &= e_{\nu_k} - e_{\nu_l} \qquad \nu = 1,2,\ldots,f \end{aligned} \tag{4.112}$$

where superscript t represents the transpose of vector/matrix and e_ν represents an *nx1* vector of zeros except for the νth entry, which is equal to one.

The equation that describes the faulty circuit under the same excitations is

$$TX = \left(T_0 + \sum_{\nu=1}^{f} p_\nu \, \delta_\nu \, q_\nu^t \right) X = W_0 \tag{4.113}$$

or

$$\left(T_0 + P \; diag \;\; (\delta) \;\; Q^t \right) X = W_0 . \tag{4.114}$$

Here P and Q are *nxf* matrices which contain 0 and ± 1 entries:

$$\begin{aligned} P &= \lfloor p_1 \;\; p_2 \; \cdots \; p_f \rfloor \\ Q &= \lfloor q_1 \;\; q_2 \; \cdots \; q_f \rfloor \end{aligned} \tag{4.115}$$

and $diag(\delta)$ is an *fxf* diagonal matrix while δ is an *fx1* vector:

$$\delta = [\delta_1 \;\; \delta_2 \; \ldots \; \delta_n]^t \tag{4.116}$$

Equation (4.114) can be written as

$$\left(T_0 + P \, diag(\delta) \, Q^t \right) \left(X_0 + \Delta X \right) = W_0 \tag{4.117}$$

where

$$X = X_0 + \Delta X \tag{4.118}$$

After substituting (4.109) into (4.117), the following equation is established:

$$\Delta X = - T_0^{-1} P \, diag(\delta) \, Q^t X \tag{4.119}$$

Assume that the faulty parameter $F_\nu \, (\nu = 1,2,\ldots,f)$ is located on intersection of rows i_ν and j_ν and columns k_ν and l_ν of the coefficient matrix T then, considering (4.112), matrices P and Q in (4.115) have the following form:

$$P=[e_{i_1}-e_{j_1}\ \ e_{i_2}-e_{j_2}\ \ \dots\ \ e_{i_f}-e_{j_f}]$$
$$Q=[e_{k_1}-e_{l_1}\ \ e_{k_2}-e_{l_2}\ \ \dots\ \ e_{k_f}-e_{l_f}] \tag{4.120}$$

Let us denote an *nxn* matrix S_0 as follows

$$S_0=[s_1\ s_2\ \dots\ s_n]=-T_0^{-1} \tag{4.121}$$

and re-write vector X in the following form:

$$X=[x_1\ x_2\ \dots\ x_n] \tag{4.122}$$

where $s_v\ (v=1,2,\dots,n)$ is an *nx1* vector while $x_v\ (v=1,2,\dots,n)$ is a scalar. Thus the products of S_0 and P, Q' and X can be written as

$$\begin{aligned} S_{NF} &= S_0P=S_0[e_{i_1}-e_{j_1}\ \ e_{i_2}-e_{j_2}\ \ \dots e_{i_f}-e_{j_f}] \\ &=[s_{i_1}-s_{j_1}\ \ s_{i_2}-s_{j_2}\ \ \dots\ \ s_{i_f}-s_{j_f}] \\ Q'X &=[e_{k_1}-e_{l_1}\ \ e_{k_2}-e_{l_2}\ \ \dots e_{k_f}-e_{l_f}]'X \\ &=[x_{k_1}-x_{l_1}\ \ x_{k_2}-x_{l_2}\ \ \dots\ x_{k_f}-x_{l_f}]' \end{aligned} \tag{4.123}$$

where N indicates the set of all nodes and the **faulty set** F represents the set of all the faulty parameters. Denote an unknown *fx1* vector

$$\lambda_F=diag(\delta)\ Q'X \tag{4.124}$$

and consider (4.116) and (4.123)

$$\begin{aligned} \lambda_F &=diag(\delta)\ Q'X \\ &=diag(\delta)[x_{k_1}-x_{l_1}\ \ x_{k_2}-x_{l_2}\ \dots\ x_{k_f}-x_{l_f}]' \\ &=[\delta_1(x_{k_1}-x_{l_1})\ \ \delta_2(x_{k_2}-x_{l_2})\ \dots\ \delta_f(x_{k_f}-x_{l_f})]' \end{aligned} \tag{4.125}$$

Thus (4.119) can be re-written as

$$\Delta X=S_{NF}\lambda_F \tag{4.126}$$

Notice that introduction of the unknown vector λ_F reduced apparent complexity of the test equation by combining several unknown values into a single one. Assume that the first m elements of ΔX can be measured and that $f+1<m<p$. Under this assumption we obtain

$$\begin{bmatrix} \Delta X^{M} \\ \Delta X^{N-M} \end{bmatrix} = \begin{bmatrix} S_{MF} \\ S_{N-M,F} \end{bmatrix} \lambda_F \tag{4.127}$$

where M represents the set of measurements. Hence, the following equation is obtained:

$$\Delta X^{M} = S_{MF}\,\lambda_F \tag{4.128}$$

Here S_{MF} is an *mxf* matrix whose columns correspond to the faulty parameters in the analyzed circuit. Similarly S_{MP} is an *mxp* matrix whose columns corresponding to all of the parameters in the circuit, where P indicates the set of all parameters. Equation (4.128) is called the **test equation** and matrix S_{MP} is called the **test matrix**.

Fault Diagnosis

If the vector ΔX^M is a zero vector, it means that the CUT is fault-free, i.e., no faults can be detected by the existing measurements. Otherwise, the CUT is judged to be faulty. Typically we do not know the exact locations of faulty parameters, i.e., the elements of faulty set F are unknown. Thus, the matrices P, Q, S_{MF} are also unknown. In order to diagnose the faulty circuit, we need to find out the sets of columns in the test matrix S_{MP} that satisfy (4.128) with the minimum size faulty set F, or otherwise to find out the ambiguity groups F with the minimum size using columns of S_{MP} that satisfy (4.128). Ambiguity group locating techniques proposed in [241], [239] can be utilized to implement this objective, yielding the following procedure:

Fault Diagnosis Based on Large Change Sensitivity

1. Append the vector ΔX^M into matrix S_{MP} to construct a new $mx(p+1)$ matrix with ΔX^M being its first column.
2. Eliminate the first column ΔX^M from the new matrix by Gauss Elimination step and call the resulting $(m-1)xp$ matrix **test verification matrix**.
3. Apply the QR factorization step to the test verification matrix to obtain a linear combination matrix C. The rows of matrix C correspond to the components in the basis while the columns of matrix C correspond to the components in the co-basis. Any zero entries in matrix C correspond to an ambiguity group.
4. If column k of matrix C has at least one zero entry, then the corresponding ambiguity group consists of one co-basis component corresponding to column k and of all the basis components corresponding to nonzero entries in column k.
5. To find out the ambiguity group F with the minimum size, swap an element in the co-basis with an element in the basis of faulty set F in order to increase the number of zero entries in the new resulting matrix C.

Two lemmas concerning the swapping operation and establishing relationship between faulty set F and test equation are excerpted as follows:

Lemma 4.8 [239]. A necessary condition for swapping to increase the number of zero coefficients in C is that the columns of basis and co-basis to be swapped have a 2x2 singular submatrix of nonzero coefficients.
Lemma 4.9 [239]. A necessary condition for an ambiguity group F of the linear combination matrix C to contain the set of all faults in the tested circuit is that the rank of the corresponding columns in the original test matrix S_{MP} is equal to the cardinality of F

$$rank(S_{MF}) = \overline{\overline{F}} . \tag{4.129}$$

The methods discussed in [239] utilize a similar ambiguity groups locating technique. In addition, the method proposed in [239] uses nodal analysis and faulty current nodes are found in two steps. In the first step ambiguity groups are located (see the following section). Subsequently, all the faulty parameters are located by using the incident signal matrix. Described in this section verification method is based on the large change sensitivity analysis, and faulty parameters are located directly.

After locating ambiguity groups in the test equation, we know which parameters in the circuit are faulty. Assume that S_{MF} is a full column rank matrix, vector λ_F is then obtained by solving (4.128):

$$\lambda_F = \left(S_{MF}{}^t S_{MF}\right)^{-1} S_{MF}{}^t \Delta X^M \tag{4.130}$$

After that, full vector ΔX can be computed by (4.126) since matrix S_{NF} and vector λ_F are known. The solution vector X is consequently determined by (4.118). Finally the parameter deviations δ can be obtained by solving (4.124):

$$\delta = \begin{bmatrix} \dfrac{\lambda_1}{x_{k_1} - x_{l_1}} & \dfrac{\lambda_2}{x_{k_2} - x_{l_2}} & \cdots & \dfrac{\lambda_f}{x_{k_f} - x_{k_f}} \end{bmatrix}^t \tag{4.131}$$

Although the large change sensitivity approach gives exact solutions in arbitrary large deviations of design parameters from their nominal values, it requires finding proper combination of columns of the test matrix S_{MP} that satisfy (4.128). This problem is indirectly related to testing low testability circuits as discussed in the next section.

4.5. Low Testability Analog Circuits

Analog circuit testability quantifies the degree of problem solvability and is related to the network element value solvability introduced by Berkowitz [19]. A well-defined quantitative testability measure was introduced by Sen et al. [210], [42]. Given the circuit topology, selected test points and unknown components, testability can establish a unique solvability of the testing problem. In the worst case it is necessary to

consider further test points or to decrease the number of potentially faulty components that can be uniquely diagnosed.

There is an important phenomenon which is commonly encountered and is difficult to solve in analog testing and fault diagnosis: this phenomenon is the measurement ambiguity. Distinct faults may result in the measurements whose values are close to each other. Therefore, it is difficult to clearly recognize a specific fault. Such faults are said to be in the same ambiguity set associated with a specific measurement. The concept of an **ambiguity set** was first introduced by Hochwald and Bastian [113]. It was defined as a list of inseparable faults that fall into a distinct band of the measurement levels. Such bands can be determined by Monte Carlo simulation considering component tolerances, tester errors, and circuit partition methods.

In the case of low-testability circuit, the canonical ambiguity group becomes extremely useful. Roughly speaking, an ambiguity group is a set of components that, if considered faulty, do not give a unique solution of the fault location. A canonical ambiguity group is an ambiguity group that contains no other ambiguity groups and it is related to the solvability of the fault diagnosis problem with a bounded number of faults (fault hypothesis). Moreover, if circuit testability and ambiguity groups are not taken into account properly, the quality of the obtained test results is severely limited [78].

Testability measure algorithms have been developed first by using a numerical approach [36]. Due to inevitable roundoff errors, they were limited to networks of a moderate size. However, this problem was solved by the symbolic approach [32], [142], [33] through manipulation of algebraic expressions. Furthermore, an efficient algorithm to find ambiguity groups has been proposed by Stenbakken, Souders, and Stewart [244]. A new method to determine all canonical ambiguity groups of an analog circuit has been co-developed by the author [77] by finding all possible ambiguity groups and all sets of circuit parameter values consistent with the test equations. However, the proposed algorithm was combinatorial in nature and was useful only for small analog circuits.

In [241], a numerically efficient approach was developed to identify complex ambiguity groups for analog fault diagnosis in low-testability circuits. Some major results from this work are presented here. In particular the discussion will focus on how circuits with low testability can be analyzed and how to handle ambiguity groups. The method presented in [241] finds which groups of components can be uniquely determined thus effectively solving low-testability problem.

Mathematically, an ambiguity group can be defined as a set of circuit parameters which correspond to linearly dependent columns of $m \times p$ testability matrix B, where the number of the measurements m is greater than the number of tested parameters p. In addition, a canonical ambiguity group is defined as a minimal set of parameters which correspond to linearly dependent columns of the testability matrix B. This means that if a single parameter is removed from the canonical ambiguity group, then the remaining set corresponds to independent columns of B and can be uniquely testable. All canonical ambiguity groups have the rank deficiency equal to one, which means that the rank of the corresponding set of columns is equal to the number of parameters in the canonical ambiguity group minus one.

The following presentss some of the theoretical results obtained for ambiguity group analysis.

Lemma 4.10 [241]. If A is a subset of columns of B that corresponds to the canonical ambiguity group, then the following equation is satisfied:

$$AC = 0 \tag{4.132}$$

where C is a vector with all nonzero coefficients.

The order of a canonical ambiguity group was defined in [77] as equal to the number of components included in the ambiguity group. A combination of canonical ambiguity groups with at least one common element is named the ambiguity cluster. Finally, all circuit components which correspond to columns of testability matrix that are not included in any ambiguity group are called surely testable components.

Finding a proper partition of the testability matrix is not a trivial task, as it may require combinatorial searches. In order to efficiently find such a partition for any ambiguity group, represent B in the factorized form $B = B_1[I\ C_1]$ with the linear combination matrix C_1 in a minimum form. The linear combination matrix C_1 is in a minimum form if it has the maximum number of coefficients equal to zero. The corresponding partition is called a canonical form of the testability matrix B.

As a result of the QR factorization of $m \times p$ testability matrix B one can formulate the following equation:

$$BE = QR \tag{4.133}$$

where E is a column selection matrix, Q is an orthogonal matrix, and R is an upper triangular matrix.

Each column of matrix E has only a single nonzero element equal to one. Matrix product BE represents a permutation of the original columns of the testability matrix. Matrix R has its rank equal to the rank of the testability matrix B. Therefore, let us assume that R was reduced to $p \times p$ matrix by removing all its zero rows. Furthermore, in the presence of ambiguity groups in the testability matrix, its rank and the rank of R are less than p. Therefore

$$R = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix} \tag{4.134}$$

where R_1 is $r \times r$ upper triangular and has its rank equal to the rank of the testability matrix B. The following theorem provides a basis for a numerically efficient approach to finding the ambiguity groups, the ambiguity clusters, and surely testable components.

Theorem 4.6 [241]. A linear combination matrix C_1 can be numerically obtained from the QR factorization of the testability matrix using

$$C_1 = R_1^{-1} R_2 \tag{4.135}$$

Based on the above discussion, identification of ambiguity groups becomes easy provided that the linear combination matrix C_1 is in its minimum form. The following lemma provides a sufficient condition for the matrix C_1 to be in the minimum form.

Lemma 4.11 [241]. If any two columns of the linear combination matrix C_1 have simultaneously nonzero elements in at most one common row, then C_1 is in its minimum form.

Different partitions define different linear combination matrices C_1. Let us define the basis of a partition as the set of components that correspond to columns of matrix B_1 and the cobasis as a set of components that correspond to columns of matrix B_2. A minimum form C_1 is not unique, as it is enough to switch a component of the basis (that corresponds to a row of C_1 with a single nonzero component) with the corresponding component of the cobasis (that corresponds to a column which includes this nonzero component) to obtain another minimum form of C_1.

As was discussed in [241], the system testability measure defined as the rank of the testability matrix is independent on parameter values, which means that the rank of the testability matrix is equal to a given testability measure almost everywhere in the parameter space. This result can be extended to ranks of all submatrices of the testability matrix that are used to determine the existence of ambiguity groups. To this end, properties of the linear combination matrix C_1 are determined considering its equivalent binary matrix D that has the same size as C_1. An element of the matrix D is equal to one if the corresponding element of C_1 is nonzero, and all other elements are set to zero. As it is in matrix C_1, rows of D correspond to the elements of the basis and columns correspond to the elements of the cobasis on a given partition. This equivalent representation simplifies the analysis of as the set theory can be used to study its structural properties.

Using the equivalent binary matrix, Lemma 4.11 can be written in the equivalent form as follows.

Lemma 4.12 [241]. If the intersection any two columns of the equivalent binary matrix D have at most one nonzero element, then C_1 is in its minimum form.

A useful result closely related to Lemma 4.12 describes the existence of surely testable components, canonical ambiguity groups, and ambiguity clusters.

Lemma 4.13 [241]. A circuit component is surely testable if and only if the corresponding row of matrix C_1 is zero.

In order to define the canonical ambiguity groups and the ambiguity clusters, let us identify a set of elements of the cobasis a_2 that corresponds to a union of columns of D with nonempty intersection. Let the set of the elements of the basis a_1 correspond to nonzero rows in the set of columns described by a_2.

Lemma 4.14 [241]. A set of components described by the union $a = a_1 \cup a_2$ constitutes an ambiguity group of the testability matrix.

Ambiguity groups identified by Lemma 4.14 are either ambiguity clusters or canonical ambiguity groups. Canonical ambiguity groups can be identified by using the following lemma.

Lemma 4.15 [241]. The ambiguity group represented by the set $a = a_1 \cup a_2$ is canonical if and only if cardinality of a_2 is equal to one.

Obviously, if the ambiguity group identified in Lemma 4.15 is not canonical then it is an ambiguity cluster. In the above lemmas it is not required for the linear combination matrix C_1 to be in the minimum form. All the columns of D are divided into row disjoint sets and this division identifies all the ambiguity groups, ambiguity clusters, and all surely testable components. The numerical cost of this partition is very modest compared to the combinatorial search. Detailed analysis of the ambiguity clusters and related examples are presented in [241].

The following procedure identifies all surely testable components, ambiguity clusters, and canonical ambiguity groups, as well as finds a canonical form of the testability matrix.

Procedure - Canonical Form of the Testability Matrix:

1) Formulate test equations (4.132) and identify the testability matrix.
2) Run the QR factorization on to obtain (4.133).
3) Use the column selection matrix E to find initial elements of the basis and the cobasis.
4) Represent R in the form (4.134).
5) Find the linear combination matrix using (4.135).
6) Find the equivalent binary matrix D for C_1.
7) Use Lemmas 4.13-4.15 to identify all surely testable components, canonical ambiguity groups, and ambiguity clusters.
8) Find a minimum form partition of each ambiguity cluster [241].
9) Combine all the basis components from all ambiguity clusters with all surely testable components to form the final basis and the cobasis.
10) Use the final basis to obtain the canonical form of the testability matrix.

The presented approach led to a fully automated method that, starting from the topology of the circuit under test, determines the ambiguity groups and the surely testable circuit parameters used to diagnose low-testability systems.

4.5.1. Fault Verification by Locating Ambiguity Groups

Using ambiguity groups approach a method was developed for the location of fault-free nodes and their verification in a linear network with N nodes, M measurement nodes and F faulty nodes with $M>F$. The network nodal equations can be formulated as follows

$$T_0 \begin{bmatrix} V_M \\ V_{N-M} \end{bmatrix} = [W_0] + [W_F] \tag{4.136}$$

where T_0 is the nominal multiterminal matrix of the decomposed network (size equal to the number of decomposition nodes), V_M and V_{N-M} are measured and unknown decomposition node voltages respectively, W_0 is known excitation vector, and W_F is unknown vector of faulty sources at faulty nodes.

Since, in general, location of fault-free nodes is unknown we need to determine the unknown voltages, identify fault free nodes and verify fault free equations. To this end let us first modify (4.136) as follows

$$T_1 V_M + T_2 V_X = [W_0] + [W_F] \quad (4.137)$$

$$T = [T_1 \quad T_2] \quad (4.138)$$

and move the first term from the left hand side to the right hand side and combine it with the right hand side vector to get

$$T_2 V_X = [\hat{W}_o] + [W_F] \quad (4.139)$$

where

$$[\hat{W}_o] = [W_0] - T_1 V_M \quad (4.140)$$

is a known vector. Let us formulate the ambiguity group matrix

$$B = [T_2 \quad \hat{W}_0] \quad (4.141)$$

This matrix has N rows and $N-M+1$ columns.

Since the entries in vector W_F corresponding to faulty nodes are nonzero while the entries corresponding to fault-free nodes are zeroes, $N-F$ equations can be obtained from (4.139) with $N-M$ unknowns if we know the exact location of faulty nodes. Hence, the unique solution to all V_X can be determined. To avoid a combinatorial search for faulty nodes, ambiguity groups locating technique can be utilized to efficiently locate all fault-free nodes [239]. The maximum number of fault-free nodes can be obtained using the following result.

Lemma 4.16 [237]. If $M>F$ and matrix B has the full column rank ambiguity group, then the row indices of the submatrix which forms this ambiguity group are fault-free nodes and all the subnetworks incident to these nodes are fault free.

The above method to locate the fault-free nodes is based on linear nodal analysis, thus is only applicable to linear networks. For nonlinear networks, Lemma 4.3 can be used to implement the location of fault-free nodes with the incident current I_c^i computed by (4.13). Note that only the measurement nodes should be used to partition the nonlinear networks.

After location of fault-free nodes and faulty subnetworks, the computation efforts required by faulty component location are limited. Since there is no strict requirement for the memory and testing time in today's analog test, any faulty component location

techniques inside the faulty subnetwork can be used. Such techniques are provided in part V of [201] or in other references for linear analog circuits [239], [253] and for nonlinear analog circuits [279].

For a linear network, equation (30) in [201] can be utilized to compute the external current. For the network with faulty nonlinear components, fault models of nonlinear components can be utilized to locate the faulty nonlinear components. Finally, for the network with faulty linear components and fault-free nonlinear components, utilize nonlinear network solver such as Pspice to locate the faults.

Discussed in this section testing conditions are independent of the network and an excitation type, thus the method is applicable to both linear and nonlinear networks, and to both time domain and frequency domain testing. The proposed method is particularly effective for the large-scale analog circuits. The method presented in this section can be extended to mixed mode systems as discussed in the next section.

4.5.2. Fault Diagnosis in Mixed-Signal Low Testability System

Mixed-signal ICs become more and more important with the quick development in many areas such as mobile communications, process control, automotive ASIC and smart sensors [107], [52], [22]. Due to strict the market requirements, these products must have high quality and low cost. As a result, test costs, time, and quality of mixed mode, mixed signal systems become more and more important.

Testability of digital IC has been studied extensively [86], [150]. Standard fault models such as "stuck-at" model provide a good fault coverage and short testing time. In addition, scan path access and boundary scan design [264] facilitate testing, and commercial design for testability (DFT) tools are available. However, since mixed-signal circuits lack standard analog fault models, standard test methodology and computer-aided test tools, the DFT of such circuits has lagged far behind [35], [79]. Furthermore, high quality analog tests are the most expensive in terms of both test development costs and test implementation. In the commercial designs, up to 80% of the test costs are due to the analog subcircuits that typically occupy only around 10% of the chip area.

Considering modern industry requirements, system testability in presence of ambiguity groups in mixed-signal mixed-mode systems deserves more attention. In low testability systems, no simple solution of test equations can be found using traditional methods, because the equations are singular. In what follows an efficient approach to identify ambiguity groups in such systems is presented. Although a unique solution is not always possible in such systems, the presented method provides the best possible alternative by providing a unique solution for all testable components. In addition, components within an ambiguity group have unique solution under the assumption that the number of faults is smaller than the rank of the corresponding ambiguity group.

Let us consider the test equations

$$BP = M \tag{4.142}$$

where B in $m \times p$ testability matrix, P and M are the changes of parameters and measurements from the nominal values, respectively. Since the number of faulty parameters is small, most of elements of vector P are zero. For a numerical stability and

a reduction of the roundoff errors the testability matrix B must have larger number of rows than columns. If the testability matrix has the full column rank, the tested circuit is fully testable and all the tested parameters P can be uniquely identified. However, for low testability system, this is not the case.

Using a pseudo inverse of B we can determine a solution of (4.142) as follows

$$\overline{P} = pinv\,(B)M \tag{4.143}$$

Depending on the structure of C_1 we either have or not have determined correct values of the faulty parameters. From (4.142) we can identify a partition of the solution components into P_1 and P_2 as follows:

$$BP = B_1[IC_1]\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = M \tag{4.144}$$

where P_1 corresponds to independent columns of B in a selected minimum form of C_1.

Since all the ambiguity groups involve different columns of B_1 matrix, their solutions can be obtained independently of each other. Let us consider a partition of (4.142) according to the ambiguity groups:

$$BP = \sum_{i=1}^{g} B_i P_i = \sum_{i=1}^{g} M_i = M \tag{4.145}$$

where B_i contains columns of B, which correspond to i-th ambiguity group (including independent columns of B). We know that in a low testability system the solution vector P is not unique, however, vectors M_i in equation (4.145) do not depend on a specific solution vector. In particular, M_i for each ambiguity group (including independent components) can be uniquely calculated using $\overline{P}$. Solving (4.142) with minimum number of faulty parameters will be obtained by solving each ambiguity group separately. We call such solution a **minimum form solution.**

An efficient procedure to find the minimum form solution of the test equations (4.142) was described in [182] and applied to a mixed-signal testing example. The approach presented in [182] finds the minimum form of the ambiguity groups. For each ambiguity group, an equivalent fault vector is built for which the system test equations have unique solution. The method defines the solution invariant test matrix required to test the low testability circuits.

Before testing procedure can be developed and efficiently applied to analog circuit, the optimum selection of test points is required to achieve the desired degree of fault diagnosis and to maintain a reasonable computational cost. Entropy based test point selection is discussed next and it is based on the method presented in [246].

4.6. Entropy Based Optimum Test Points Selection

Optimum selection of test points is very important stage in analog fault verification and fault dictionary techniques. The global minimum solution is only guaranteed by an exhaustive search which is NP-hard, therefore it is impractical for medium or large systems. A local minimum set is a useful compromise for efficiently achieving the desired degree of fault diagnosis. The method presented in [246] finds the minimum test set by using the entropy index for test points' selection. Since no theoretical proof of optimality can be given for this method, statistical experiments were utilized for its evaluation. Experiments demonstrated that the method is superior to other methods in computational efficiency and quality of the final solution; therefore, it is a good candidate for testing large scale systems.

Fault dictionary techniques are usually used to diagnose the open or short faults [144] with possible inclusion of the parametric fault diagnosis [141], [278]. Fault dictionary techniques have the advantage of minimum on-line computation, but a significant off-line computation needed to construct the database limits their applications. For that reason, an optimum selection of test points is important to reduce the computation cost by reducing dimensionality of the fault dictionary. Simultaneously, an optimum selection of test points reduces the test cost by eliminating redundant measurements. This section emphasizes selection procedures of an optimum set of test points. One straightforward solution is to have an exhaustive search for such a set with a minimum size to fully isolate the faults. The exhaustive search is proven to be NP-hard. Therefore, such search is not practical considering the computation cost, while any polynomial-bounded method cannot guarantee selection of a global minimum set. An acceptable tradeoff between the desired degree of fault diagnosis and the computation cost is to select a local minimum set.

A heuristic method for test points selection based on the concept of the confidence levels was proposed by Varghese *et al.* [262] with large computational complexity of $O(kfn(f+n))$ [192] where f is the number of faults, n is the total number of system nodes, and k is the number of times to compute the confidence level. Stenbakken and Souders [250] proposed a method using the QR factorization of a system sensitivity matrix. The complexity is primarily determined by the complexity of QR factorization $O(n^3)$. Abderrahman *et al.* 1 used sequential quadratic programming and constraint logic programming to generate test sets. Lin and Elcherif [144] proposed two heuristic methods based on two criteria proposed by Hochward and Bastian [113]. The complexities are $O(f^2p^2)$ [192] and $O(f^2p)$ [191], respectively, where p is the number of examined test nodes. Spaandonk and Kevenaar [219] looked for a set of test points by combining the decomposition method of system sensitivity matrix and an iterative algorithm. A set of test points whose size is equal to the rank of system sensitivity matrix is selected randomly. Then, in the iterative algorithm, they randomly exchange a test point in the set with a randomly selected integer in order to compute, the determinant of covariance matrix. The new set is accepted if it has a lower determinant than the previous set. Manetti *et al.* [152] wrote a computer program in PROLOG to automatically select a set of test points for linear analog systems. The program was based

on a knowledge base constituted by some simple rules derived from experience and heuristic reasoning. It was an example application of expert systems. All these methods are to find a local minimum set with complexity which is polynomially bounded. Prasad and Pinjala [192] proposed a polynomial-bounded method with complexity of $O(fp)$, but the selection of a hashing function needed is difficult for general test cases. Prasad and Babu [191] proposed four algorithms based on three different strategies for inclusive approaches and three strategies for exclusive approaches. The complexity is $O(fp\log(f))$ or $O(f(p+m)\log(f))$ where m is the number of the selected test points, hence these methods are also polynomially bounded. Note that the so called "minimal set" in [191] is a local minimum set.

Test points selection techniques can be classified into two categories: inclusive and exclusive [191]. For the inclusive approaches, the desired optimum set of test points is initialized to null, then new test points are added to this set if needed. For the exclusive approach, the desired optimum set is initialized to include all available test points. Then a test point is deleted if its exclusion does not degrade the degree of fault diagnosis.

For the optimum test points set to distinguish the ambiguity sets, the **integer-coded dictionary** was first proposed by Lin and Elcherif [144] and subsequently researched by Prasad and Babu [191]. This approach proved to be an effective tool for the optimum test points' selection.

4.6.1. Integer Coded Dictionary

The two-dimensional integer-coded dictionary [144], or fault-wise table [191], is constructed as follows. The rows of the integer-coded dictionary represent all the potential faults (including fault-free case) while its columns represent all the available test points. For each test point, different ambiguity sets are classified based on computer simulation. A specific integer code is then assigned to each ambiguity set. Thus, the entries of the dictionary correspond to the simulated system responses. Note that for a given test point, distinct ambiguity sets have distinct integer codes. However, the same integer code can be assigned to different ambiguity sets associated with different test points, because each test point is an independent measurement and ambiguity sets for the same test points are independent.

Let $F=\{f_0, f_1, \ldots, f_f\}$ be the set of all potential faults to be diagnosed (including the fault-free case f_0) and $N=\{n_1, n_2, \ldots, n_t\}$ be the set of all available test points where subscript f is the number of potential faults and t is the number of available test points. To define the concept of a diagnosable circuit, let us associate the measurement equivalent classes that are the elements of the dictionary A as follows:

$$\forall f_i \in F\,(0 \le i \le f) \text{ and } \forall n_j \in N\,(1 \le j \le t),\ \exists\ \phi(f_i, n_j) = a_{ij} \in A \qquad (4.146)$$

where a_{ij} is element of the dictionary A corresponding to the ith fault and j^{th} test point.

Definition 4.2:

*Let $A_j = \{a_{0j}, a_{1j}, a_{2j}, \ldots, a_{fj}\} \subset A$ be a subset of dictionary A with a single test point n_j. If for any pair of elements ($a_{pj} \in A_j$, $a_{qj} \in A_j$, $p \ne q$), we have $a_{pj} \ne a_{qj}$, then the CUT is **diagnosable** by the test point n_j.*

It would be very optimistic to expect that the CUT is diagnosable by using a single

test point n_j. Usually different faults f_p and f_q ($p \neq q$) may have identical dictionary elements $a_{pj}=a_{qj}=i$, where i is the integer code. Under such conditions, we claim that faults f_p and f_q belong to an ambiguity set S_i^j associated with test point n_j and integer code i where ambiguity set S_i^j is defined as follows:

Definition 4.3:

Ambiguity set S_i^j in the fault dictionary is defined as the set of faults with identical dictionary elements

$$S_i^j = \{ f_m \in F \,\middle|\, a_{mj} = i,\ 0 \le m \le f \} \tag{4.147}$$

An immediate proposition is obtained as a result of Definitions 16 and 17:

Proposition:

The CUT corresponding to the dictionary A is ***diagnosable*** *for a set of test points* $N_i \subset N$ *if and only if for any pair of faults* $\{ f_p \in F, f_q \in F, p \neq q \}$, *there always exists* $n_j \in N_i$ *with* $a_{pj} \neq a_{qj}$.

The purpose of test points' selection is to identify the faults using the minimum number of test points. Integer coded dictionary transforms test point selection problem into the selection of dictionary columns to isolate its rows.

4.6.2. Entropy Test Point Selection

It can be shown that the computation time to search for the global minimum set of test points is not polynomially bounded [144]. To efficiently achieve the desired degree of fault diagnosis, an alternative solution is to search for a local minimum set of test points. In [246], an entropy index computation method, which belongs to inclusive approaches, was proposed to efficiently search for a set of test points. The idea is similar to the one presented by Hartmann et al. [108] in which an entropy-based method was developed to efficiently construct a decision tree.

In [246] probability for each fault to be separated is evaluated in accordance with the cardinalities of ambiguity sets. Assume that there are k non-overlapping ambiguity sets for test points n_j, and F_i^j is the number of faults in ambiguity set S_i^j associated with test point n_j and integer code i. The probability of a fault being isolated from the ambiguity set S_i^j is approximated by F_i^j / f where f is the number of all potential faults listed in the dictionary. Suppose that the test points are independent from each other and that the faults have equal probability of occurrence. In such case the entropy based measure, or information content $I(j)$, for the specific test point n_j is expressed by

$$I(j) = -\sum_{i=1}^{k}\left(\frac{F_{ij}}{f}\log\frac{F_{ij}}{f}\right) = \frac{\log f}{f}\sum_{i=1}^{k}F_{ij} - \frac{1}{f}\sum_{i=1}^{k}F_{ij}\log F_{ij} = \log f - \frac{1}{f}E(j) \tag{4.148}$$

where the entropy index $E(j)$ is defined as

$$E(j)=\sum_{i=1}^{f}F_{ij}\log F_{ij} \tag{4.149}$$

Because the number of all faults f in a given dictionary is constant, the information content $I(j)$ for the specific test point n_j in (4.148) is maximized with the minimization of the entropy index $E(j)$. If a test point n_j with the minimum value of $E(j)$ is added to the desired test point set N_{opt}, this will guarantee the maximal increase of information in N_{opt}. Consequently, this inclusive strategy guarantees that the maximal degree of fault diagnosis is achieved at each stage of test point selection.

The problem of searching for the appropriate test point for N_{opt} is transformed to the problem of a linear search for a minimum value of $E(j)$, which can be easily and efficiently implemented.

A generalized algorithm for the proposed entropy-based method is given below.

Entropy based test point selection algorithm

Step 1: *Initialize the desired optimum set of test points N_{opt} to null.*

Step 2: *Calculate the number of faults in F_i^j $(1 \le j \le t)$ for the ambiguity set S_i^j and the test point n_j.*

Step 3: *Calculate the entropy index E(j) by (4.152) for the test point n_j and search for the minimum value of E(j) for each test point except for those already included in N_{opt}.*

Step 4: *Add the test point n_j with the minimum value of E(j) to the desired optimum set N_{opt}. n_j will not be considered for future computation of entropy indexes.*

Step 5: *If the minimum value of E(j) is zero or if the new value of E(j) is the same as the previous E(j) for all test points, then stop.*

Step 6: *Partition the rows of the dictionary according to the ambiguity sets of N_{opt} and rearrange the dictionary by removing the rows of unit size. Go to Step 2.*

Remark 1**:** In Step 6, if there are *m* ambiguity sets for the resulting optimum set N_{opt}, create *m* horizontal partitions of the dictionary.

Remark 2**:** If there is only one row in a partition in Step 6, the corresponding fault is concluded as uniquely isolated and should be removed from the dictionary. Thus the size of ambiguity sets and the size of dictionary are gradually decreasing.

Remark 3**:** In Step 5, a maximum information increase to N_{opt} is guaranteed each time by adding the test point n_j with a minimum value of $E(j)$. Such increase does not indicate that the combination of all selected test points in N_{opt} will yield the maximum information. The global minimum set of test points can only be implemented by an exhaustive search for the minimum value of the system entropy index by analyzing all combinations of test points, which is very expensive in computer resources and simulation time.

Most methods for the optimum selection of test points presented in the literature [144], [262], [250], [1], [113], are not polynomially bounded. Although the method in [192] is polynomially bounded, it is not applicable to general cases. Among the polynomially bounded algorithms in [191], the most efficient algorithm has the complexity of $O(f\,p\,\log f)$. The efficiency of the entropy-based method presented in [246] is better than all of these methods and it is limited by the numerical complexity of the sorting algorithm.

A similar entropy-based method has been presented in [199]. It exploits concept

of the information channel and minimization of the information deficit. The algorithm starts with the null set of test points. The channel inputs are circuit conditions $F=\{f_0,f_1,..,f_f\}$. Ambiguity sets of the jth measurement $S_j=\{S_{j1},S_{j2},...\}$ are the outputs. In such channel, misinformation is equal to zero, and therefore, the obtained relative information is equal to the output entropy (information content) as given by equation (4.148). This concept leads to the identical entropy index $E(j)$ as given by equation (4.149) and test point that gives the minimum index is selected at each consecutive step.

4.6.3. Statistical Experiments

The globally minimum set of test points can only be guaranteed by an exhaustive search which is computationally expensive. Any efficient polynomially bounded algorithm for test points' selection only guarantees a local minimum solution. Since no proof can be offered to demonstrate a non-exhaustive algorithm's performance, such algorithm must be tested statistically on large number of fault dictionaries in order to evaluate its computation efficiency and qualities of the generated results.

Such statistical experiments were carried out for the randomly generated integer coded dictionaries by using the proposed entropy-based algorithm, three inclusive algorithms, and one exclusive algorithm presented in [191], respectively. 200 randomly computer-generated integer coded dictionaries were generated, and every dictionary included 100 simulated faults, 30 test points and 5 ambiguity sets per test point. These dictionaries were first analyzed by the exhaustive search algorithm EXPANSION [240] which found the global minimum set of test points for each fault dictionary. The size of the global minimum test set was equal to 5 for all 200 cases. The same dictionaries were then analyzed by using the proposed entropy-based method as well as three inclusive methods and one exclusive method presented in [191], respectively. The obtained statistical results concerning the solution accuracy are shown in Table 4.1.

Table 4.1

Statistical results of the solution accuracy

	Percentage of the optimum sets with a specific size found by a specific method					
Size of the optimum sets found	EXPANSION	Entropy Method	Exclusive Method	Inclusive Method 1	Inclusive Method 2	Inclusive Method 3
5	100	35.5	1.5	0	0.5	0.5
6	0	64.5	74.0	16.5	30.0	29.0
7	0	0	24.5	47.0	49.5	50.0
8	0	0	0	32.5	17.5	19.0
9	0	0	0	4.0	2.5	1.5

The results show that the proposed method has significantly better quality in finding near-minimum solution. The global minimum sets of test points (size 5) were found by the proposed method in 35.5% of all the simulated cases, while the global minimum sets were found by the other algorithms only in 0% to 1.5% of all the simulated cases. Thus we may conclude that the entropy-based method developed in [246] has

much higher possibilities in finding the global minimum sets than the competing algorithms. In addition, 100% of the test point sets found by the developed algorithm have sizes larger by at most 1 (size 5 or 6) than the global minimum solution (size 5). For the exclusive algorithm, 24.5% of the simulated cases are found to have the solution sets with sizes larger by 2 (size 7) than the global minimum. All three inclusive algorithms are even worse, since over 70% of the simulated cases have the selected sets of test points with sizes larger by at least 2 (sizes 7, 8 or 9) than the global minimum.

Computing efficiency of the developed algorithm was compared with efficiency of the exhaustive search algorithm. The relationship between the computation time and system complexity for the exhaustive search algorithm can be seen in Fig. 1 in [240]. Computation time strongly depends on the number of potential faults (number of signals in Fig. 1 in [240]) and the number of available test points (number of attributes in Fig. 1 in [240]). For large or medium systems such as the dictionaries with more than 40 faults and more than 40 test points, the exhaustive search is impractical. Therefore, the exhaustive search algorithm is limited to small or medium size analog systems. Thus, the significance of the developed entropy-based method is that it offers a test points' selection method with high quality for large and medium size systems with a reasonable computational cost.

4.7. Summary of Topological Diagnosis Methods

Exploitation of network topology simplifies analog fault diagnosis by providing guidance to setting test points, interpreting test results, improving computational performance and removing effect of numerical roundoff errors.

Network decomposition based on block graph and measurement nodes is a very effective way to reduce computations in the analysis of large networks. Decomposition of a network into smaller subnetworks facilitated testing by localizing the effect of faults to specific subnetworks. A special version of the decomposition approach can be used for fault location in large-scale networks, in which faulty elements are located within separable regions of the tested network. Decomposition of a network into smaller subnetworks facilitated testing by localizing the effect of faults to specific subnetworks.

Network topology can be used to facilitate testing in sensitivity based diagnosis of a CUT. By using network decomposition through the measurement nodes, dense Jacobian matrix that describes relationship between the element values and the measurements can be transformed to a block diagonal form, facilitating the testing task and increasing accuracy of the test results.

Verification methods rely on an overdetermined system of equations formulated on the basis of the nominal design topology and nominal parameters values. Consistency of this system of equations is a necessary condition to identify the faulty nodes and faulty components. Various approaches can be used to formulate and solve test equations for fault verification.

Network topology can define restrictions on network testability, which is particularly important if the number of test points is small in relation to the network size. A practical problem in analog test is a limited number of independent measurements which leads to ambiguities in locating and evaluating faults.

An optimum selection of test points is the most important stage in analog fault verification and fault dictionary techniques. It determines the fault coverage, cost to build a fault dictionary, and the testing cost. The global minimum solution is only guaranteed by an exhaustive search which is NP-hard and thus it is impractical for medium or large systems.

5. Concluding Remarks

Topological methods have been shown useful in various applications in analog circuits' simulation and diagnosis. This work presents the author's view of developments in this area, stressing the results developed by the author and his coworkers.

The network topology defines fundamental properties of electronic circuits expressed by Kirchhoff laws. These properties support the discovery of many qualitative results that are unaltered by computational complexity, numerical dependencies of system equations, or roundoff errors. The presented work tries to integrate descriptions for topological methods in analysis and testing of analog and mixed signal systems. It includes topological analysis formulas and techniques appropriate to various network graphs, and computationally efficient algorithms to evaluate these formulas. Considered in this work are topological methods of circuit analysis based on signal-flow graphs, directed graphs, and conjugate graphs. These include direct and decomposition based methods and the associated algorithms.

Direct decomposition significantly improves computing efficiency of the topological methods, while hierarchical decomposition transforms topological analysis from an academic research tool to an effective tool in the hands of circuit designers. Efficient algorithms for generating all of the many kinds of multiconnections and multitrees needed in topological analysis were presented in this work.

Topological methods were also used in this work to illustrate analysis of high frequency interconnects, and analysis of decomposed networks with the help of the large change sensitivity approach. The same tools of topological analysis as are used in stationary analog circuits are applied to the development of element models and topological formulas for switched capacitor networks. Analyses of large scale interconnect networks benefit from their regular, hierarchical structures, delivering accurate results in very short simulation times. These analyses are more accurate and faster than analyses of interconnect networks based on a popular method of moments [187]. The large change sensitivity method, combined with hierarchical decomposition of the network graph and use of the ideal switch model, led to a simple explanation of hierarchical analysis of large, sparse networks.

Topological diagnosis is a second, strong feature of the presented work. Such issues as fault location in linear and nonlinear circuits, sensitivity based testing, fault verification approaches, test point selection, and issues related to low testability circuits are all discussed. Network topology is used to decompose the network hierarchically into smaller subnetworks and, using Kirchhoff laws as test equations, to detect the existence of faults. Faults are localized to within a subnetwork level and either a subnetwork is declared fault free or faulty, aiding the fault diagnosis process.

In faulty networks testing can proceed to remote regions under topological conditions of circuit testability. Next, topological conditions for node fault diagnosis are presented. This is followed by the consideration of parameter tolerances and a specialized multiport approach to fault location. Subsequently, the network topology is used to improve the computer efficiency and accuracy of the popular sensitivity based

testing by developing a decomposed form of the test equations. This results in a sparsity of the corresponding Jacobian matrix and significant savings in testing time.

A separate section is devoted to multiple fault verification techniques that utilize the network topology to identify the parameter values. These techniques are viable alternatives to fault dictionary testing, providing effective means to diagnose multiple faults without the overhead of generating large dictionaries. The large change sensitivity approach is used once more to make the fault verification techniques more effective by directly locating faulty elements within the tested circuit.

An important question in fault diagnosis techniques is how to test low testability circuits for which test equations do not have unique solution. The approach presented makes practical use of the ambiguity groups to locate fault free nodes. This facilitates testing of ambiguous circuits, and in many cases yields a unique solution to otherwise unsolvable test equations.

In the final section, devoted to fault diagnosis, an efficient entropy based test point selection is presented. This method optimizes test point selection to minimize the testing cost – a critical aspect of industrial test procedures. It also leads to more efficient and smaller fault dictionaries.

Let us summarize this discussion with few concluding remarks. The network topology enforces a system of constraints that define the network properties. Proper understanding of these constraints and their proper use in solving network related problems are critical to efficient system level design, analysis, and testing. This work addresses some of the approaches applied to analog circuit analysis and diagnosis, and demonstrates the benefits of considering these topologically defined constrains.

References

1. Abderrahman A., Cerny E., Kaminska B.: Optimization-based multifrequency test generation for analog circuits, J. Electronic Testing: Theory Applicat., vol. 9, pp. 59–73, 1996.
2. Acar C.: New expansion for signal-flow-graph determinant, Electr. Letters, 1971.
3. Ackland B.D., Clark R.A.: Event-EMU: An event driven timing simulator for MOS VLSI circuits, Proc. IEEE Int. Conf. Comp. Aided Design, pp. 80-83, 1989.
4. Alaybeyi, M.M. Lee, J.Y. Rohrer, R.A. : Numerical integration algorithms and asymptotic waveform evaluation(AWE), Proc. Int. Conf. Computer-Aided Design, ICCAD-92, pp. 76-79, Santa Clara, 1992.
5. Alderson G.E., Lin P.M.: Computer generation of symbolic network functions - a new theory and implementation, IEEE Trans. on CT, vol. CT-20, no 1, 1973.
6. Algorithm 459, The elementary circuits of a graph Comm. ACM 16, pp. 632-633, 1973.
7. Alippi C., Catelani M., Fort A.,Mugnaini M.: SBT soft fault diagnosis in analog electronic circuits: a sensitivity-based approach by randomized algorithms, IEEE Transactions on Instrumentation and Measurement, vol. 51, no 5, pp. 1116–1125, 2002.
8. Amadori M., Guerrieri R., Malavasi E.: Symbolic analysis of simplified transfer functions, Kluwer J. Analog Integrated Circuits and Signal Process., vol. 3, no 1, pp. 9-29, 1993.
9. Bandler J.W., Biernacki R.M. and Salama A.E.: A linear programming approach to fault location in analog circuits, Proc. IEEE Int. Symp. Circuits and Systems (Chicago, IL), pp. 256-260, 1981.
10. Bandler J.W., Biernacki R.M., Starzyk J.A., Salama A.E.: Fault isolation in linear analog circuits, using the L1 norm, in Proc. IEEE Inc. Symp. Circuits and Systems (Rome, Italy 1982), pp. 1140-1143, 1981.
11. Bandler J.W., Salama A.E.: Fault diagnosis of analog circuits, Proc. IEEE, vol. 73, pp. 1279-1325, 1985.
12. Bandler J.W., Zhang Q.J.: Large change sensitivity analysis in linear systems using generalized householder formulae, IEEE Trans. Circuit Theory and Applications, vol. 14, pp. 89-101, 1986.
13. Bellert S., Traczyk T.: Representation of rings of structural numbers, Demonstratio Mathematica, t. V, 1973.
14. Bellert S., Wojciechowski J.: Grafy blokowe i liczby strukturalne drugiej kategorii, Prace naukowe P.W., Seria Elektronika, nrl, 1972.
15. Bellert S., Wozniacki H.: Analiza i synteza układów elektrycznych metoda, liczb strukturalnych, WNT, Warszawa, 1968.
16. Bellert S.: Topological analysis and synthesis of linear systems, Journal of the Franklin Inst., vol. 274, 1962.
17. Bellert S.: Topological considerations and synthesis of linear networks by means of the method of structural numbers, Arch. Elektr., t.XII, z.3, 1963.
18. Berge C.: Graphs and Hypergraphs, Amsterdam, North-Holland, 1973.
19. Berkowitz R.S.: Conditions for network-element-value solvability, IEEE Trans. Circuit Theory, vol. CT-9, pp. 24–29, 1962.

20. Biernacki R.M. and Bandler J.W.: Multiple-fault location of analog circuits, IEEE Trans. Circuits Syst., vol. CAS-28, pp. 361-367,198l.

21. Biernacki R.M. Starzyk J.A.: A test generation algorithm for parameter identification of analog circuits, Proc. European Conf. Circuit Theory and Design, The Hague. Netherlands, pp. 993-997, 1981.

22. Bowen M., Smith G.: Considerations for the design of smart sensors, Sensors and Actuators - A - Physical, vol. 47, Issue: 1/3, pp. 516-520, 1995.

23. Branin F.H.: A sparse matrix modification of Kron's method piecewise analysis, IEEE Int. Symp. on CAS, Newton, 1975.

24. Braun J.: Analytical methods in active network theory. Acta Polytechnica - Prace CVUT v Praze. IV. I, 1966.

25. Braun J.: Topological analysis of networks containing nullators and norators, Electr. Lett., pp. 427, 1966.

26. Brownell R.A.: Growing the Trees of a graph, Proc. of IEEE vol. 56, pp. 1121-1123, 1968.

27. Brtkovic F., Vehovec M.: Matrix reduction method for solution of sparse circuit equations, Proc. IEEE Int. Symp. on CAS, pp.175-179, 1978.

28. Buchanan J.E.: BiCMOS/CMOS Systems Design, McGraw-Hill, New York, 1990.

29. Cajka J.: New formula for the signal-flow graph reduction, Electr. Lett., pp. 437-438, July, 1970.

30. Callier F.M., Chan W.S., Desoer C.A.: Input-output stability theory of interconnected systems using decomposition techniques, IEEE Trans. Circuits and Systems, vol. CAS-23, pp. 714-729, 1976.

31. Carlin H.J.: Singular network elements, IEEE Trans. Circuit Theory, 1964.

32. Carmassi R., Catelani M., Iuculano G., Liberatore A., Manetti S., Marini M.: Analog network testability measurement: A symbolic formulation approach, IEEE Trans. Instrum. Meas., vol. 40, pp. 930–935, 1991.

33. Catelani M., Fedi G., Luchetta A., Manetti S., Marini M., Piccirilli M.C.: A new symbolic approach for testability measurement of analog networks , Proc. MELECON '96 , pp. 517–520, Bari, Italy, May 1996.

34. Catelani M., Fort A.: Soft fault detection and isolation in analog circuits: some results and a comparison between a fuzzy approach and radial basis function networks, IEEE Transactions on Instrumentation and Measurement, vol. 51, no 2, pp. 196–202, 2002.

35. Catelani M., Giraldi S.: A measurement system for fault detection and fault isolation of analog circuits, Measurement, vol. 25, Issue: 2, pp. 115-122, 1999.

36. Catelani M., Iuculano G., Liberatore A., Manetti S., Marini M.: Improvements to numerical testability evaluation, IEEE Trans. Instrum. Meas., vol. IM-36, pp. 902–907, 1987.

37. Centkowski G., Konczykowska A., Starzyk J., Śliwa E.: Modernizacja programu HADEN z uwzględnieniem analizy hierarchicznej wstępującej, analizy tolerancji z możliwością analizy układow z przełączanymi pojemnościami, Problem badań podstawowych nr 1.11.07.02,Warszawa, 1980.

38. Centkowski G., Starzyk J., Śliwa E.: Computer implementation of topological method in the analysis of large networks, Proc. of the ECCTD Warszawa, 1980.

39. Centkowski G., Starzyk J., Śliwa E.: Program analizy topologicznej dużych układów liniowych, Raport z problemu badan podstawowych nr 1.11.07.02, Warszawa, 1979.

40. Centkowski G., Starzyk J., Śliwa E.: Symbolic analysis of large LLS networks by means of upward hierarchical analysis. ECCTD, The Hague, 1981.

41. Chen C.T.: Introduction to linear systems theory, Holt, Rinehard and Winston 1970.

42. Chen H.M.S., Saeks R.: A search algorithm for the solution of multifrequency fault diagnosis equations, IEEE Trans. Circuits Syst., vol. CAS-26, pp. 589–594, 1979.

43. Chen W.K: Flow graphs: some properties and methods of simplifications, IEEE Trans. on CT 1, pp. 128-130, 1965.

44. Chen W.K., Li H.C.: Computer generation of directed trees and complete trees, Int. J. Electronics, vol. 34, no 1, pp. 1-21, 1973.

45. Chen W.K.: Applied graph theory - graphs and electrical networks, North-Holland P.C., 1976.

46. Chen W.K.: Characterizations of complete directed trees and two-trees, IEEE Trans. on Circuit Theory, vol. CT-19, no 3 pp. 241-247, 1972.

47. Chen W.K.: On directed trees and directed k-trees of a digraph and their generation, SIAM J. Appl. Math. vol. 14, no 3, 1966.

48. Chen W.K.: Topological analysis for active networks, IEEE Trans. CT-12, 1965.

49. Chen W.K.: Unified theory on the generation of trees of a graph, Int. J. Electr., The Wang algebra formulation, vol.27 no 2, 1969, Part.II. The matrix formulation, vol.27 no 4, 1969, Part. III. Decomposition and elementary transformations, Vol.31, no 4, 1971.

50. Cheng Y., Lin P.M.: Symbolic analysis of general switched capacitor networks-New methods and implementation, Proc. IEEE Int. Symp. Circuits Syst., pp. 55-59, 1987.

51. Chow Y., Cassignol E.: Linear signal-flow graphs and applications, Wiley, New York 1962.

52. Christian D., Hassan I.: A BIST-DFT technique for DC test of analog modules, Journal of Electronic Testing: Theory and Applications, vol. 9, Issue: 1/2, pp. 117-133, 1996.

53. Christofides N.: Graph Theory – an algorithmic approach, Academic Press, 1975.

54. Chu T.: Synthesis of Self-timed VLSI Circuits from Graph Theoretic Specifications, Technical Report: TR-393, Massachusetts Institute of Technology, 1987.

55. Chua L.O., Chen L.K.: Diakoptic and generalized hybrid analysis, IEEE Trans. on CAS, vol.CAS-23, no 12, pp. 694-705, 1976.

56. Chua L.O., Lin P.M.: Computer aided analysis of electronic circuits-algorithms and computational techniques, Prentice-Hall Inc., 1975.

57. Coates C.L.: Flow graph solutions of linear algebraic equations, IRE Trans. CT, pp. 170-187, 1959.

58. Coates C.L.: General topological formulas for linear networks functions, IRE Trans. CT., vol. 5, 1958.

59. Csiszar, I. Korner, J.: Graph decomposition: A new key to coding theorems, IEEE Transactions on Information Theory, vol. 27, no 1, 1981.

60. Dantzig G.B., Wolfe P.: The decomposition algorithm for linear programming, Econometria, vol. 29, no 4, pp. 767-778, 1961.

61. Davies A.C.: Nullator-norator equivalent networks for controlled sources, Proc. IEEE, pp. 722-723, 1967.

62. Davies A.C.: On matrix analysis of networks containing nullators and norators, Electronic Letters, pp. 48, 1966.

63. Davies A.C.: Topological solutions of networks containing nullators and norators, Electr. Lett., pp. 90,1966.

64. Deo N.: Graph theory with applications to engineering and computer sciences, Englewood Cliffs, NJ, Prentice-Hall, 1974.

65. Deo N.: Teoria grafów i jej zastosowania w technice i informatyce, PWN, Warszawa, 1980.

66. Desai M. P., Narayanan H., Patkar S.B.: The realization of finite state machines by decomposition and the principal lattice of partitions of a submodular function, Discrete Applied Mathematics, vol. 131, no 2, pp. 299-310, 2003.

67. Dor D., Tarsi M.: Graph decomposition is NP-complete: A complete proof of Holyer's conjecture, SIAM Journal on Computing, vol. 26, no 4, pp. 1166-1187, 1997.

68. Duffin R.J.: An analysis of Wang algebra of network, Trans. Am. Math. Soc., 1959.

69. Duhamel P., Rault J.C.: Automatic test generation techniques for analog circuits and systems: A review, IEEE Transactions on Circuits and Systems, vol. CAS-26, pp. 411-440, 1979.

70. Dunn W.R., Chan S.P.: Analysis of active networks by a subgraph-construction technique, Second Asilomar Conf. on CAS, 1968.

71. Dunn W.R., Jr., Chan S.P.: Topological formulation of active network functions, IEEE Trans. Circuit Theory, vol. CT-13, pp. 554-557, 1971.

72. Ehrenfeucht A., Harju T, Rozenberg G.: Gene assembly through cyclic graph decomposition, Theoretical Computer Science, vol. 281, no 1-2, pp. 325-349, Elsevier, 2002.

73. Elmore W.C.: The transient response of dumped linear networks with particular regard to wide-band amplifiers, J. Appl. Phys., vol. 19, pp. 5563, 1948.

74. Engel A.B., Mlynski D.A.: Maximal partitioning of graphs, Proc. IEEE Int. Symp. on CAS, Tokyo, pp. 84-87, 1979.

75. Engel A.E., Mlynski D.A.: Cliques and partition of graphs, Proc. IEEE Int. Symp. on CAS, pp. 81-82, 1978.

76. Fedi G., Giomi R., Luchetta A., Manetti S., Piccirilli M.: On the application of symbolic techniques to the multiple fault location in low testability analog circuits, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no 10, pp. 1383-1388, 1998.

77. Fedi G., Manetti S., Piccirilli M., Starzyk J.: Determination of an optimum set of testable components in the fault diagnosis of analog linear circuits, IEEE Transactions on Circuit and Systems I: Fundamental Theory and Applications, vol. 46, no 7, pp. 779-787, 1999.

78. Fedi G., Manetti S., Piccirilli M.C.: Comments on Linear circuit fault diagnosis using neuromorphic analyzers, IEEE Trans. Circuits Syst. II, vol. 46, pp. 483–485, 1999.

79. Feng L., Zhenghui L., William L.T.: A uniform approach to mixed-signal circuit test, Int. Journal of Circuit Theory and Applications, vol. 25, Issue: 2, pp. 81 – 93, Mar./Apr. 1997.

80. Fernandez F. V., Oscar Guerra O., Rodrıguez-Garcıa J. D., Rodrıguez-Vazquez A.: Symbolic Analysis of Large Analog Integrated Circuits: The Numerical Reference Generation Problem, IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no 10, pp. 1351-1361, 1998.

81. Ferrari D.: Improving locality by critical working sets, Comm. of ACM vol. 17, no 11, pp. 614-620, 1974.

82. Filippetti F., Artioli M.: Cycling verify: fault diagnosis for linear analog circuits based on symbolic calculus and interval algebra, Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, vol. 1, pp. 589 -594, 2002.

83. Fidler J.K., Sewell J.I.: Symbolic analysis for computer-aided design - The interpolative approach, IEEE Trans. on Circuit Theory, vol.CT. 20, pp.738-741, Nov. 1973.

84. Fino M.H., Franca J.E., Steiger-Garcao A.: Automatic symbolic analysis of switched-capacitor filtering networks using signal flow graphs, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 14, no 7, pp. 858 – 867, 1995.

85. Frederickson G.N.: Planar graph decomposition and all pairs shortest paths, Journal of the ACM, vol. 38, no 1, pp. 162-204, 1991.

86. Fujiwara H.: Logic Testing and Design for Testability, MIT Press, Cambridge, MA, 1985.

87. Gandhi B.R.M., Rao V.P., Raju G.S.: Passive and active circuit analysis by the method of structural numbers, Int. J. Electr., vol. 32, no 6, 1972.

88. Gandhi B.R.M., Rao V.P.: Unistor graphs a systematic procedure for generation of directed trees and 2-trees, J.I.E. (India), vol. 50, no 8, 1970.

89. Gibbs N.E.: A cycle generation algorithm for finite undirected linear graphs, Operat. Res. vol. 17, pp. 838- -847, 1969.

90. Gielen G., Sansen W.: Symbolic Analysis for- Automated Design of Analog Integrated Circuits. Norwell, MA: Kluwer, 1991.

91. Gielen G., Walscharts H., Sansen W.: ISAAC: A symbolic simulator for analog integrated circuits, IEEE J. Solid-State Circuits, vol. 24, no. 6, pp. 1587-1597, 1989.

92. Gielen G., Wambacq P., Sansen W.M.: Symbolic analysis methods and applications for analog circuits: a tutorial overview, Proceedings of the IEEE, vol. 82, no 2, pp. 287-304, 1994.

93. Gorsztein L.L.: O Razbijenii grafow. Izw. AN SSSR - Techniczeskaja Kibernetika, no 1, pp. 98-101, 1969.

94. Goto S., Sangiovanni-Vincentelli A.: A new decomposition algorithm for the shortest path problem, Proc. IEEE Int. Symp. on CAS, pp. 653-656, Tokyo, 1979.

95. Gower J.C.: Comparison of some methods of cluster analysis, Biometrics, vol. 54, pp. 623-637, 1967.

96. Gross J.L., Yellen J.: Graph Theory and Its Applications, Chapman & Hall/CRC; 2 ed., Sept. 22, 2005.

97. Guardabassi G., Sangiovanni-Vincentelli A.: A two levels algorithm for tearing, IEEE Trans. Circuits and Systems, vol. CAS-23, pp. 783-791, 1976.

98. Gupta H., Bandler J.W., Starzyk J.A., Sharma J.: A Hierarchical Decomposition Approach for Network Analysis, Proc. IEEE Int. Symp. Circuits and Systems (Rome, 1982), pp. 643-646.

99. Hajj I.N.: Sparsity considerations in network solution by tearing, IEEE Trans., CAS-27, pp. 357-366, 1980.

100. Hakimi S.L., Green D.G.: Generation and realization of trees and k-trees, IEE Trans. on CT-11, 1964.

101. Haley S.B., Pham B.V.: Nonlinear system analysis: Computationally efficient modification algorithms, IEEE Trans., CAS-34, pp.639-649, 1987.

102. Haley, S.B., Current, K.W.: Response change in linearized circuits and systems: Computational algorithms and applications, Proc. IEEE, 1985, vol. 73, pp. 5-24.

103. Hankley W.J. and Merrill H.M.: A pattern recognition technique for system error analysis, IEEE Trans. Reliability, vol. R-20, pp. 148-153, 1971.

104. Happ H.H.: Diakoptics and networks, New York: Academic Press,1971.

105. Happ H.H.: Diakoptics - the solution of system problems by tearing, Proc. of IEEE vol.62, pp. 930-940, July 1974.

106. Happ W.W.: Flowgraph Techniques for Closed Systems, IEEE Trans. AES-2, no 3, pp. 252-264, 1966.

107. Harjani R.: Design mixed-signal ICs, IEEE spectrum, pp. 49-51, Nov. 1992.

108. Hartmann C.R., Varshney P.K., Mehrotra K.G., Gerberich C.: Application of information theory to the construction of efficient decision trees, IEEE Trans. Inform. Theory, vol. IT-28, pp. 565–576, 1982.

109. Hashemian R.: Symbolic representation of network transfer functions using norator-nullator pairs, Electronic Circuits and Systems, vol. 1, no 6, pp. 193-197, 1977.

110. Hassoun M.M., McCarville K.S.: Symbolic analysis of large-scale networks using a hierarchical signal flowgraph approach, "Analog Integrated Circuits and Signal Processing", Springer Netherlands, vol. 3, no 1, pp. 31-42, Jan. 1993.

111. Hassoun M.M., Lin P.M.: A new network approach to symbolic simulation of large-scale networks, Proc. ISCAS, pp. 806-809, 1989.

112. Ho J.K., Loute E.: An advanced implementation of the Dantzig-Wolfe decomposition algorithm for linear programming, Mathematical Programming vol. 20, pp. 303-326, 1981.

113. Hochwald W. and Bastian J.D.: A dc approach for analog fault dictionary determination, IEEE Trans. Circuits Syst., vol. CAS-26, pp. 523-529, 1979.

114. Hopcroft J.E., Tarjan R.E.: Dividing a graph into triconnected components, SIAM J. Comput., vol. 2, no 3, 1973.

115. Hsu J.J., Sechen C.: Fully symbolic analysis of large analog integrated circuits, Proc. of the 1994 Custom Integrated Circuits Conf., pp. 457-460, 1994.

116. Hu T.C.: Integer programming and network flow Reading, MA, Addison-Wesley, 1969.

117. Huang Z.F., Lin C.S., Liu R.W.: Topological conditions on multiple fault testability of analog circuits, Proc. IEEE Int. Symp. Circuits and Systems (Rome, Italy), pp. 1152-1155, 1982.

118. Huelsman L.: Personal computer symbolic analysis programs for undergraduate engineering courses, Proc. ISCAS, pp. 798-801, 1989.

119. Jony M.T., Zobrist G.W.: Topological formulas for general linear networks, IEEE Trans. Circuit Theory, vol. CT 15, pp. 251-259, 1968.

120. Katzenelson J., Tsur S.: On the symbolic analysis of linear networks, IEEE Trans. on CT, vol. CT-20, no 5, 1973.

121. Kaufman I.: On poles and zeros of linear systems, IEEE Trans. on CAS, vol. 20, no 2, pp. 93-101, March 1973.

122. Kernighas B.W., Lin S.: An efficient heuristic procedure for partitioning graphs, Bell System Tech. J., vol. 49, no 2, pp. 291-307, 1970.

123. Kirchhoff G.: On the solution of the equations obtained from the investigation of the linear distribution of galvanic currents, IRE Trans. CT-5, 1958 (tlum. pracy z 1947).

124. Kishi, Kajitani: Topological considerations on minimal sets of variables describing an LCR network, IEEE Trans. on CT, 1973.

125. Kitozawa H., Sagawa M.: P-parameter: a new representation of linear network functions and its application to network analysis, Proc. Int. Symp. CAS, pp. 1048-1051, Tokyo, 1979.

126. Ko M-Y., Murthy P.K., Bhattacharyya1 S.S.: Compact Procedural Implementation in DSP Software Synthesis through Recursive Graph Decomposition, "Software and Compilers for Embedded Systems", (Lecture Notes in Computer Science 3199), pp. 47-61, Springer Verlag 2004.

127. Kobylarz T., Nadratowski J.: An algorithm for the generation of nonduplicate complete trees, Fifth Asilomar Conf. on CAS, Pacific Grove, Nov. 1971.

128. Konczykowska A., Bon M.: Analog design optimization using symbolic approach, Proc. ISCAS, 1991, pp. 786-789, 1991.

129. Konczykowska A., Bon M.: Automated design software for switched-capacitor IC's with symbolic simulator SCYMBAL, Proc. DAC, pp. 363-368, 1988.

130. Konczykowska A., Bon M.: Symbolic simulation for efficient repetitive analysis and artificial intelligence techniques in C.A.D., Proc. ISCAS, pp. 802-805, 1989.

131. Konczykowska A., Bon M.: Topological analysis of switched-capacitor networks, Electr. Letters, vol. 16, pp. 89-90, 1980.

132. Konczykowska A., Starzyk J.: Computer analysis of large signal flowgraphs by hierarchical decomposition method, Proc. of the ECCTD, Warszawa, 1980.

133. Konczykowska A., Starzyk J.: Hierarchical decomposition of signal-flow graphs, III Int. Conf. on Electronic Circuits, Prague, 1979.

134. Konczykowska A., Starzyk J.: Wyznaczanie liczby strukturalnej grafu zdekomponowanego, Czesc I i II, Arch. Elektrot., z.2, 1975.

135. Konczykowska A.: Analiza ukladow elektrycznych z zastosowaniem metod dekompozycji, Praca doktorska, Warszawa, 1976.

136. Konczykowska A.: Computer program of analysis of decomposed networks, Proc. of the ECCTD, Geneva, 1976.

137. Kotkowski J.: Komputerowa analiza uklad6w z przelaczanymi pojemnosciami, Praca dyplomowa, IPE, Politechnika, Warszawska, Warszawa, 1981.

138. Kron G.: Diakoptics: the piecewise solution of large scale systems, London, Me Donald, 1963.

139. Kuh B.S., Rohrer R.A.: The state-variable approach to network analysis, Proc. IEEE, vol. 53, no 3, pp. 672-686, July 1965.

140. Li B., Gu D.: SSCNAP: A program for symbolic analysis of switched-capacitor circuits, IEEE Trans. Computer-Aided Design, vol. 11, pp. 334-340, 1992.

141. Li F. and Woo P.: The invariance of node-voltage sensitivity sequence and its application in a unified fault detection dictionary method, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Application, vol. 46, pp. 1222-1227, 1999.

142. Liberatore A., Manetti S., Piccirilli M.C.: A new efficient method for analog circuit testability measurement, Proc. IEEE Instrumentation Measurement Technology Conf. , pp. 193–196, Hamamatsu, Japan, May 1994.

143. Lin P.: Symbolic Network Analysis, Elsevier, Amsterdam, The Netherlands, 1991.

144. Lin P.M., Elcherif Y.S.: Analogue circuits fault dictionary – New approaches and implementation, Int. J. Circuit Theory Appl., vol. 13, pp. 149–172, 1985.

145. Lin P.M.: A survey of applications of symbolic network functions, IEEE Trans. on CAS-20, pp. 732-737, 1973.

146. Lin P.M.: Alderson G.E.: SNAP - A computer program for generating symbolic network functions, School Elec. Eng.Purdue Univ. Lafayette, Ind. Techn. Rep. TR-EE 70-16, Aug. 1970.

147. Lin S., Kuh E.S.: Pade approximation applied to lossy transmission line circuit simulation, Proc. IEEE Int. Symp. Circuits and Syst. pp. 93-96, 1992.

148. Liu D. and Starzyk J.A.: A generalized fault diagnosis in dynamic analogue circuits, International Journal of Circuit Theory and Applications, vol. 30, no 5, pp. 487-510, 2002.

149. Liu R.W.: Testing and Diagnosis of Analog Circuits and Systems, Van Nostrand Reinhold, 1991.

150. Lubaszewski M. Courtois B.: On the Design of Self-Checking Boundary Scannable Boards, Proc. Int. Test Conf. , pp. 372-281, 1992.

151. Luccio F., Sami M.: On the decomposition of networks in minimally interconnected subnetworks, IEEE Trans. Circuit Theory, vol. CT-16, pp. 184-183, 1969.

152. Manetti S., Piccirilli M., Liberatore A.: Automatic test point selection for linear analog network fault diagnosis, in Proc. IEEE Int. Symp. Circuits and Systems, vol. 1, pp. 25–28, 1990.

153. Manetti S., Piccirilli M.: A singular-value decomposition approach for ambiguity group determination in analog circuits, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 50, no 4, pp. 477-487, 2003.

154. Manetti S.: New approaches to automatic symbolic analysis of electric circuits, Proc. Inst. Elec. Eng. part G, pp. 22-28, 1991.

155. Mann H., Kvasil J.: An algorithm for the formulation of state-space equations, Proc. of ISCAS, pp.161-162, Tokyo, 1979.

156. Marinov C.A., Neittaanamaki P.: A theory of electrical circuits with resistively coupled distributed structures: Delay time predicting, IEEE Trans. Circuits and Systems, vol. CAS-35, pp. 173-183, 1988.

157. Mason S.J.: Feedback theory - further properties of signal flow graphs, Proc. IRE, vol. 44 pp. 920-926, 1956.

158. Mason S.J.: Feedback theory - some properties of signal flow graphs, Proc. IRE, vol. 41 pp. 1144-1156, 1953.

159. Mason S.J.: Topological analysis of linear nonreciprocal networks, Proc. IRE, vol. 45, pp. 829-338, 1957.

160. Mayeda W., Kahimi S.L., Chen W.K., Deo N.: Generation of complete trees, Trans. IEEE CT-15, no 2, 1968.

161. Mayeda W., Seshu S.: Generation of trees without duplications, IEEE Trans. CT-12, 1965.

162. Mayeda W.: Graph Theory, New York, Wiley-Interscience, 1972.

163. Mayeda W.: Topological formulas for active net-world, Int. Tech. Rpt. Nr 8, U.S. Army Contract nr DA-11-022-ORO-1983, Univ. of Illinois, Jan. 1958.

164. Maxwell L. M., Reed M.B.: The theory of graphs - a basic for network theory, Pergamon Press Inc. N.Y. 1971.

165. Mc Ilroy M.D.: Algorithm 354: Generator of Spanning Trees, Comm. ACM, vol. 12, no 9, pp. 511, 1969.

166. McCormick S.P., Allen J.: Waveform moment methods for improved interconnection analysis, Proc. 27th ACM/IEEE Design Autom. Conf., pp. 406-412, June 1990.

167. Merrill H.M.: Failure diagnosis using quadratic programming, IEEE Trans. Reliability, vol. R-22, pp. 207-213, 1973.

168. Mielke R.R.: A new signal flow graph formulation of symbolic network functions, IEEE Trans. CAS, vol. CAS-25, PP. 334-340, 1978.

169. Minty G.J.: A simple algorithm for listing all the trees of a graph, IEEE Trans. CT, vol. CT-12, no 1, pp. 120, 1965.

170. Mulawka J.: Biwezlowe grafy Masona i ich zastosowanie do analizy i syntezy ukladow RC ze wzmacniaczami operacyjnymi, Rozprawa doktorska, Warszawa, 1976.

171. Nadratowski J.: Analiza liniowych ukladow elektronicznych w oparciu o metody topologiczne i techniki maszynowego liczenia, Rozprawa doktorska, Warszawa, 1971.

172. Narayanan H.: A Theorem on graphs and its application to network analysis, Proc. Int. Symp. Circuits and Systems, pp. 1007-1011, Tokyo, 1979.

173. Nathan A.: Topological rules for linear networks, IEEE CT-12, no 3, 1965.

174. Newton A.R.: Techniques for the Simulation of Large-Scale Integrated Circuits, IEEE Transactions on CAS, vol. CAS-26 no 9, Sept. 1979.

175. Nicholson H.: Structure of interconnected systems, Peter Peregrines LTD, 1978.

176. Nievergelt J., Farrar J.C., Reingold E.M.: Informatyczne rozwiazywanie zadan matematycznych, WNT, Warszawa, 1978.

177. Nitta T., Kishima A.: State equation of linear networks with dependent sources based on net topology, Trans. Electr. Commun. Japan, vol.160-A, pp. 1046-1053, 1977.

178. Numata J., Iri M.: Mixed-type topological formulas for general linear networks, IEEE Trans. on CT, vol. CT-20, no 5, 1973.

179. O'Brian P.R., Savarino T.L.: Modeling of the driving-point characteristic of resistive interconnect for accurate delay estimation, Proc. IEEE Int. Conf Comp. Aided Design, pp. 512-515, Nov. 1989.

180. Ogbuobiri E.C., Tinney W.F., Walker J.W.: Sparsity-directed decomposition for Gaussian elimination on matrices, IEEE Trans. Power, Appr. Syst., vol. PAS-89, pp. 141-150, 1970.

181. Ozguner U.: Signal flow graph analysis using only loops, Electr. Letters, pp. 359-360, 1973.

182. Pang J., Starzyk J.A.: Fault Diagnosis in Mixed-Signal Low Testability System, An Int. Journal of Analog Integrated Circuits and Signal Processing, vol. 28, no 2, pp. 159-170, 2001.

183. Paul A.J. Jr.: Generation of directed trees and 2-trees without duplication, IEEE Trans. CT-14, pp. 354, 1967.

184. Pfannkoch T.A.: Advice 1N User's Giude, TM 52173, AT&T Bell Laboratories, April 1988.

185. Pian L.: Linear active network theory, Prentice-Hall, 1962.

186. Piekarski M.: Listing of all possible trees of a linear graph, IEEE Trans. CT, vol. CT-12, pp. 124, 1965.

187. Pillage L. T., Rohrer R. A. and Visweswariah C.: Electronic and System Simulation Methods, McGraw-Hill, Inc., 1995.

188. Pillage L.T., Rohrer R.A.: Asymptotic waveform evaluation for timing analysis, IEEE Trans. Computer-Aided Design. vol. 9, pp.: 352-366, 1990.

189. Pillage L.T.: Asymptotic Waveform Evaluation for Timing Analysis, Res. Rept. CMUCAD-89-34, Carnegie-Mellon Univ., 1989.

190. Pothen A., Simon H.D., Liou K-P.: Partitioning sparse matrices with eigenvectors of graphs, SIAM Journal on Matrix Analysis and Applications, vol. 11, no 3, pp. 430-452, 1990.

191. Prasad V.C., Babu N.S.C.: Selection of test nodes for analog fault diagnosis in dictionary approach, IEEE Trans. Instrum. Meas., vol. 49, pp. 1289–1297, 2000.

192. Prasad V.C., Pinjala S.N.R.: Fast algorithms for selection of test nodes of an analog circuit using a generalized fault dictionary approach, Circuits Syst. Signal Processing, vol. 14, no. 6, pp. 707–724, 1995.

193. Rabat N.B., Hsieh H.Y.: A latent macro-modular approach to large-scale networks, IEEE Trans. on CAS, CAS-23, pp.745-752, Dec. 1976.

194. Raghunatan A., Thompson C.D.: Signal delays in RC trees with charge sharing and leakage, Proc. 19th Assilomar Conf. Circuits, Systems and Computers, pp. 557-561, Nov. 1985.

195. Ransom M.N. and Saeks R.: Fault isolation with insufficient measurements, IEEE Trans. Circuit Theory, vol. CT-20, pp. 416-417, 1970.

196. Riegle D.E., Lin P.M.: Matrix signal flow graphs and an optimum topological method for evaluating their gains, IEEE Trans. CT-19, pp. 427-435, 1972.

197. Robotycki A., Zielonko R.: Fault diagnosis of analog piecewise linear circuits based on homotopy, IEEE Transactions on Instrumentation and Measurement, vol. 51, no 4, pp. 876–881, 2002.

198. Roychowdhury J.S., Newton A.R., Pederson D.O.: An impulse response based linear time-complexity algorithm for lossy interconnect simulation, Proc. IEEE Int. Conf. Comp. Aided Design, pp. 62-65, Nov. 1991.

199. Rutkowski J.: A dc approach for analog fault dictionary determination, Proc. European Conf. Circuit Theory and Design, pp. 877–880, 1993.

200. Saeks R., Criteria for analog fault diagnosis, Proc. European Conf. Circuit Theory and Design (The Hague, Netherlands), pp. 75-78, 1981.

201. Salama A.E., Starzyk J.A. Bandler J.W.: A Unified Decomposition Approach for Fault Location in Large Analog Circuits, IEEE Trans. on Circuits and Systems, vol. CAS-31, 1984, pp.609-622.

202. Sandberg I.W., So H.C.: A two-sets of eigenvalues approach to the computer analysis of linear systems, IEEE Trans. CT 16, pp. 509-517, 1969.

203. Sangiovanni-Vincentelli A., Chen L.K., Chua L.O.: Node tearing nodal analysis, Electronics Research Laboratory, University of California, Berkeley, Memo ERL-M582, Sept. 1976.

204. Sangiovanni-Vincentelli A., Chen L.K., Chua L.O.: An efficient heuristic cluster algorithm for tearing large - scale networks, IEEE Trans. on CAS, vol. CAS-24, no 12, 1977.

205. Sannuti P., Puri N.N.: Generation of symbolic network-functions via exterior algebra, Proc. Int. Symp. on CAS, pp.190-194, New York, 1978.

206. Sannuti P., Puri N.N.: Symbolic network analysis - an algebraic formulation, Proc. IEEE Int. Symp. on CAS, pp. 1044-1047, Tokyo, 1979.
207. Sansen W., Gielen G., Walscharts H.: A symbolic simulator for analog circuits, Proc. ISSCC, pp. 204-205, 1989.
208. Seda S., Degrauwe M., Fichtner W.: A symbolic analysis tool for analog circuit design automation, Proc. ICCAD, pp. 488-491, 1988.
209. Seda S., Degrauwe M., Fichtner W.: Lazy-expansion symbolic expression approximation in SYNAP, Proc. ICCAD, pp. 310-317, 1992.
210. Sen N., Saeks R.: Fault diagnosis for linear systems via multifrequency measurement, IEEE Trans. Circuits Syst., vol. CAS-26, pp. 457–465, 1979.
211. Seppänen J.J.: Algorithm 399, Spanning tree, Comm. ACM, vol. 13, pp. 621-622, 1970.
212. Seshu S., Reed M.B.: Linear graphs and electrical networks, Addison-Wesley P.C., 1961.
213. Shi C.-J.R., Tan X-D.: Canonical symbolic analysis of large analog circuits with determinant decision diagrams, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 19, no 1, pp. 1 – 18, 2000.
214. Shiea S.D., Chan S.P.: Topological formulas of symbolic network functions for linear active networks, Proceedings ISCT, pp. 95-98, 1973.
215. Singhal K., Vlach J.: Generation of immitance functions in symbolic form for lumped-distributed active networks, IEEE Trans. on CAS, vol. CAS-21, 1, 1974.
216. Singhal K., Vlach J.: Symbolic analysis of analog and digital circuits, IEEE Trans. CAS, vol. CAS-24, pp.598-609, Nov. 1977.
217. Sliwa E.: Analiza duzych sieci liniowych reprezentowanych grafem skierowanym, Praca dyplomowa, Politechnika Warszawska, IPE, Warszawa, 1979.
218. Sorensen E.V.: A linear semisymbolic circuit analysis program based on algebraic eigenvalue technique, Report. Techn. University of Denmark, Lyngbey 1972.
219. Spaandonk J., Kevenaar T.: Iterative test-point selection for analog circuits, Proc. 14th VLSI Test Symp., pp. 66–71, 1996.
220. Spina R., Upadhyaya S.: Linear circuit fault diagnosis using neuromorphic analyzers, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 44, no 3, pp. 188-196, 1997.
221. Starzyk J.A., Konczykowska A.: Flowgraph Analysis of Large Electronic Networks, IEEE Trans. on CAS, vol. CAS-33, no 3,pp. 302-312, 1986.
222. Starzyk J.A.: Hierarchical Analysis of High Frequency Interconnect Networks, IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, vol.13, no.5, 1994, pp. 658-664.
223. Starzyk J.A., Sliwa E.: Hierarchic decomposition method for the topological analysis of electronic network, International Journal of Circuit Theory and Applications, vol. 8, pp. 407-417, 1980.
224. Starzyk J.A., Sliwa E.: Topological analysis by hierarchic decomposition method, IV International Symposium on Network Theory, Ljubljana, 1979.
225. Starzyk J.A.: An efficient cluster algorithm, Acta Polytechnica, 1981.
226. Starzyk J.A.: Analiza topologiczna duzych ukladow elektronicznych, Prace Naukowe PW, Elektronika, z. 55, 1981.

227. Starzyk J.A.: Metody topologiczne analizy akladow skupionych liniowych i stacjonarnych z nulatorami i noratorami, Prace Naukowe PW, Elektronika, nr 20, 1975.

228. Starzyk J.A.: Nowa metoda wyznaczania drzew calkowitych pary grafow sprzezonych, Arch. Elektrot. z.1, 1977.

229. Starzyk J.A.: Opredelenie polnych derewiew nulatorno-noratornogo grafa, Radioelektronika, tom XX, z.12, 1977.

230. Starzyk J.A.: Signal flow-graph analysis by decomposition method, IEE Proceedings on Electronic CAS, no 2, 1980.

231. Starzyk J.A.: The dispersor graphs, Proc. of 3-rd Czech-Polish Workshop on CT, Prenet, 1978.

232. Starzyk J.A.: Wyznaczanie drzew calkowitych metoda zmodyfikowanej macierzy strukturalnej, Archiwum Elektrot. z.4, 1978.

233. Starzyk J.A. and Bandler J.W.: Location of fault regions in analog circuits, Faculty of Engineering, McMaster University, Hamilton, Canada, Rep. SOC-281, 1981.

234. Starzyk J.A. and Dai H.: Time Domain Testing of Large Nonlinear Circuits, Proc. European Conf. Circuit Theory and Design, (Brighton, United Kingdom), 1989.

235. Starzyk J.A. El-Gamal M.A.: Topological conditions for element evaluation, 28th Midwest Symp. on Circuits and Systems Louisville Kentucky, 1985.

236. Starzyk J.A., Bandler J.W.: Multiport Approach to Multiple-Fault Location in Analog Circuits, IEEE Trans. on Circuits and Systems, vol. CAS-30, pp.762-765, 1983.

237. Starzyk J.A., Liu D.: A Decomposition Method for Analog Fault Location, IEEE Int. Symposium on Circuits and Systems, May 26-29, Scottsdale, Arizona, 2002.

238. Starzyk J.A., Liu D.: Multiple Fault Diagnosis of Analog Circuits Based on Large Change Sensitivity Analysis, Proc. ECCTD, (Espoo, Finland), Aug. 2001.

239. Starzyk J.A., Liu D.: Multiple-fault diagnosis of analog circuits by locating ambiguity groups of test equation, Proc. Int. Symp. of Circuits and Systems, May 2001.

240. Starzyk J.A., Nelson D.E., Sturtz K.: A mathematical foundation for improved reduct generation in information systems, Knowledge Inform. Syst., vol. 2, no. 2, pp. 131–146, 2000.

241. Starzyk J.A., Pang J., Manetti S., Fedi G., Piccirilli C.: Finding Ambiguity Groups in Low Testability Analog Circuits, IEEE Trans. Circuits and Systems, Part I, vol. 47, no. 8, pp. 1125-1137, 2000.

242. Starzyk J.A., Sliwa E.: Upward Topological Analysis of Large Circuits Using Directed Graph Representation, IEEE Trans. on Circuits and Systems, vol. CAS-31, pp. 410-414, 1984.

243. Starzyk J.A.: Decomposition Approach to a VLSI Symbolic Layout with Mixed Constraints, Proc. IEEE Int. Symp. Circuits and Systems, pp. 457-460, Montreal, 1984.

244. Starzyk J.A., Bandler J.W.: Nodal Approach to Multiple-Fault Location in Analog Circuits, Proc. IEEE Int. Symp. Circuits and Systems (Rome), pp. 1136-1139, 1982.

245. Starzyk J.A., Dai H.: A Decomposition Approach for Testing Large Analog Networks, Journal of Electronic Testing - Theory and Applications, no.3, pp. 181-195, 1992.

246. Starzyk J.A., Liu D., Liu Z-H., Nelson D., Rutkowski J.: Entropy-based optimum test points selection for analog fault dictionary techniques, IEEE Trans. on Instrum. and Measurement, vol. 53, no. 3, pp. 754-761, 2004.

247. Starzyk J.A., Bandler J.W.: Design of Tests for Parameter Evaluation within Remote Inaccessible Faulty Subnetworks, Proc. IEEE Int. Symp. Circuits and Systems (Newport Beach, CA), pp. 1106-1109, 1983.

248. Starzyk J.A., Biernacki R.M., Bandler J.W.: Evaluation of Faulty Elements within Linear Subnetworks, Int. Journal of Circuit Theory and Applications, vol. 12, pp. 23-37, 1984.

249. Stenbakken G., Souders T., Stewart G.: Ambiguity groups and testability, IEEE Transactions on Instrumentation and Measurements, vol. 38, pp. 941–947, 1989.

250. Stenbakken G.N., Souders T.M.: Test point selection and testability measure via QR factorization of linear models, IEEE Trans. Instrum. Meas., vol. IM-36, pp. 406–410, 1987.

251. Stenbakken G.N., Starzyk J.A.: Diakoptic and Large Change Sensitivity Analysis, IEE Proc. Part G, Circuits, Devices and Systems, vol. 139, no 1, pp. 114-118, 1992.

252. Tadeusiewicz M., Halgas S., Korzybski M.: An algorithm for soft-fault diagnosis of linear and nonlinear circuits, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 49, no 11, pp. 1648-1653, 2002.

253. Tadeusiewicz M., Korzybski M.: A method for fault diagnosis in linear electronic circuits, International Journal of Circuit Theory and Applications, vol. 28, pp. 245—262, 2000.

254. Talbot A.: Topological analysis for active networks, IEEE Trans. CT, vol. 13, 1966.

255. Tan X-D., Shi C.-J.R.: Hierarchical symbolic analysis of analog integrated circuits via determinant decision diagrams, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 19, no 4, pp. 401 – 412, 2000.

256. Thulasiraman k., Swamy M. N. S.: Graphs: Theory and Algorithms, Wiley-Interscience, 1992.

257. Tiernan J.C.: An efficient search algorithm to find the elementary circuits of a graph, Comm. ACM, vol. 13, no 12, pp. 722-726, 1970.

258. Ting S.L.: On the general properties of electrical network determinant, Chinese J. Phys., no 1, 1935.

259. T. N. Trick and Y. Li, A sensitivity based algorithm for fault isolation in analog circuits, Proc. IEEE Int. Symp. Circuits and Systems (Newport Beach, CA), pp. 1098-1101,1983.

260. Trick T.N., Mayeda W., Sakla A.A.: Calculation of parameter values from node voltage measurements, IEEE Trans. Circuits Syst., vol. CAS-26, pp. 466-474, 1979.

261. Tsai C.T.: Short cut methods for expanding the determinants involved in network problems, Chinese, J. Phys., no 3, 1939.

262. Varghese K.C., Williams J.H., Towill D.R.: Computer-aided feature selection for enhanced analog system fault location, Pattern Recognition, vol. 10, pp. 265–280, 1978.

263. Varghese K.C., Williams J.H., Towill D.R.: Simplified ATPG analog fault location via clustering and reparability technique, IEEE Trans. Circuits Syst., vol. CAS-26, pp. 496-505, 1979.

264. Various papers: special issue on partial scan methods, Journal of Electronic Testing: Theory and Applications, vol. 7, Aug./Oct. 1995.

265. Vinnakota B.: Analog and Mixed-Signal Test. Prentice Hall, 1998.

266. Visvanathan V. and Sangiovanni-Vincentelli A.: Diagnosability of nonlinear circuits and systems-Part I: The dc-case, IEEE Trans. Circuits Syst., vol. CAS-28, pp. 1093-1102, 1981.

267. Vlach J., Singhal K.: Computer Methods for Circuit Analysis and Design, Chapman & Hall, 1994.

268. Vlach M.: LU decomposition and forward-backward substitution of recursive bordered block diagonal matrices, IEE proc. Part G, 132, (I). pp. 24-31, 1985.

269. Wambacq P., Dobrovolny, P., Gielen G.G.E., Sansen W.: Symbolic analysis of large analog circuits using a sensitivity-driven enumeration of common spanning trees, IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no 10, pp. 1342-1350, 1998.

270. Wambacq P., Gielen G., Sansen W.M.: Symbolic simulation of harmonic distortion in analog integrated circuits with weak nonlinearities, Proc. ISCAS, pp. 536-539, 1990.

271. Wambacq P., Gielen G.E., Sansen W.: Symbolic network analysis methods for practical analog integrated circuits: a survey, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no 10, pp. 1331-1341, 1998.

272. Wang K.T.: On a new method for the analysis of electrical networks, Nat. Res. Inst. for Engineering, Academia Sinica Memoir, no 2, 1934.

273. Wei T., Wong M.W.T., Lee Y.S.: Efficient multifrequency analysis of fault diagnosis in analog circuits based on large change sensitivity computation, Proceedings of the 5th Asian Test Symposium, pp. 232-237, 1996.

274. Wierzba G. et al.: SSPICE-A symbolic SPICE program for linear active circuits, Proc. Midwest Symp. on Circuits and Systems, 1989.

275. Winblatt H.A.: New search algorithm for finding the simple cycles of a finite directed graph, J. ACM, no 1, pp. 43-56, 1972

276. Winklestein D., Steer M.B., Pomerleau R.: Simulation of arbitrary transmission line network with nonlinear terminations, IEEE Trans. Circuits and Syst., vol. 38, pp. 418-421, 1991.

277. Wojciechowski J.: Okreslanie liczby strukturalnej grafu blokowego metoda rozciec, Arch. Elektrot. z.2, 1976.

278. Worsman M., Wong M.W.T.: Non-linear analog circuit fault diagnosis with large change sensitivity, International Journal of Circuit Theory and Application, vol. 28, pp.281-303, 2000.

279. Worsman M., Wong M.W.T.: Nonlinear circuit fault diagnosis with large change sensitivity, IEEE Intl. Conference on Electronics, Circuits and Systems, vol. 2, pp. 225-228, 1998.

280. Wu C.-C., Nakajima K., Wey C.-L., Saeks R.: Analog fault diagnosis with failure bounds, IEEE Trans. Circuits Syst., vol. CAS-29, pp. 271-284, 1982.

281. Wu F.: Solution of large-scale networks by tearing, IEEE Trans. on CAS, vol. CAS-23, no 12, pp. 706-713, 1976.

282. Wu S. D., Byeon E-S., Storer R. H.: A Graph-Theoretic Decomposition of the Job Shop Scheduling Problem to Achieve Scheduling Robustness Operations Research, vol. 47, no 1, pp. 113-124, 1999.

283. Xie D.H., Nakhla M.: Delay and crosstalk simulation of high speed VLSI interconnects with nonlinear terminations, Proc. IEEE Int. Conf. Comp. Aided Design, pp. 66-69, Nov. 1991.

284. Yu Q., Sechen C.: A unified approach to the approximate symbolic analysis of large analog integrated circuits, IEEE Trans. Circuits Syst., vol. 43, pp. 656–669, 1996.

285. Yu Q., Sechen C.: Efficient approximation of symbolic network functions using matroid intersection algorithms, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 16, no 10, pp. 1073 – 1081, 1997.

286. Zemanian A.H.: Delay time bounds for on-chip and off-chip interconnection networks, Proc. IEEE Int. Symp. Circuits and Systems, pp. 2636-2637, Aug. 1990.

287. Zwolinski, M., Nichols, K.G.: The design of a hierarchical circuit-level simulator'. European Conference on Electronic Design Automation, University of Warwick, pp. 9-12, 1984.

Topological Analysis and Diagnosis of Analog Circuits

Summary

This work presents a comprehensive view of topological methods for analog electronic circuit analysis and diagnosis. It details a number of issues related to proper understanding and efficient use of topological methods, starting from graph representation of the network topology, topological formulas used in circuit analysis, hierarchical decomposition of topological representations, and related computational algorithms. It shows how to generate multiconnections and multitrees for various types of topological analyses of analog networks represented by their graphs. Other types of topological analyses are also discussed. These include high frequency interconnect and clock networks of modern integrated circuits, and large change sensitivity based diakoptics of large analog networks that use network topology to improve efficiency of sparse matrix analysis algorithms.

Topological techniques in analog fault diagnosis are also discussed. Network decomposition is used to effectively locate faults within subnetworks of the tested network. Fault location under parameter tolerances and corresponding topological conditions are developed for nodal and multiport representations. Network topology is beneficial to a popular sensitivity based testing by simplifying the test matrix (Jacobian matrix of the network equations). Fault diagnosis based on verification of the test equations related to the network topology has been developed. This led to formulation of ambiguity groups in low testability circuits and an effective use of the network topology to test such circuits. Final sections of this work make an efficient use of the network topology and information theory to select an optimum set of test points needed in both fault dictionary and verification methods used in analog fault diagnosis.

Topologiczna Analiza i Diagnostyka Układów Analogowych

Streszczenie

Praca przedstawia syntetyczny opis metod topologicznych stosowanych do analizy i diagnostyki elektronicznych układów analogowych. W sposób szczegółowy przedstawia zagadnienia związane z właściwym zrozumieniem i efektywnym wykorzystaniem metod topologicznych, poczynając od graficznej reprezentacji topologii układu, poprzez formuły topologiczne użyte do analizy układu i hierarchiczną dekompozycję reprezentacji topologicznych, do stowarzyszonych algorytmów obliczeniowych. Pokazuje jak generować wielo-połączenia i wielo-drzewa dla różnych typów analizy topologicznej układów analogowych reprezentowanych grafami.

Dyskutowane są też inne rodzaje analizy topologicznej. Wchodzą w to takie metody jak analiza połączeń wysokiej częstotliwości i sieci zegara nowoczesnych układów scalonych, czy też diakoptyka dużych układów analogowych w oparciu o metodę wrażliwości wielkoprzyrostowej, w której topologia układu jest wykorzystana do zwiększenia wydajności algorytmów analizy macierzy rzadkich.

Dyskutowane są też techniki topologiczne stosowane do diagnostyki układów analogowych. Dekompozycja układu użyta jest do efektywnej lokalizacji uszkodzeń wewnątrz podukładów układu testowanego. Lokalizacja uszkodzeń przy uwzględnieniu tolerancji parametrów, wraz z niezbędnymi warunkami topologicznymi, jest rozwinięta dla reprezentacji węzłowej i wielo-wrotnikowej. Topologia układu usprawnia, opartą na wrażliwościach, popularną metodę testowania poprzez uproszczenie macierzy testowej (macierz Jakobianu równań układu). Rozwinięta została diagnostyka uszkodzeń w oparciu o weryfikacje równań testowych odniesionych do topologii układu. Doprowadziło to do zdefiniowania grup wieloznacznych w układach o niskiej testowalności i efektywnego użycia topologii układu do testowania takich układów. Końcowe sekcje pracy w sposób efektywny wykorzystują topologie układu i teorię informacji do optymalnego wyboru punktów pomiarowych potrzebnych przy testowaniu układów analogowych metodami słownikowymi i weryfikacji.

WYDAWNICTWO POLITECHNIKI ŚLĄSKIEJ
ul. Akademicka 5, 44-100 Gliwice, tel./faks (0-32) 237-13-81
www.wydawnictwopolitechniki.pl

Sprzedaż i Marketing
tel. (0-32) 237-18-48
wydawnictwo_mark@polsl.pl

Nakł. 100+50 Ark. wyd. 13 Ark. druk. 8,75 Papier offset. 70x100,80g
Oddano do druku 28.12.2007 r. Podpisano do druku 28.12.2007 r. Druk ukończ. w styczniu 2008 r.

Wydrukowano w Zakładzie Graficznym Politechniki Śląskiej w Gliwicach, ul. Kujawska 1
zam. 459/07

ZESZYTY
NAUKOWE
POLITECHNIKI
ŚLĄSKIEJ

P. 4474/07

Janusz A. STARZYK

TOPOLOGICAL ANALYSIS AND DIAGNOSIS OF ANALOG CIRCUITS

GLIWICE 2007