Flowgraph Analysis of Large Electronic Networks

JANUSZ A. STARZYK, SENIOR MEMBER, IEEE, AND A. KONCZYKOWSKA

Abstract —**The paper presents a new method for signal flowgraph analysis of large electronic networks. A hierarchical decomposition approach is realized using the so-called upward analysis of the decomposed network. This approach allows fully symbolic network formulas to be obtained in time linearly proportional to the size of the network. A multiconnection characterization, suitable for upward analysis, has been defined and used in topological formulas. Examples of large scale networks analysis are discussed. The approach can be used to obtain symbolic solutions of linear systems of equations.**

I. Introduction

THE NOTION of *topological analysis* of electrical networks is concerned with the determination of the network characteristics from the knowledge of elements and their connections (*network topology*) without applying numerical methods. As a result, for linear, lumped, stationary (LLS) networks, the transfer functions (defined as the ratio of the Laplace transform of the output to the input signals under zero initial state) are obtained.

Topological methods, independently of the graph representation used, allow a network transfer function to be obtained in a rational function form. The numerator and denominator of this function are expressed as a sum of products of edges weights [3]

$$K(s)=\frac{L(s)}{M(s)}=\frac{\sum_{1}\prod_{i} y_i}{\sum_{m}\prod_{j} y_j}. \tag{1}$$

These weights depend directly on the type and value of network elements.

Realization of topological formulas requires the knowledge of all graph connections [3]. To make the computations efficient, one should use the algorithms which generate connections rapidly and without duplication. Only in this case, it is possible not to check any new connection with all previously generated ones. There are many efficient algorithms to generate graph connections [20], [25], [27]. They form the basis of topological analysis programs intended for small linear networks [14], [16].

Direct application of topological formulas permits the analysis of networks with graphs having approximately 10 nodes [6]. This limitation is not the result of the low efficiency of generation algorithms but of great number of terms in topological formulas for determinant of a coefficient matrix [3]. Even if we could generate all terms in zero time, the time needed for weights evaluation would grow at least proportionally to the number of terms, and for relatively small networks (with about 20 nodes), will attain enormous values. In any case, it is obvious that application of topological formulas for networks having more than 10 nodes is much more time consuming than the methods of symbolic analysis based on the numerical techniques of determinant evaluation [2], [26].

For these reasons the methods of topological analysis were judged by McCalla and Pederson as completely inefficient [15]. Nevertheless, research in this area has been carried out [1], [17]–[19].

Attempts to introduce methods of graph reduction [4], [9], [11] or decomposition [5], [22] to the analysis did not provide universally efficient programs and were deemed unacceptable in a paper by Alderson and Lin [2].

An important development has been achieved with the introduction of *hierarchical decomposition*. In [24] the method of signal flowgraph analysis has been presented and in [23] the hierarchical analysis of directed graphs has been discussed. Based on both these methods and downward decomposition, a program for topological analysis of large networks has been successfully developed [12].

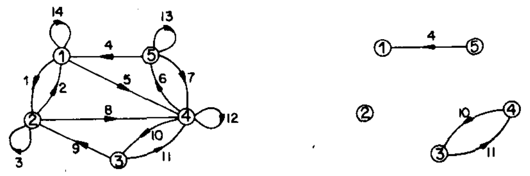
Further improvement was attained when the *upward hierarchical method* was introduced [13]. The details of the latter method will be presented in this paper. Our goal is to reduce the time consumption from the involution dependence for the previous (downward) decomposition to the linear dependency. We only consider Coates flowgraph representation of the network [3]. A similar approach is possible with other representations (e.g., unistor graph [8]).

II. Topological Formulas

The form of topological formulas is different for direct analysis and analysis with decomposition. It depends on the type of partition and on the kind of topological representation. In practice two-terminal immittances and two-port transfer functions are the most frequently calculated. As the basis for topological dependencies we consider evaluation of network immittances and transfer functions expressed by the determinant and cofactors of the nodal admittance matrix, as discussed in [3].

Manuscript received August 10, 1983; revised July 22, 1985.
J. A. Starzyk is with the Department of Electrical and Computer Engineering, Ohio University, Athens, OH 45701.
A. Konczykowska is with the Centre National d'Etudes des Telecommunication, 92220 Bagneux, France.
IEEE Log Number 8406682.

0098-4094/86/0300-0302$01.00



(a) (b)

Fig. 1. (a) Flowgraph. (b) 1-connection.

Let us denote W—a set of pairs of nodes in the Coates graph G_c

$$W=\{(v_1,r_1),\cdots,(v_k,r_k)\},\qquad v_l\neq v_m,\ v_l\neq r_m,\ r_l\neq r_m \quad \text{for } l\neq m.$$

Definition 1

We call a *k-connection* (multiconnection) of graph G_c a subgraph p_W, composed of k node-disjoint directed paths and node-disjoint directed loops incident with all graph nodes. The initial node of ith path is v_i and the terminal node is r_i (pairs of nodes from the set W).

In Fig. 1 a flowgraph and its 2-connection p_W is presented. In this case $W=\{(5,1),(2,2)\}$. A 0-connection or simply a connection is denoted by p, because $W=0$. When $v_i=r_i$, a multiconnection has the isolated node v_i. A multiconnection is a natural generalization of terms "connection" and "1-connection" defined by Coates [7] and is useful for the topological analysis of a decomposed network. This notion corresponds to that of a *k-tree* (multitree) occurring in the analysis (with decomposition) when the representation with a pair of conjugated graphs or a directed graph is used. A tree can be obtained from the k-tree by adding $k-1$ edges. Similarly, a k-connection can be transformed into a connection by adding k edges. A set of all k-connections p_W will be denoted by P_W.

Definition 2

The weight function of $|P_W|$ of a multiconnection set P_W of a Coates graph with n nodes is defined as follows:

$$|P_W|=\sum_{p\in P_W}\text{sign}\prod_{e\in p}y_e \tag{2}$$

where

$$\text{sign}\, p=(-1)^{n+k+l_p}\,\text{ord}\,(v_1,\cdots,v_k)\,\text{ord}\,(r_1,\cdots,r_k)$$

$$\text{ord}\,(x_1,x_2,\cdots,x_k)=\begin{cases}1, & \text{when the number of permutations ordering the set is even}\\ -1, & \text{otherwise}\end{cases}$$

- n number of graph nodes,
- l_p number of loops in multiconnection p,
- y_e weight of an element e.

Consider a flowgraph of a two-port network shown in Fig. 2. Let Y be an indefinite admittance matrix of the two-port. Denote Y_{uv} the first-order cofactor of Y as

$$Y_{uv}=(-1)^{u+v}\det \boldsymbol{Y}_{uv} \tag{3}$$

where $\boldsymbol{Y}_{uv}$ is the submatrix obtained from Y by deleting the uth row and vth column. The second order cofactor

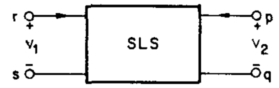


Fig. 2. Two-port.

$Y_{rp,ss}$ is defined as

$$Y_{rp,ss}=\text{sgn}\,(r-s)\,\text{sgn}\,(p-s)\,(-1)^{r+p}\det \boldsymbol{Y}_{rp,ss},\qquad r\neq s,\ p\neq s \tag{4}$$

where $\boldsymbol{Y}_{rp,ss}$ is the submatrix obtained from Y by deleting rows r and s and columns p and s. Similarly we define the third-order cofactor $Y_{pq,rr,ss}$ ($p\neq s$, $q\neq s$, $r\neq s$, $p\neq r$, $q\neq r$). Using these cofactors we can obtain formulas for transfer functions of the two-port (see [3]).

Theorem 1

Cofactors of the indefinite admittance matrix of a given multiterminal network can be expressed by the weight functions of multiconnection sets as follows:

$$Y_{uv}=|P_{\{(s,s)\}}| \tag{5}$$

$$Y_{rp,ss}=|P_{\{(r,p),(s,s)\}}| \tag{6}$$

$$Y_{pq,rr,ss}=|P_{\{(p,q),(r,r),(s,s)\}}|. \tag{7}$$

Proof:

If the Coates graph is based on $n\times n$ indefinite admittance matrix $Y=[y_{ij}]$, then its edge directed from node x_j to node x_i has the weight equal to y_{ij}. We have

$$Y=\lambda_-Y_e\lambda_+^T \tag{8}$$

where the element ij of λ_- is equal to 1 if the jth edge is directed towards the ith node and zero otherwise, and the element ij of λ_+ is equal to 1 if the jth edge is directed away from the ith vertex and is zero otherwise, and Y_e is a diagonal matrix of element admittances.

The submatrix $Y(A|B)$ obtained from Y by deleting rows represented by the set of nodes A and columns represented by the set B can be written in the form

$$Y(A|B)=\lambda_{-A}Y_e\lambda_{+B}^T \tag{9}$$

where $\lambda_{-A}(\lambda_{+B})$ is obtained from $\lambda_-(\lambda_+)$ by deleting rows $A(B)$, respectively. According to the Binet–Cauchy theorem [10] and relation (9), we have

$$\det Y\,(A|B)=\sum \det C^-\det C^+ \tag{10}$$

where C^- is a major submatrix of $\lambda_{-A}Y_e$ with order equal to (n-card A) and C^+ is the corresponding major submatrix of λ_{+B}^T. A major determinant of $\lambda_{-A}Y_e$ is different from zero if and only if there exists one nonzero element in every row of the chosen submatrix C^-. This corresponds to the set of (n-card A) edges, such that every edge has a different terminal node from the set of nodes $(N-A)$, where N indicates the set of all nodes of the Coates graph. The corresponding submatrix C^+ is different from zero if the same edges have different initial nodes from the set of nodes $(N-B)$. Now it is easy to check that these edges form a multiconnection p_W, such that if $(v_i,r_i)\in W$ then $v_i\in A$ and $r_i\in B$ (see Fig. 3). Formulas (5), (6), and (7) follow from this general observation. □

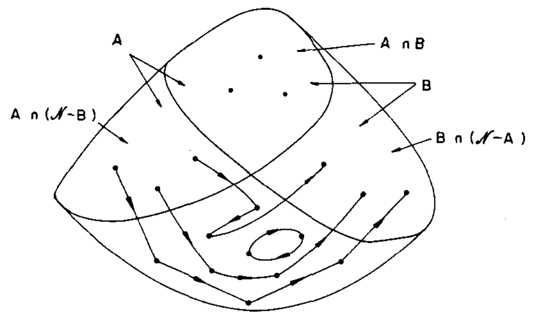
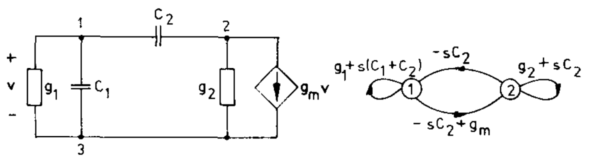


Fig. 3. Required multiconnections.



(a) (b)

Fig. 4. (a) Network. (b) Coates graph.

Y is a singular matrix and only its cofactors are used to evaluate network functions. From (5), (6) and (7) it can be seen that multiconnections used to evaluate cofactors of Y contain s as an isolated node. We can treat this node as a reference and delete all edges incident to it. From now on the Coates graph of a network will be assumed with the reference node deleted.

Example 1

An active linear network and its Coates graph with node 3 deleted are shown in Fig. 4. The indefinite admittance matrix is

$$Y=\begin{bmatrix} g_1+s(C_1+C_2) & -sC_2 & -g_1-sC_1 \\ -sC_2+g_m & g_2+sC_2 & -g_2-g_m \\ -g1-sC_1-g_m & -g_2 & g_1+g_2+sC_1+g_m \end{bmatrix}. \tag{11}$$

It is easy to confirm that

$$\begin{aligned} Y_{uv} &= |P_{\{(s,s)\}}| \\ &= (sC_2+g_2)\,(g_1+s(C_1+C_2)) \\ &\quad + sC_2(g_m-sC_2) \end{aligned}$$

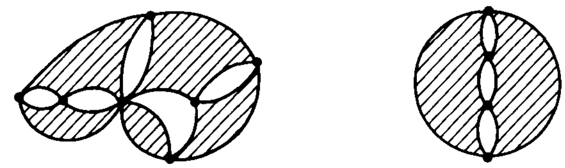
for any u,v and

$$Y_{12,33}=|P_{\{(1,2),(3,3)\}}|=-(g_m-sC_2).$$

Computer time needed for realization of direct topological formulas is proportional to the number of connections in a flowgraph. Let $\boldsymbol{D}(G)=[d_{ij}]$ be a matrix denoting the connection of a Coates flowgraph; d_{ij} is equal to the number of edges directed from the node i to the node j. $\boldsymbol{D}$ is a square matrix with the dimension equal to the number of nodes. The *number of connections* in a graph is equal [4] to

$$\text{card } P = \text{ per }[\boldsymbol{D}(g)] \tag{12}$$

where per$\boldsymbol{A}$ is a permanent of the matrix $\boldsymbol{A}$ [3].



(a) (b)

Fig. 5. (a) Node decomposition. (b) Four terminal bisection.

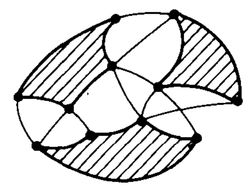


Fig. 6. Edge decomposition.

A very rough estimation for the number of connections for the graph with n nodes and k edges is given by [24]

$$\text{card } P \leqslant \left(\frac{k+1}{n}-1\right)^n. \tag{13}$$

Although (13) is only an upper estimate, it expresses correctly the rate of change in the number of terms. The exponential increase in the number of terms is observed in practice for direct topological analysis, which causes such analysis of large networks to be inexecutable.

III. The Graph Decomposition

The graph of an electrical network can be analyzed directly with the aid of (5)–(7), and the transfer function of the analyzed network can be obtained in all symbolic form. From the previous discussion, it is evident that the number of terms in the symbolic function is too large. As a result, the analysis of medium and large networks is a formidable task; network and graph decomposition becomes necessary.

The procedure of graph partition and determination of parts called *blocks* will be called decomposition.

A flowgraph can be decomposed in one of the three manners.

1) Node Decomposition: A graph is divided into edge disjoint subgraphs (blocks) (Fig. 5). Nodes common to two or more blocks are called *block nodes.* A particular case of node decomposition is bisection or decomposition into two subgraphs.

2) Edge decomposition: In a graph we isolate node disjoint blocks. Blocks are connected together by the means of edges which form cutsets of the graph (Fig. 6). These edges are called *cutting edges.* In the case of edge decomposition, nodes incident with cutting edges are called block nodes.

3) Hybrid Decomposition: This partition is a combination of two previous decompositions (Fig. 7). Nodes incident with cutting edges or common for more than one block are block nodes.

In both edge and hybrid decompositions, a bisection can be distinguished as a special case. We focus our attention

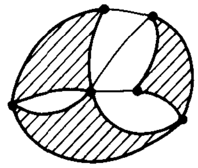


Fig. 7. Hybrid decomposition.

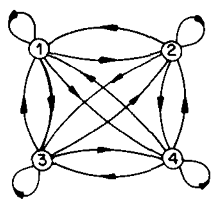


Fig. 8. Substitute graph spanned on four block nodes.

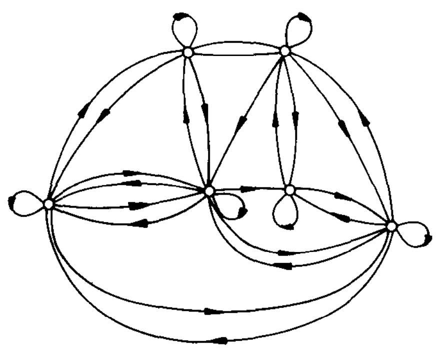


Fig. 9. Decomposition substitute graph for the decomposition from Fig. 7.

on bisection because of its special usefulness for the hierarchical decomposition. It is evident that any decomposition can be represented as a sequence of bisections, and for computer algorithms such an assumption produces simple data structures and simplifies organization of computations.

Definition 3

A complete symmetrical directed graph with self loops spanned on block nodes of subgraph G_i is called a *substitute graph* for that block and is denoted G_i^s (Fig. 8).

Definition 4

Graph G^d obtained when replacing blocks G_i by their substitute graphs is called a *decomposition substitute graph* (Fig. 9).

In the case of edge or hybrid decompositions, cutting edges belong to the decomposition substitute graph.

A decomposition substitute graph should not be too complex because the complexity of its analysis depends on the number of edges and nodes exactly as estimated for the case of proper graph (12), (13). Hence, it appears necessary to limit the number of blocks and block nodes. This limitation results in the *simple decomposition* method being ineffective for the case of very large networks. For large networks either the decomposition substitute graph G^d is too complex for analysis or blocks G_i are still too complex for direct topological analysis.

When simple decomposition is applied to subgraphs, we deal with *hierarchical decomposition*.

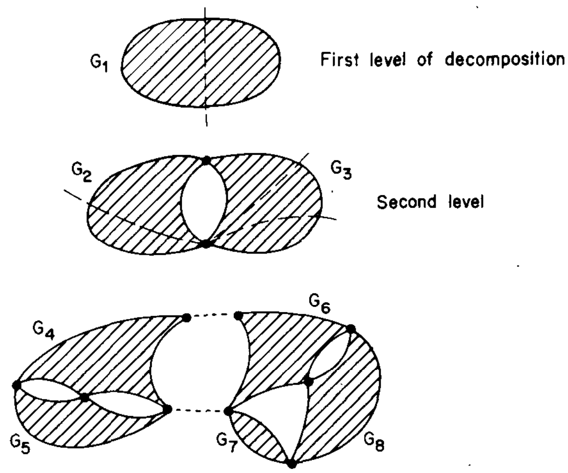


Fig. 10. Two level hierarchical decomposition.

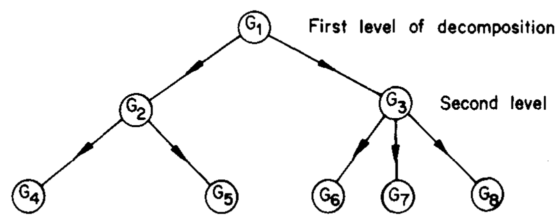


Fig. 11. Tree of decomposition shown in Fig. 10.

Decomposition of a network graph should be executed automatically. There are two reasons for this. First, the graph structure is not known when network data are provided and an *a priori* decision about block partition regarding only network structure could be nonoptimal. Second, elaboration of data would be cumbersome for the program user and would demand the knowledge of decomposition methods and calculations regardless of whether the partition is desirable or not.

The problem of graph decomposition is of the nonpolynomially bounded class. This means that time τ to find an optimal decomposition cannot be limited by a polynomial expressed in terms of nodes (n) or edges (k) number.

Taking the above into account we should not expect an efficient algorithm giving optimal solutions. Useful algorithms will provide a correct and nearly optimal solution in a short time. One such efficient algorithm has been presented in [21]. A modification of this algorithm gives the time of graph decomposition bounded linearly by the number of nodes.

IV. Hierarchical Analysis

Let us concentrate first on the case of node hierarchical decomposition. In Fig. 10 an example of hierarchical decomposition is presented. The hierarchical decomposition structure can be illustrated by a tree of decomposition. Nodes of the tree correspond to subgraphs obtained on different levels of decomposition. If a subgraph G_k was obtained during decomposition of subgraph G_i, then there is an edge from node G_i to node G_k. Fig. 11 shows the tree of decomposition from Fig. 10.

In the decomposition tree we have one *initial node* which is the root of the tree. *Terminal nodes* are leaves of the tree. All nodes that are not terminal nodes are *middle nodes*. For middle nodes we determine the *decomposition level* which is equal to the number of nodes in the path from the initial node to that node. *Range of hierarchical decomposition* is equal to the maximal decomposition level.

Every middle node has its *descendants* and every node except the initial one has its *ascendant*. If we limit ourselves to the bisection as the only graph partition, every middle node has exactly two descendants. As remarked previously, every decomposition can be considered as a sequence of bisections in hierarchical structure. Hence, without loss of generality, we shall examine this case only, obtaining a simpler expression of formulas and easier algorithm organization.

During the course of hierarchical decomposition analysis the following tasks are to be performed:

a) direct topological analysis of terminal blocks, and

b) analysis of middle blocks used to combine results from the higher level.

Analysis of Terminal Block

Let us consider a connection of a Coates graph. When we deal with a decomposed graph we can see that the part of the connection contained in a particular terminal block forms a multiconnection in this block.

The incidence of the block nodes determines the type of multiconnection. It means that topological analysis of terminal blocks will consist of enumeration of multiconnections, with paths linking different block nodes. Analysis of middle blocks will consist of combining together various types of multiconnections.

It is evident that combining multiconnections one by one will not reduce the computation time considerably. Multiconnections should be generated in groups and whole groups should be combined together. The larger the groups of multiconnections are the simpler the terminal block analysis is, and the more efficient middle block analysis is. One rule should be obeyed, namely, the resulting multiconnections should be generated without duplications.

The most detailed characterization is that presented in Definition 1, which is the generalization of Coates definition of 0- and 1-connections. For a block the different multiconnections may be grouped in sets P_W, of multiconnections characterized by the same set of nodes W.

However, it should be noted that a block with nb block nodes has

$$M(nb)=\sum_{i=0}^{nb}\binom{nb}{i}^2 i! \tag{14}$$

different types of multiconnection sets. This dependence could seriously limit the decomposition method. This led us to investigate other characterizations of multiconnection sets. After some trials [12], [13] the following type of characterization was chosen.

Definition 5

$P(B,E)$ is a set of multiconnections which have the following properties:

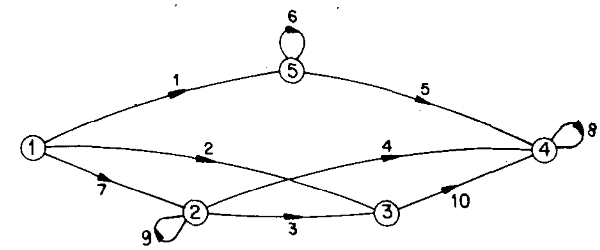


Fig. 12. Terminal block.

a) the incidence of block nodes is defined by sets B and E only—where B represents initial nodes and E represents terminal nodes of multiconnection edges;

b) all other nodes (internal nodes of a block) have full incidences, i.e., they are initial nodes as well as terminal nodes of multiconnection edges, where $B=\{b_1,b_2,\cdots,b_m\}$, $E=\{e_1,e_2,\cdots,e_m\}$ and $B\cup E\subset NB$—the set of block nodes.

Remark 1

Block nodes which are not included in $B\cup E$ are isolated nodes. Block nodes which are included in $B\cap E$ have full incidence.

Remark 2

In the sense of Definition 5 the set $P(B,E)$ contains k-connections with $k=\text{card}\,(NB-B\cap E)$.

Example 2

Let us consider the block shown in Fig. 12. If $NB=\{1,2,3,4\}$, $B=\{1,2\}$, $E=\{3,4\}$, then the set $P(B,E)$ is equal to $\{\{1,5,3\},\{2,4,6\}\}$. According to the Definition 1 each set of these multiconnections has different characterizations by sets W

$$\{1,5,3\}\in P_{\{(1,4),(2,3)\}}$$
$$\{2,4,6\}\in P_{\{(1,3),(2,4)\}}.$$

For another pair $B=\{1,2\}$, $E=\{2,3\}$ with the common node 2, the set of multiconnections $P(B,E)$ is equal to $\{\{2,6,9\},\{3,6,7\}\}$. In this case node 4 is isolated. The equivalent characterization by sets W, according to Definition 1, is as follows:

$$\{\{2,6,9\},\{3,6,7\}\}=P\{(1,3),(4,4)\}.$$

The weight function of multiconnection set $P(B,E)$ is defined as in (2). Note that for a block with nb block nodes, the number of different types of multiconnections sets $P(B,E)$ is

$$MR(nb)=\sum_{i=0}^{nb}\binom{nb}{i}^2 \tag{15}$$

which means an important reduction in comparison with (14). It will be shown that multiconnections characterized by sets $P(B,E)$ can be generated without duplication.

Analysis of Middle Block

Analysis on an intermediate level consists of evaluation of multiconnections of a block that result from the association of two (or in general, more) blocks. Let us denote the sets of block nodes for both blocks and the resulting block by NB_1, NB_2 and NB, respectively. When connecting two blocks, some of their block nodes become internal nodes, which means that no other blocks are connected to these nodes on upper levels. These nodes will be 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 (16)

$P_1(B_1, E_1)$, $P_2(B_2, E_2)$, and $P(B, E)$—sets of multiconnections (as defined in Definition 5) for both blocks and resulting block, respectively.

An important result is presented in the following theorem.

Theorem 2

Any set of multiconnections $P(B, E)$ can be obtained according to the following rule:

$$P(B,E) = U\, P_1(B_1, E_1) \times P_2(B_2, E_2) \qquad (17)$$

where summation is performed over all sets of multiconnections $P_1(B_1, E_1)$ and $P_2(B_2, E_2)$ satisfying conditions

$$B_1 \cap B_2 = \varnothing, \quad E_1 \cap E_2 = \varnothing$$
$$RED = (B_1 \cup B_2) \cap (E_1 \cup E_2) \qquad (18)$$

and $\times$ is a Cartesian product [3] of sets $P_1(B_1, E_1)$ and $P_2(B_2, E_2)$. Sets B and E are in this case equal to

$$B = B_1 \cup B_2 - RED, \quad E = E_1 \cup E_2 - RED.$$

If all element weights are different, there are no duplicate terms in the formula (17). For every multiconnection $p \in P$, the sign of p can be calculated as follows:

$$\text{sign}\, p = \text{sign}\, p_1 \cdot \text{sign}\, p_2 \cdot (-1)^k \cdot \Delta \qquad (19)$$

where

$$p = p_1 \cup p_2, \quad p_1 \in P_1, \quad p_2 \in P_2$$
$$k = \min(\text{card}(E_1 \cap B_2 \cap COM), \text{card}(E_2 \cap B_1 \cap COM)) + \text{card}(COM)$$
$$\Delta = \text{ord}(b_{11}, b_{22}, \cdots, b_{1m_1})\, \text{ord}(e_{11}, e_{12}, \cdots, e_{1m_1})\, \text{ord}(b_{21}, b_{22}, \cdots, b_{2m_2})\, \text{ord}(e_{21}, e_{22}, \cdots, e_{2m_2})$$
$$B_1 = \{b_{11}, b_{12}, \cdots, b_{1m_1}\}, \quad E_1 = \{e_{11}, e_{12}, \cdots, e_{1m_1}\}$$
$$B_2 = \{b_{21}, b_{22}, \cdots, b_{2m_2}\}, \quad E_2 = \{e_{21}, e_{22}, \cdots, e_{2m_2}\}.$$

Proof of Theorem 2 is based on the observation that each element of $P_1(B_1, E_1) \times P_2(B_2, E_2)$ is a multiconnection of the type $P(B, E)$, and similarly, for each element $p \in P(B, E)$ there is a unique pair of elements $p_1 \in P_1(B_1, E_1)$ and $p_2 \in P_2(B_2, E_2)$ such that $p = p_1 p_2$. Therefore, multiconnection sets on both sides of (17) are equal. Since $P_1(B_1, E_1)$ and $P_2(B_2, E_2)$ are defined on edge disjoint subgraphs there will be no duplicate terms in (17). The sign of multiconnections is important in realization of formulas for transfer functions and (19) is to update the sign according to the topology of the association of two blocks.

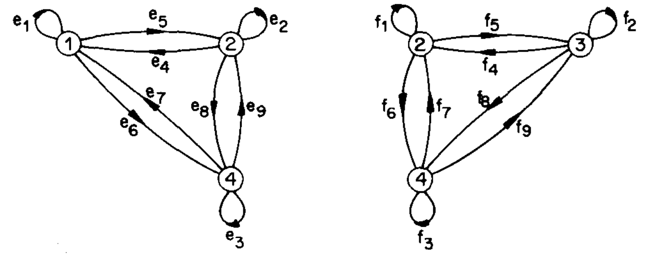


Fig. 13. Blocks to be connected.

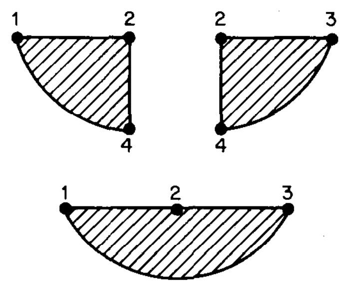


Fig. 14. Association of two blocks.

Remark

An important feature of (17) is the possibility of obtaining a set of multiconnections $P(B, E)$ by combining whole groups of multiconnections from the lower level. At the same time, from (19) we notice that the new sign is attributed simultaneously to the whole group of terms $P_1 \times P_2$, as k and Δ depend only on sets B_1, E_1, B_2, E_2. These features are of great importance in the computer realization because we do not have to deal with each multiconnection separately.

Example 3

Consider an association of two blocks presented in Figs. 13 and 14. We have $NB_1 = \{1,2,4\}$, $NB_2 = \{2,3,4\}$. $NB = \{1,2,3\}$, $COM = \{2,4\}$, $RED = \{4\}$. Let us calculate multiconnections of the type $P(\{1,2\},\{2,3\})$ of the resulting block. From the formula (17), with the condition (18), we obtain

$$\begin{aligned} P(\{1,2\},\{2,3\}) = {} & P_1(\{1,4\},\{2,4\}) \times P_2(\{2\},\{3\}) \\ & \cup P_1(\{1\},\{2\}) \times P_2(\{2,4\},\{3,4\}) \\ & \cup P_1(\{1,2\},\{2,4\}) \times P_2(\{4\},\{3\}) \\ & \cup P_1(\{1\},\{4\}) \times P_2(\{2,4\},\{2,3\}). \end{aligned}$$

A formula similar to that of Theorem 2 can be derived for the case of edge decomposition. Analysis of terminal blocks is realized in the same way as described previously. An edge bisection will be considered. We denote:

E_{cut}	a cutset of a graph G;
$G_1(E_1, V_1), G_2(E_2, V_2)$	two disconnected graphs obtained from G after removing edges E_{cut};
NB, NB_1, NB_2	sets of block vertices for G, G_1 and G_2, respectively,
$RED = (NB_1 \cup NB_2 - NB)$	the set of reducible nodes,

P_{cut} the set of multiconnections formed by edges E_{cut} only.

Theorem 3 [12]

Any set of multiconnections $P(B, E)$ can be obtained according to the following rule[1]

$$P(B,E) = \cup P_1(B_1, E_1) \times P_2(B_2, E_2) \times P_{cut}(B_c, E_c) \tag{20}$$

where summation is performed over all sets of multiconnections P_{cut}, with sets B_1, E_1, B_2, E_2 satisfying the following conditions:

$$B_c \cap B_1 = B_c \cap B_2 = E_c \cap E_1 = E_c \cap E_2 = \varnothing$$
$$RED \subset (B_c \cup B_1 \cup B_2) \cap (E_c \cup E_1 \cup E_2).$$

Sets B and E are then equal to

$$B = B_1 \cup B_2 \cup B_c - RED$$
$$E = E_1 \cup E_2 \cup E_c - RED.$$

If all element weights are different, there are no duplicate terms in the formula (20). For every multiconnection $p \in P$, the sign of p can be calculated as follows:

$$\text{sign } p = \text{sign } p_1 \cdot \text{sign } p_2 \cdot \text{sign } p_{cut}. \tag{21}$$

Downward and Upward Hierarchical Analysis

Now the method of analysis of terminal blocks and middle blocks is completed. The remaining step is the exploration of hierarchical structure to obtain a description of the initial network.

Two approaches are possible and are called the upward and downward methods of analysis. The upward method presents many advantages over the downward method, including savings of computer time and memory, so the latter will be only briefly outlined.

In the downward method, the analysis starts at the 1-level (initial block) and proceeds down to the next levels according to the connections in the hierarchical tree. The substitute graphs of blocks corresponding to the middle nodes are analyzed. On each intermediate level the type of necessary functions from the next level is determined. Arriving at the terminal node the analysis of the terminal block is executed to get the necessary function of this block. Then one proceeds upward. For each pass through the middle node, the multiplication of two functions from the lower level is executed. After arriving at the 1-level, we obtain a part of the function of the initial network. Many passes up and down the tree are necessary; much processing has to be performed. The formula (17) expresses a set of multiconnections of the middle block as a sum of products of multiconnection sets from lower level. Each term of this sum requires the above described up and down procedure.

The downward method presented in [12], permits the hierarchical analysis of large networks but has the following disadvantages:

a) multiple passes through the hierarchical structure causes multiple calculations of the same function;
b) complicated organization scheme;
c) problems with efficient storage of all-symbolic results.

For these reasons a new form of hierarchical tree exploration was elaborated. In the new method, only one pass along the hierarchical structure is necessary. The name *upward method* is due to the direction in which the decomposition tree is worked out—from the terminal nodes upward to the initial node.

Let us describe the upward hierarchical analysis in more detail. First, to facilitate the organization of the algorithm, a specific numeration of blocks is introduced. If N is the number of blocks (i.e., terminal and middle nodes), we shall number them from 1 to N in such way that each ascendant has lower number than its descendants. Such a numeration is easy to perform, e.g., we can number nodes starting from level 1 and move sequentially to the lowest level (as in Fig. 11). With this numeration the initial block has always number 1.

The upward method of analysis starts from the block having the number N and is performed sequentially to the number 1. When the terminal block is reached, the analysis presented in Section V is executed. When we arrive at the middle block, descendants of which have been previously analysed, the formula (17) is used. Two approaches to realize formula (17) are possible:

1) using the substitute graph, the combinations of sets B_1, E_1, B_2 and E_2 which satisfy conditions of the Theorem 2 are obtained directly, or
2) examining all possible combinations of multiconnections of descendant blocks, only those which satisfy conditions of the Theorem 2 are retained.

Since a simple test for combinations has been found (see Section V), the second approach was chosen for the algorithm and the program. The procedure ends after the initial block is analyzed. Then the functions of the original network are calculated.

V. Algorithm of Upward Hierarchical Analysis

As can be noted from the general presentation of the method, there are two distinct stages in the upward hierarchical analysis: analysis of terminal blocks and analysis of middle blocks. These two stages are resolved separately and each one can be ameliorated without affecting the other.

Analysis of the Terminal Block

An algorithm to generate multiconnections of the Coates graph will be presented. This part of the method corresponds to the methods of direct topological analysis of electrical circuits. Generation of 0-connections of a flowgraph can be converted to the problem of generation of disjoint cycles of a graph (see [6]).

[1] This form of the formula (20), a modification of the formula presented in [12], has been proposed by M. Bon.

Let us consider a Coates graph with n nodes. Let M be an incidence matrix defined as follows: $M=[m_{ij}]_{n\times n}; m_{ij}$ = the set of edges starting from the ith node and ending at the jth node. The set of 0-connections of a flow-graph can be calculated from the formula

$$P = \bigcup_{(i_1,\cdots,i_n)\in I} m_{1i_1}\times m_{2i_2}\times\cdots\times m_{ni_n} \qquad (22)$$

where I is the set of all permutations of numbers $(1,2,\cdots,n)$. There is no duplication in the formula (22). The sign of 0-connection $p\in m_{1i_1}\times m_{2i_2}\times\cdots m_{ni_n}$ is equal to $(-1)^{n+h}$, where h is a number of permutations necessary to order the set $i_1,\cdots,i_n$.

In the formulas for the hierarchical analysis, not only the set of all 0-connections is necessary but also sets of multiconnections characterized in Definition 5. This problem can be transformed to the generation of all 0-connections of the modified graph as follows.

Lemma 1:

The set of multiconnections $P(B,E)$ of a graph with an incidence matrix M is equal to the set of 0-connections of a graph described by a 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 4

To generate the set of multiconnections $P(\{1,2\},\{3,4\})$ of the graph discussed in the Example 2, let us reduce the incidence matrix M, where

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

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

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

Applying the formula (22) to $M(B,E)$ we obtain the sets of multiconnections $P(B,E)$ so that

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

as expected.

The complete description of the block with nb block nodes is given by weight functions of all possible sets $P(B,E)$ with $B\cup E\subset NB$. Different sets B,E can be generated in the manner described below.

Let us numerate nodes NB from 1 to nb. For $i=0,\cdots,nb$, all i-element subsets of the set $\{1,\cdots,nb\}$ are generated. For a given i, let $\binom{nb}{i}$ such subsets form the set $K(i)$. Each pair of sets (m,k), where $m,k\in K(i)$ (note that m may be equal to k) has a corresponding set of potential multiconnections $P(B,E)$ with $B=m$ and $E=k$. Such sets of multiconnections are generated and stored. Each set may be identified by its type B,E. This type may be coded on a single computer word. The $2*nb$ bits would be occupied. Successive pairs of bits describe block nodes from 1 to nb. All elements b from B produce 1 on the position $2*b-1$ and elements e from E produce 1 on the position $2*e$. All other positions are equal to 0. The *identification code* C of a set of multiconnections $P(B,E)$ can be completely calculated from the formula

$$C=\sum_{b\in B}2**(2b-2)+\sum_{e\in E}2**(2e-1). \qquad (23)$$

Example 5

For the set of block nodes $NB=\{1,2,3,4\}$, 8 bits are occupied to code different sets of multiconnections. If $B=\{1,2\}$ and $E=\{2,3\}$ the code for $P(B,E)$ is equal to

$$C=2^0+2^2+2^3+2^5=45.$$

This coding permits an easy identification of a multiconnection set by one interger number and a simple practical realization formula (17).

Analysis of the Middle Block

In the upward hierarchical method, the analysis of a middle block is performed at the moment when both its descendants have already been analyzed. The sets of multiconnections of these blocks are stored in the computer memory each having its identification code. The following rules of block nodes numeration are to be observed (renumerate if necessary):

first group is formed of reducible nodes *RED*;
second group is formed of other common nodes *COMRED*;
third group is formed of other block nodes.

Both the first and second group should have the same numeration in blocks to be associated. We examine all possible combinations of functions describing two blocks. Let us denote the following bit fields in a computer word containing the code of a multiconnection:

RED_1, RED_2	corresponding to the nodes *RED* in both blocks (first group);
CR_1, CR_2	corresponding to the second group of nodes;
$REST_1, REST_2$	corresponding to the third group of nodes.

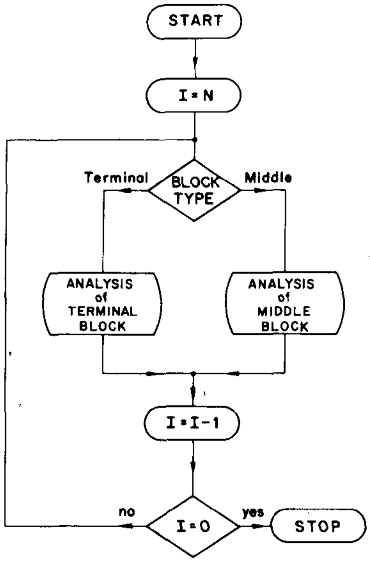
The following tests are performed

$$\text{AND}(RED_1,RED_2)=0$$
$$\text{AND}(CR_1,CR_2)=0$$
$$\text{OR}(RED_1,RED_2)=\text{ field having 1 on each bit.} \qquad (24)$$

For the chosen code of multiconnection (23), conditions (24) are equivalent to (18). So if any of these conditions are not fulfilled, the combination is rejected. In the contrary case, we have the combination characterized by sets of nodes satisfying the formulas (18).

The code for resulting multiconnections can easily be composed from the parts of codes of component multiconnections. Since none of the first group of nodes remains a block node, there is no information concerning this group.

TABLE I
ORGANIZATION OF ALGORITHM OF HIERARCHICAL ANALYSIS



Nodes from the second group have code equal $\text{OR}(CR_1, CR_2) = CR_1 + CR_2$. As nodes from the third group are distinct in two blocks, their description remains $REST_1, REST_2$.

General Organization of the Algorithm

The general organization of the algorithm is presented in Table I. Once the proper numeration of block nodes is established, the analysis can be carried out as presented. With this numeration, analysis of any middle block is performed when both its descendants have been analyzed. The last analyzed block is the initial block.

The all-symbolic or semi-symbolic descriptions for large networks are intermediate results only. These results are used later in various types of network analysis.

The symbolic form of the transfer function for a large network contains a very large number of terms. To make possible the storage and to facilitate further work the decomposed form of results is preserved.

A terminal block is described by the weight functions of its multiconnection sets. Each term of a weight function has the form

$$t = r \cdot s^k \prod_i y_i \tag{25}$$

where $r =$ numerical factor; $s =$ Laplace variable; $y_i =$ symbolic admittances or symbolic element parameters. Any weight function is stored in the form of three vectors with successive elements equal: r, k and coded y_i. Each function can be recognized by its identification code (23).

From formula (17) we see that any function for a middle block is expressed as a sum of products of functions from the lower level. In the upward hierarchical method, the analysis of any middle block is performed after its descendants have been previously analyzed and resulting functions stored. The function of a middle block can be stored in an unexpanded form containing only addresses or functions to be multiplied. A term of such function is of the form

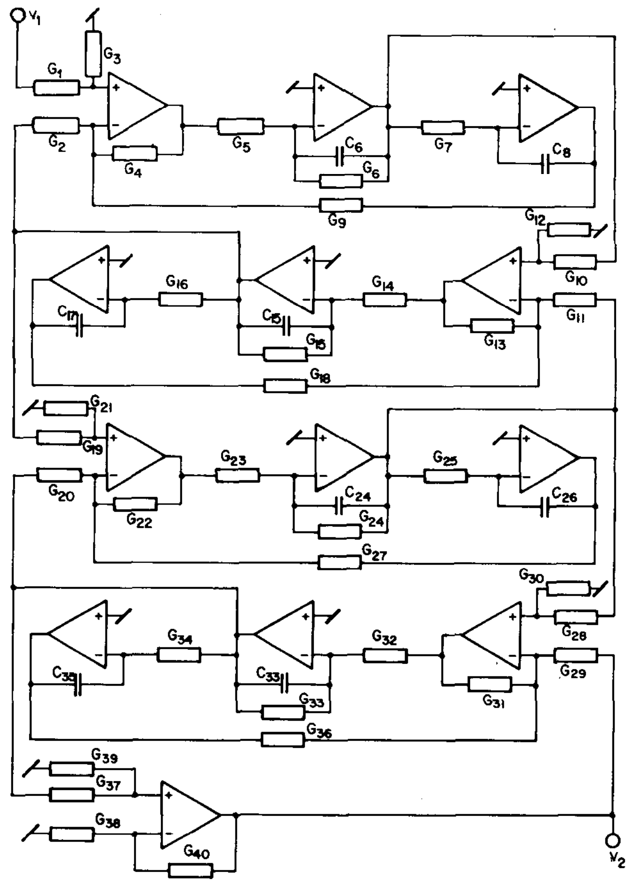


Fig. 15. Band-pass filter.

$$m = v \cdot F(i) \cdot F(k) \tag{26}$$

where $v =$ sign of term equal to $\pm$ 1, and $F(i), F(k) =$ functions describing descendants of the analyzed block.

The term m can be represented by three numbers: v and addresses of $F(i)$ and $F(k)$ stored previously.

The analysis is terminated by analyzing the initial block. Therefore, the whole hierarchical structure should be run through. From the functions of the initial block we only choose the necessary ones. The given addresses send us to next blocks. At the end we find functions of the terminal block. On these functions different kinds of operations can be performed, depending on what kind of analysis is required.

Example 6

Let us take a practical network to illustrate the algorithm. In Fig. 15 the scheme of an analyzed band-pass filter is shown. Operational amplifiers are considered ideal. The Coates flowgraph corresponding to this network is shown in Fig. 16. This graph has been decomposed into 5 terminal blocks (Fig. 17). The hierarchical structure of successive associations is presented in Fig. 18.

To illustrate the analysis of the terminal blocks let us consider the terminal block 8. Its flowgraph is indicated in Fig. 16 by the dashed line. In this block the only nonempty

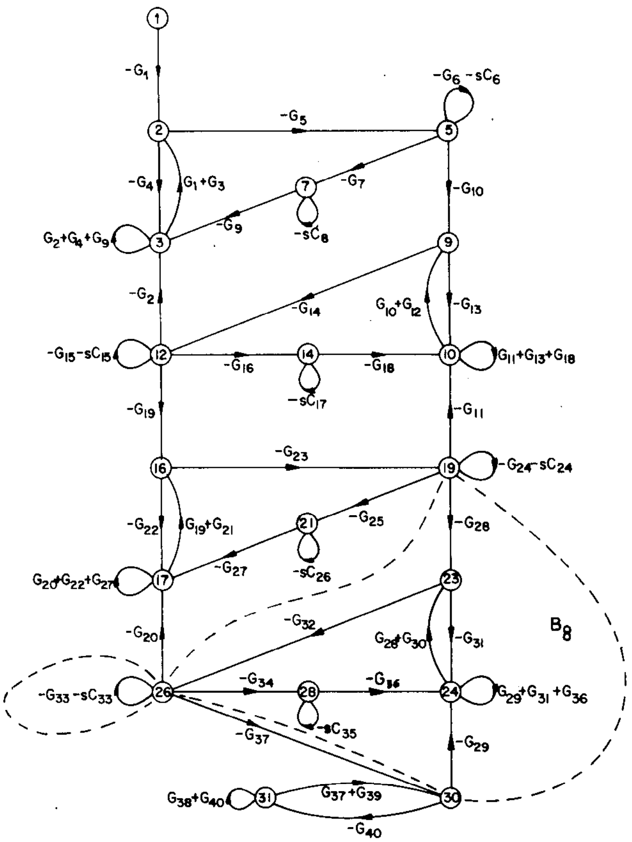


Fig. 16. Flowgraph for bandpass filter.

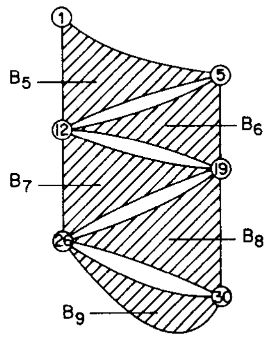


Fig. 17. Block graph.

types of sets of multiconnections $P_8(B, E)$ are

1) For $B=\{30\}$, $E=\{26\}$

$$|P_8(B,E)| = F(3) = G_{29}G_{32}sC_{35}(G_{28}+G_{30}).$$

2) For $B=\{26\}$, $E=\{26\}$

$$|P_8(B,E)| = F(4) = (G_{28}+G_{30}) \cdot [G_{32}G_{34}G_{36}+G_{31}sC_{35}(G_{33}+sC_{33})].$$

3) For $B=\{19\}$, $E=\{26\}$

$$|P_8(B,E)| = F(5) = G_{28}G_{32}sC_{35}(G_{29}+G_{31}+G_{36}).$$

4) For $B=\{0\}$, $E=\{0\}$

$$|P_8(B,E)| = sC_{35}G_{31}(G_{28}+G_{30}).$$

In Table II the first three types only are shown as they are required for voltage transfer function evaluation.

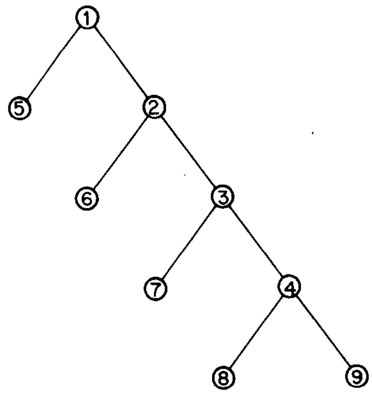


Fig. 18. Tree of the hierarchical structure.

TABLE II
RESULTS OF SYMBOLIC ANALYSIS

Block number	NB	B	E	Weight Function	
9	26,30	26	30	F(1)	$G_{37}(G_{38}+G_{40})$
		30	30	F(2)	$G_{40}(G_{37}+G_{39})$
8	19,26,30	30	26	F(3)	$G_{29}G_{32}sC_{35}(G_{28}+G_{30})$
		26	26	F(4)	$(G_{28}+G_{30})[G_{32}G_{34}G_{36}+G_{31}sC_{35}(G_{33}+sC_{33})]$
		19	26	F(5)	$G_{28}G_{32}sC_{35}(G_{29}+G_{31}+G_{36})$
7	12,19,26	26	19	F(6)	$G_{20}G_{23}sC_{26}(G_{19}+G_{21})$
		19	19	F(7)	$(G_{19}+G_{21})[G_{23}G_{25}G_{27}+G_{22}sC_{26}(G_{24}+sC_{24})]$
		12	19	F(8)	$G_{19}G_{23}sC_{26}(G_{20}+G_{22}+G_{27})$
6	5,12,19	19	12	F(9)	$G_{11}G_{14}sC_{17}(G_{10}+G_{12})$
		12	12	F(10)	$(G_{10}+G_{12})[G_{14}G_{16}G_{18}+G_{13}sC_{17}(G_{15}+sC_{15})]$
		5	12	F(11)	$G_{10}G_{14}sC_{17}(G_{11}+G_{13}+G_{18})$
5	1,5,12	12	5	F(12)	$G_2G_5sC_8(G_1+G_3)$
		5	5	F(13)	$(G_1+G_3)[G_5G_7G_9+G_4sC_8(G_6+sC_6)]$
		1	5	F(14)	$G_1G_5sC_8(G_2+G_4+G_9)$
4	19,26,30	26,30	30,30	F(15)	- F(3)F(1) + F(4)F(2)
		19,26	26,30	F(16)	- F(5)F(1)
		19,30	26,30	F(17)	F(5)F(2)
3	12,19,30	12,19	19,30	F(18)	F(8)F(16)
		19,30	19,30	F(19)	F(7)F(15) - F(6)F(17)
		12,30	19,30	F(20)	- F(8)F(15)
2	5,12,30	5,12	12,30	F(21)	F(11)F(18)
		12,30	12,30	F(22)	F(10)F(19) - F(9)F(20)
		5,30	12,30	F(23)	- F(11)F(19)
1	1,30	1	30	F(24)	F(14)F(21)
		30	30	F(25)	F(13)F(22) - F(12)F(23)

Association of blocks is performed according to the Theorem 2. For example the middle block 2 has $NB = \{5,12,30\}$ and is obtained as the association of block 6 with $NB_6 = \{5,12,19\}$ and block 3 with $NB_3 = \{12,19,30\}$ (see Fig. 18). For this association we have $COM = NB_6 \cap NB_3 = \{12,19\}$, $RED = COM - NB = \{19\}$. Considering only multiconnections necessary to obtain te required transfer function we evaluate:

$$\begin{aligned} F(21) &= |P_2(\{5,12\},\{12,30\})| \\ &= |P_6(\{5\},\{12\})|\cdot|P_3(\{12,19\}\{19,30\})| \\ &= F(11)F(18), \\ F(22) &= |P_2(\{12,30\},\{12,30\})| \\ &= |P_6(\{12\},\{12\})|\cdot|P_3(\{19,30\},\{19,30\})| \\ &\quad - |P_6(\{19\},\{12\})|\cdot|P_3(\{12,30\},\{19,30\})| \\ &= F(10)F(19) - F(9)F(20), \\ F(23) &= |P_2(\{5,30\},\{12,30\})| \\ &= -|P_6(\{5\},\{12\})|\cdot|P_3(\{19,30\},\{19,30\})| \\ &= -F(11)F(19). \end{aligned}$$

Symbolic results for the total network are shown in Table II. The results are presented in the unexpanded form as they are computed by the program. The voltage transfer function for the considered filter can be expressed as

$$K_v = \frac{v_2}{v_1} = \frac{|P_1(\{1\},\{30\})|}{|P_1(\{30\},\{30\})|} = \frac{F(24)}{F(25)}. \tag{27}$$

The obtained formula represents the symbolic transfer function, which can be used in compact form or expanded if necessary. This network has 44 elements and consequently 44 symbolic parameters in the symbolic results. Fully symbolic analysis of networks of this size can require considerable computer time when direct topological methods are applied. In the case of hierarchical analysis, it is even possible to obtain these results by hand calculations.

Notice that for this structure, the graphs of blocks 8, 7, 6, and 5 are isomorphic. If an isomorphism of graphs is detected, it is possible to execute block analysis only once since the symbolic descriptions of isomorphic blocks are identical. This permits reduction of computer time as well as the memory needed to store the results.

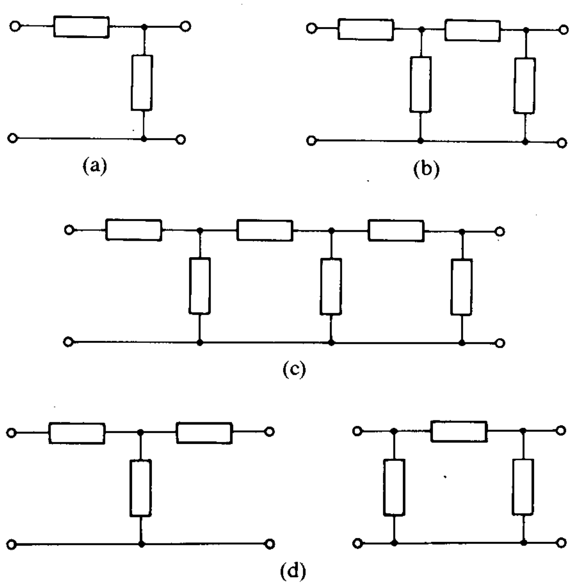


Fig. 19. Terminal blocks of ladder decomposition.

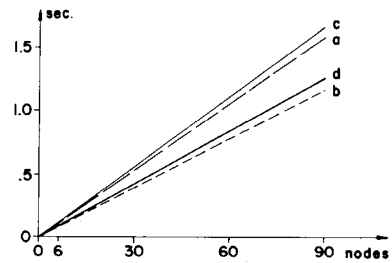


Fig. 20. Relationship between the analysis time and the number of nodes.

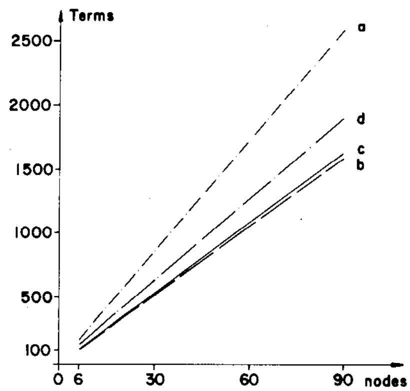


Fig. 21. Relationship between the number of terms and the number of nodes.

VI. Computer Realization and Results

Two computer programs were developed on the basis of presented algorithms. Programs FANES [12] realizes the downward analysis of hierarchical structure. The edge decomposition is used in this program. Some comparisons between SNAPEST, NAPPE, SNAP [14], [26] and FANES are presented in [12].

First results of the program FLOWUP realizing the upward hierarchical method were published in [13]. Program FLOWUP is written in Fortran and is implemented on CDC Cyber 73 and CIIHB DPS/8 computers. Memory demands for the program are not important and additionally two parts of the program, namely terminal and middle block analysis, can be separated and overlayed. The BASIC version for the minicomputer HP9835 (or HP9845) with standard memory has been realized.

Input data contains a node-to-node description of network elements. The program generates a signal flow-graph using element graphs obtained on the basis of modified admittance matrix (see Appendix). Then the hierarchical decomposition is automatically carried out and a structure of the decomposition tree is established by the program. Next the hierarchical analysis of the entire structure is

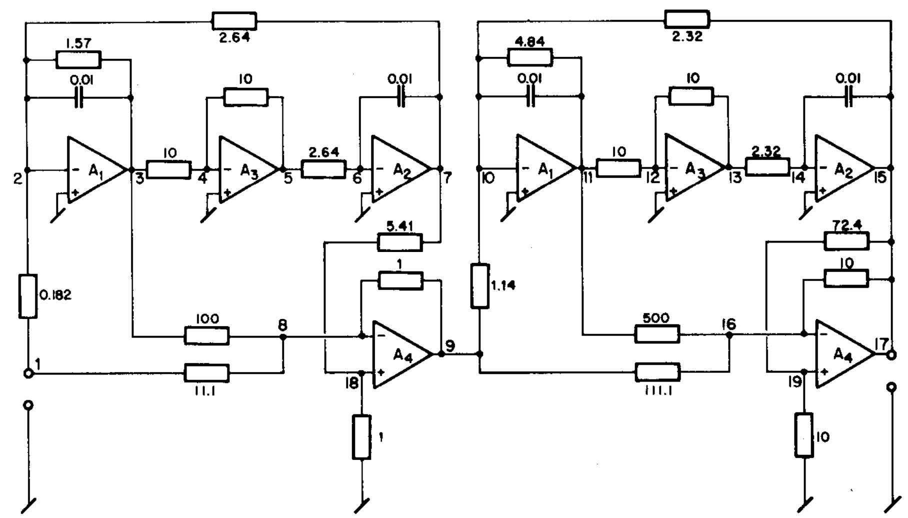


Fig. 22. Low-pass filter network.

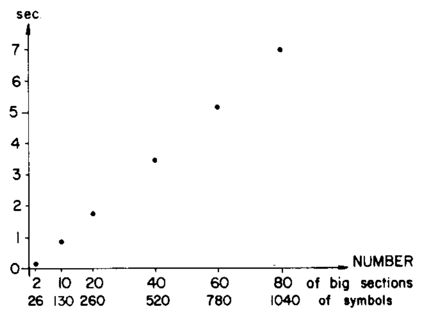


Fig. 23. Relationship between the analysis time and the size of the network.

performed starting from the graph of the highest number as shown in Table I. For the terminal block analysis, matrices of the range $n \times n$ are to be stored, where n-number of block nodes (in general not greater than 10). The demand for middle block analysis is due to the number of block nodes. In the case of analysis of large networks, the most important memory demand is due to the storage of symbolic results. Three vectors (25), each with length equal to the number of terms, are necessary. When the compact form is used the all-symbolic form for quite large networks can be calculated. In the minicomputer version the successive transfer of results to other memory supports may be performed during the program execution.

Let us present now some comparative results of analysis with the FLOWUP program. First let us examine the ladder structure decomposed into different terminal blocks, as shown in Fig. 19. Time of computer analysis and number of terms in the results are presented in Figs. 20 and 21, where the lines a, b, c, d represent the terminal blocks having the structure, as shown in Fig. 19(a), (b), (c), and (d), respectively. The isomorphism of the terminal blocks was not exploited. Both time and memory depend linearly on the number of nodes of the analyzed ladder. Linear dependence is typical for all cascade connections of blocks. Note that both time and memory depend on the kind of partition performed. These computations have been done on a CDC Cyber 73.

Analysis of the filter presented in Fig. 22 was executed on a DTS/8 GCOS. Analysis time for this filter was 0.165 s. In the case of cascade connection of many such sections, we have the linear growth of computer time as presented in Fig. 23. The isomorphism of sections has not been taken into account. When connection of blocks is more complicated than cascade, the analysis is expected to be more time consuming.

VII. Conclusion

We have discussed a new method that increases the computation power of topological analysis due to the reduction in the computer time needed for the analysis of large electronic networks. The approach will significantly affect the applications of topological methods to the analysis of large networks which was impossible even with the aid of the fastest computers.

Hence, network design problems requiring topological analysis can be solved with the help of the symbolic form of results. The method preserves the advantages of direct methods of topological analysis such as high accuracy of computations and possibility of generating fully symbolical results.

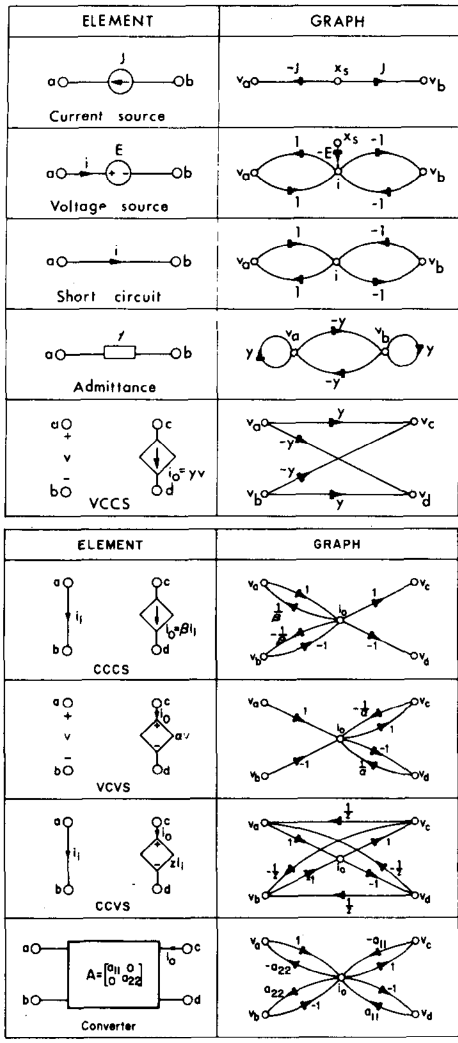
As the method is based on hierarchical decomposition, different blocks can be analyzed independently. Thus the use of parallel processing techniques is feasible and further reduction in computational time is possible.

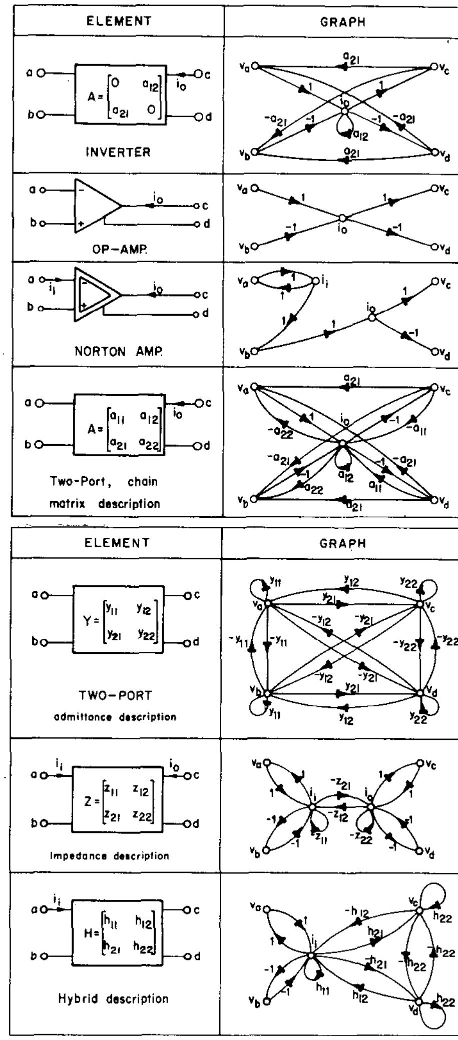
The restriction of the presented method in its application to large networks lies in the number of block nodes in each block. This is usually overcome by using an effective decomposition algorithm which minimizes the number of partition nodes.

Appendix

In Table III flowgraph models for chosen network elements are shown. They are based on modified admittance matrices of elements and they can be directly connected resulting in a flowgraph of an analyzed circuit. Adding a new element does not alter the structure of an existing

TABLE III
ELEMENT'S MODELS





ELEMENT	EQUATIONS
Current source	$i_a = -J$ $i_b = J$
Voltage source	$v_a - v_b - E = 0$ $i_a = i, \quad i_b = -i$
Short circuit	$v_a - v_b = 0$ $i_a = i, \quad i_b = -i$
Admittance	$i_a = y(v_a - v_b)$ $i_b = y(v_b - v_a)$
VCCS ($i_o = yv$)	$i_a = 0, \quad i_b = 0$ $i_c = y(v_a - v_b)$ $i_d = y(v_b - v_a)$
CCCS ($i_o = \beta i_i$)	$v_a - v_b = 0$ $i_a = \frac{1}{\beta} i_o, \quad i_b = -\frac{1}{\beta} i_o$ $i_c = i_o, \quad i_d = -i_o$
VCVS	$v_a - v_b - \frac{1}{a}(v_c - v_d) = 0$ $i_a = 0, \quad i_b = 0$ $i_c = i_o, \quad i_d = -i_o$
CCVS	$v_a - v_b = 0$ $i_a = \frac{1}{z}(v_c - v_d), \quad i_b = \frac{1}{z}(v_d - v_c)$ $i_c = i_o, \quad i_d = -i_o$
Converter, $A = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}$	$v_a - v_b - a_{11}(v_c - v_d) = 0$ $i_a = -a_{22} i_o, \quad i_b = a_{22} i_o$ $i_c = i_o, \quad i_d = -i_o$
INVERTER, $A = \begin{bmatrix} 0 & a_{12} \\ a_{21} & 0 \end{bmatrix}$	$v_a - v_b + a_{12} i_o = 0$ $i_a = a_{21}(v_c - v_d)$ $i_b = a_{21}(v_d - v_c)$ $i_c = i_o, \quad i_d = -i_o$
OP-AMP	$v_a - v_b = 0$ $i_a = 0, \quad i_b = 0$ $i_c = i_o, \quad i_d = -i_o$
NORTON AMP	$v_a = 0, \quad v_b = 0$ $i_a = i_i, \quad i_b = i_i$ $i_c = i_o, \quad i_d = -i_o$
Two-Port, chain matrix description, $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$	$v_a - v_b - a_{11}(v_c - v_d) + a_{12} i_o = 0$ $i_a = a_{21}(v_c - v_d) - a_{22} i_o$ $i_b = a_{21}(v_d - v_c) + a_{22} i_o$ $i_c = i_o, \quad i_d = -i_o$
TWO-PORT admittance description, $Y = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}$	$i_a = y_{11}(v_a - v_b) + y_{12}(v_c - v_d)$ $i_b = y_{11}(v_b - v_a) + y_{12}(v_d - v_c)$ $i_c = y_{21}(v_a - v_b) + y_{22}(v_c - v_d)$ $i_d = y_{21}(v_b - v_a) + y_{22}(v_d - v_c)$
Impedance description, $Z = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix}$	$v_a - v_b - z_{11} i_i - z_{12} i_o = 0$ $v_c - v_d - z_{21} i_i - z_{22} i_o = 0$ $i_a = i_i, \quad i_b = -i_i$ $i_c = i_o, \quad i_d = -i_o$
Hybrid description, $H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$	$v_a - v_b - h_{12}(v_c - v_d) - h_{11} i_i = 0$ $i_c = h_{22}(v_c - v_d) + h_{21} i_i$ $i_d = h_{22}(v_d - v_c) - h_{21} i_i$ $i_a = i_i, \quad i_b = -i_i$

flowgraph—only new edges will be added according to the new element's model. In equations describing the models in Table III symbol i_x $(x = a, b, c, d)$ indicates current directed away from the node x.

REFERENCES

[1] C. Acar, "New expansion for signal-flow graph determinant," *Electron. Lett.*, Dec. 1971.

[2] G. E. Alderson and P. M. Lin, "Computer generation of symbolic network functions—a new theory and implementation," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 48–56, Jan. 1973.

[3] W. K. Chen, *Applied Graph Theory—Graphs and Electrical Networks*. Amsterdam, The Netherlands: North-Holland, 1976.

[4] W. K. Chen, "Flow graphs: Some properties and methods of simplifications," *IEEE Trans. Circuit Theory*, vol. CT-12, pp. 128–130, 1965.

[5] W. K. Chen, "Unified theory on the generation of trees of a graph," *Int. J. Electron.*; "Part I. 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.

[6] L. O. Chua and P. M. Lin, *Computer Aided Analysis of Electronic Circuits—Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[7] C. L. Coates, "Flow graph solutions of linear algebraic equations," *IRE Trans. Circuit Theory*, vol. CT-6, pp. 170–187, 1959.

[8] C. L. Coates, "General topological formulas for linear networks functions," *IRE Trans. Circuit Theory*, vol. 5, pp. 42–54, Mar. 1958.

[9] J. Cajka, "New formula for the signal-flow graph reduction," *Electron. Lett.*, pp. 437–438, July 1970.

[10] N. Deo, *Graph Theory with Applications to Engineering and Computer Sciences*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[11] W. R. Dunn and S. P. Chan, "Analysis of active networks by a subgraph-construction technique," presented at *Second Asilomar Conf. Circuits Sys.* 1968.

[12] A. Konczykowska and J. Starzyk, "Computer analysis of large signal flowgraphs by hierarchical decomposition method," in *Proc. European Conf. Circuit Theory Design*, (Warsaw, Poland), 1980, pp. 408–413.

[13] A. Konczykowska and J. Starzyk, "Computer justification of upward topological analysis of signal-flow graphs," in *Proc. European Conf. Circuit Theory Design*, (The Hague), 1981, pp. 464–467.

[14] P. M. Lin and G. E. Alderson, "SNAP-A computer program for generating symbolic network functions," School Elec. Eng., Purdue Univ. Lafayette, Ind. Tech. Rep. TR-EE, 70–16, Aug. 1970.

[15] W. J. McCalla and P. O. Pederson, "Elements of computer-aided analysis," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 14–26, Jan. 1971.

[16] O. P. McNamee and N. Potash, "A user's and programmer's manual for NASAP," Univ. California, Los Angeles, Rep. 63-68, Aug. 1968.

[17] R. R. Mielke, "A new signal flow graph formulation of symbolic network functions," *IEEE Trans. Circuits Sys.*, vol. CAS-25, pp. 334–340, June 1978.

[18] U. Ozguner, "Signal flow graph analysis using only loops," *Electron. Lett.* pp. 359–360, Aug. 1973.

[19] D. E. Riegle and P. M. Lin, "Matrix signal flow graphs and an optimum topological method for evaluating their gains," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 427–435, 1972.

[20] S. M. Roberts and B. Flores, "Systematic generation of Hamiltonian circuits," *Comm. ACM*, vol. 9, pp. 690–694, 1966.

[21] A. Sangiovanni-Vincentelli, L. K. Chen and L. O. Chua, "An efficient heuristic cluster algorithm for tearing large-scale networks,"

IEEE Trans. Circuits Syst., vol. CAS-24. pp. 709–717, Dec. 1977.
[22] S. D. Shieu and S. P. Chan, "Topological formulas of symbolic network functions for linear active networks," in *Proc. Int. Symp. Circuit Theory*, pp. 95–98, 1973.
[23] J. A. Starzyk and E. Sliwa, "Hierarchic decomposition method for the topological analysis of electronic networks," *Int. J. Circuit Theory Applic.*, vol. 8, pp. 407–417, 1980.
[24] J. A. Starzyk, "Signal flow-graph analysis by decomposition method," *IEEE Proc. Electronic Circuits Sys.*, no. 2, pp. 81–86, 1980.
[25] J. C. Tiernan, "An efficient search algorithm to find the elementary circuits of a graph," *Comm. ACM*, vol. 13, pp. 722–726, Dec. 1970.
[26] M. K. Tsai and B. A. Shenoi, "Generation of symbolic network functions using computer software techniques," *IEEE Trans. Circuits Sys.*, vol. CAS-26, pp. 344–346, June 1979.
[27] H. A. Weinblatt, "New search algorithm for finding the simple cycles of a finite directed graph," *J. ACM*, no. 1, pp. 43–56, 1972.

Janusz A. Starzyk (SM'83) was born in Rybnik, Poland, on June 26, 1947. He received the M.S. degree in applied mathematics and Ph.d. degree in electrical engineering from Technical University of Warsaw, Warsaw, Poland, in 1971 and 1976, respectively.

He became a Research and Teaching Assistant in 1972 and Assistant Professor in 1977 at the Institute of Electronics Fundamentals, Technical University of Warsaw, Poland. From 1981 to 1983 he was a Post-Docorate Fellow and research engineer at the McMaster University, Hamilton, Canada. He joined the Department of Electrical and Computer Engineering, Ohio University, Athens, OH, as Associate Professor in 1983. His current research is in the areas of large scale networks, graph theory and fault analysis.

Agnieszka Konczykowska received the M.S. degree in applied mathematics and the Ph.d. in electrical engineering from the Technical University of Warsaw, Poland, in 1971 and 1977, respectively.

From 1971 to 1981 she had research and teaching activities at the same University. She is now with the CNET-Bagneux, France. Her main research interests are circuit theory, computer aided design, particularly using symbolic design methods and decomposition methods, and switched-capacitor circuits. She has authored or coauthored about 25 publications.