Grzymisław Cichowski
Janusz Starzyk
Edward Śliwa
Warsaw Technical University

COMPUTER IMPLEMENTATION OF TOPOLOGICAL METHOD IN THE ANALYSIS OF LARGE NETWORKS

1. INTRODUCTION

In recent years the various methods and user programs have been investigated, for finding symbolic network functions of linear active networks [1] - [3]. These methods based on the numerical evaluating of coefficient matrix determinant, allow to analyse effectively quite large networks keeping some network elements in symbolic form. Employment of sparse matrix technique allows also to obtain network functions as rational functions of the complex frequency variable [4] .

Topological methods, used in the symbolic network analysis first [5] [6], have been recognized ineffective because of too large analysis time and practical impossibility to analyse networks having more than 20 nodes. By topological methods we mean techniques computing network functions from the graph associated with the network. Both Mason and Coates signal flowgraph method and tree enumeration technique are topological methods. In the paper [7] the authors stated: "No mention has been made of programs using a topological tree enumeration formulation. While accurate [8], [9] and useful for sensitivity studies because of the explicit form of the coefficients of the polynomials, these programs have been found to be too slow to be of interest and are limited to very small circuits. Similar conclusions have been found to apply to programs based on flowgraphs methods [5]".

Graph partitioning schemes [10], [11], while have made some progress, did not give satisfactory results and no computer program using this approach has been developed.

Nevertheless advantages of the topological methods have caused that the investigations for improved computational techniques and efficiency of these methods was continued [12] [13]. But essential progress in this field was achieved only when hierarchical decomposition method for the graph associated with the network has been elaborated [24] [26], and when simple algorithms for the analysis of decomposed networks have been written. Program HADEN /Hierarchical Analysis of DEcomposable Networks/ which is presented in this paper allows to analyse the networks having tens nodes in the time comparable with the results obtained by numerical methods. The program is based on the hierarchical analysis method presented in [15].

2. BRIEF DESCRIPTION OF ALGORITHMS

Fig. 1a shows general program structure. To save computer memory /the program was divided into overlays; each blok showed on Fig. 1 a. forms a separate overlay. The main overlay performs two tasks: memory management and loading appropriate overlays to the central memory. Core requirements are dynamic and dependend on the size of analysed network. Assumed memory organization structure resembles that of SPICE and is shown on Fig. 1 b.

2.1. Data input and formation of network graph

HADEN is a user-oriented program. Its network description language is convenient and easy to learn. The network is described by the set of free-format cards; each card describes the network element or user's directive. After all information about the analysed network is introduced, the graph $G_1(X_1, E_1)$ of this network is being formed. Unistor representation of passive and these elements, which have an indefinite admittance matrix, is assumed in the program. These elements, which cannot be directly descibed by the indefinite admittance matrix /as e.g. ideal operational aplifiers/are represented by autonomic formal unistor models.

Such models are obtained from the flow-graph of these elements by replacing all the edges of the flow-graph with corresponding system of unistors. Examples of formal unistor models of choosen elements are shown in Tab.1.

2.2. Decomposition

The aim of the decomposition algorithm is determination of hierarchical decomposition of the graph.
This decomposition should provide the partition of the graph into the blocks, whose dimensions have been defined in advance.

This partition can be represented by means of the tree of decomposition [15] /Fig.2/. The number of levels of decomposition depends on the number of vertices of the given graph, the dimension of blocks and the filling of succesive levels.

The main idea of the algorithm consists in the determination of a sequence of bisections of graph.

The partition algorithm is realized in four following steps

1. $i:=1$, $L:=1$
2. a test whether a graph $G_i\left(X_i, E_i\right)$ should be decomposed; if not then go to 4
3. determination of bisection of G_i into G_{L+1} and G_{L+2}; $L:=L+2$
4. $i:=i+1$; if $i \leq L$ go to 2 , else end.

The most important point of the described algorithm is determination of the bisection.
The distance $d(x,x_o)$ between two vertices x, x_o is equal to the lenght of the shortest path between these vertices.
Let us denote by

$$S_i(x_o)=\left\{x \in X : d(x,x_o)= i\right\} , \quad i = 1,..,L(x_o)$$

the sets of vertices /whose distance from x_o is i /where $S_L \neq \emptyset$, $S_{L+1}=\emptyset$/

The diameter of a graph is defined as follows

$$D(x_o) = \max_{1 \leq i \leq I(x_o)} \bar{\bar{S}}_i(x_o)$$

The bisection of graph can be determined in the following maner

1/ we look for the vertex $x_o \in X$, which minimizes $D(x_o)$

2/ we determine sets S_i , $i = 1,\ldots,L$ for the vertex x_o

3/ we determine the sequence $B(i) = \bar{\bar{S}}_i$, $i = 1,\ldots L$

4/ one of the sets S_j such that $\bar{\bar{S}}_j \leq \bar{\bar{S}}_k$, $j,k \in (0,3L,0.7L)$ is chosen as the set of bisection vertices

5/ the previously determined set of vertices can be optimized in two ways

a/ the vertices from S_j incident with only one of sets S_{j-1} or S_{j+1} are removed

b/ we check the existence of bisection with the number of vertices less thes $\bar{\bar{S}}_j$ in the set $S_{j-1} \cup S_j \cup S_{j+1}$

We have observed that for the graphs of typical electrical networks the efficency of decomposition increases when we assume the vertex with maximal incidence degree as one of the bisection vertices, /usually, it is the reference node of the network/

2.3. Hierarchical analysis

The process of decomposition results in subgraphs the graph was divided into /further referenced as proper blocks/ and in the information about structure of interconnections of proper blocks, encoded in the structure of substitute graphs of bisections /defined below/ and the tree of decomposition [15]. The next stage following decomposition is a hierarchical analysis. The idea is as follows.

Let us suppose that graph G /a part of analysed network or, in the particular case, the network itself/ was partitioned to two subgraphs G_1 and G_2 /each of them might be, or not, further partitioned/. We may distinguish two disjoint subsets of vertices:

- set of block vertices B, consisting of the vertices crossed by lines of bisections or defined as terminal vertices.
- set of internal vertices I, consisting of remaining vertices of G.

In turn, the set B might be divided into two subsets: those in G_1 and in G_2 /of course, their intersection forms the set of bisection vertices of G/. After spanning a complete directed graphs over each of these two subgraphs, the substitute graph of bisection G^S, is obtained. Let us denote the sets of edges of those complete subgraphs corresponding with G_1 and G_2 by E_1 and E_2, respectively.

Our aim is to evaluate the set of directed k-trees T_V, where V is a family of k subsets of the block vertices; each of the sets of V consists of at least one vertex - reference node of a given component of a k-tree; remaining vertices are vertices belonging to the same component. Of course, all sets of V are disjoint.

Let us denote with P the set of all directed k-trees t of G^S, and such, that in the graph $t \cap E_i$, i = 1 or 2, there does not exist a directed path of length greater than one; such k-trees are called proper k-trees [15] [16].

It can be proved [15] [16], that the set of k-trees T_V may be evaluated from the formula:

$$T_V = \sum_{t \in P} T^1_{V_{1t}} \circ T^2_{V_{2t}} \qquad /1/$$

There are no duplications in the formula above. "o" denotes Wang product. $T^1_{V_{1t}}$ and $T^2_{V_{2t}}$ are sets of k-trees of G_1 and G_2

defined by V_{1t} and V_{2t}, respectively.

Families V_{it}, i = 1 or 2, can be found as follows. Let us suppose that the graph $t_i = t \cap E_i$, i = 1 or 2, has k components, each of them being a directed k-tree. Sets of family V_{it} i = 1 or 2, consist of vertices incident to edges of succesive components of t_i.

If any of G_1 or G_2 is sufficiently small, $T^i_{V_{it}}$, i = 1 or 2, can be evaluated directly. If it has been partitioned however, the procedure described above can be applied, with $G := G_i$ and $V := V_{it}$, $t \in P$.

If G is the graph of the entire network, family V follows, of course, from topological formula.

In HADEN, only the complex variable s is symbolic. So sets T_V, $T^i_{V_{it}}$, i = 1 or 2, appearing in /1/ are simply polynomials of one variable s, and Wang product can be replaced with normal polynomial multiplication.

More precise description of algorithms of hierarchic analysis can be found in [16].

3. GENERAL CHARACTERISTIC OF HADEN

HADEN was implemented on CDC-CYBER 73. Core requirements depend on the size of analysed network, from 34000B for networks having 15-20 nodes, up to about 50000B - 60000B words of core for 100-150 nodes.

Program was tested for networks having up to 130 nodes.

HADEN can evaluate four basic network functions: voltage and current transfer functions, transimpedance, transadmitance and impedance between given pair of nodes.

There are no serious restrictions for the structure of interconnections of elements. Because of assumed heuristic algorithm of decomposition, however, it is the most efficient for longish structures resembling /after removing the reference node/ an irregular cigar /Fig.3; discontinous lines denote the most probable partitions/.

HADEN is the first program performing topological analysis of large networks connecting in an effective way automatic partitioning of network graph with hierarchic analysis of decomposed graph, and equipped with convenient language of network description.

4. RESULTS FROM HADEN

HADEN was tested for dependencies of time consumed for the entire analysis (Fig.4) and decomposition (Fig. 5) versus network size and dependency of time used for the entire analysis v.s. blocks sizes the network was partitioned to (Fig. 6)

Tests perfomed show that time elapsed for decomposition is proportional to the product of numbers of nodes and levels of decomposition. A colloraly follows that time used for graph partitioning on a single level of decomposition linearly depends on the network size.

The dependency of the hierarchical analysis time v.s. network size resembles function of type const $\times n^a$ where n is the number of nodes and a is a constans little more than 2. This distinguishes described algorithm from the direct analysis with its exponential dependency.

Fig. 6 shows strong dependency of the time of analysis from blocks sizes the network was partitioned to.

With large blocks, the time of their analysis is large. From the other hand if blocks are too small, the time of analysis of interconnections of blocks increases. So there exsists a minimum, which, for the network tested /a RC ladder of 52 nodes/, was about 6-7 nodes in block.
In order to evaluate the efficiency of the program HADEN the examples from papers [1] and [3] were choosen. Programs NAPPE and SNAP were run on CDC 6500, while SNAPEST on CDC Cyber 74 and HADEN on CDC Cyber 73. The computer time required by HADEN is given in Table II and may be compared with the time required by the other three programs to analyze the same ciruit /Fig.7/.

TABLE II

Number of nodes	Running time in seconds			
	HADEN	SNAPEST	NAPPE	SNAP
11	1.16	.298	1.23	3.27
13	1.58	.370	1.67	19.8
15	1.81	.463	2.35	≈ 140.
17	2.55	.567	3.12	≈ 16min.

A few other examples analyzed by HADEN are shown in Fig.8 and 9 The computer time reguired for these examples was s and s.

5. SUMMARY

A new method and a program of topological analysis of linear networks is presented in the paper. Results achieved allow the assumption that topological methods will remain useful in electronic network design. HADEN is only the first version of computer implementation of hierachical decomposition method. It can be seen, however, that this method can compete with numerical methods in time consumption.
Some special advantages of topological methods make them better.

The elaborated method is a ground for further researches on making use of these possibities of topological methods, that are unattainable with other methods as eg.

1. the great accuracy
2. possibility of use of approximate symbolic analysis
3. possibility of generating functions describing macromodels.
4. Possibility of efficient tolerance analysis, tuning and centering with the aid of the quantile numbers arithmetics.

REFERENCES

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

[2] E.M.Butler, E.Cohen, N.J.Elias, J.J.Golembski, R.G.Olsen "CAPITOL - Circuit Analysis Program Induding TOLerances" Proc. IEEE Int. Symp. on Circuits and Systems, pp.570-574, 1977.

[3] M.I.Tsai, B.A.Shenoi "Generation of symbolic network functions using computer software techniques" IEEE Trans. on Circuits and Systems, pp.344-346 , June 1979.

[4] T.M.B.Neil, D.J.Bond, "Inversion of the nodal admittance matrix by the use of state equations" Circuit Theory and Applications, vol.1, pp.249-261, 1973.

[5] L.P.McNamee, N.Potash, "A user's and programmer's manual for NASAP", Univ. California, Los Angeles, Rep.68-38, Aug.1968.

[6] P.M.Lin, G.E.Alderson, "SNAP-A computer program for generating symbolic network functions" School Elec.Eng., Purdue Univ. Lafayette, ind., Tech.Rep. TR-EE70-16, Aug.1970.

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

[8] D.A.Calahan "Linear network analysis and realization digital computer programs, and instruction manual", Univ. Illinois Bull., vol.62, Feb.1965.

[9] E.V.Sorensen, "A preliminary note on the analytical network program ANPI", Lab.Circiut Theory, Tech.Univ.Denmark, Lyngby, Rep. LKT 23, 1967.

[10] S.D.Shien, S.P.Chan, "Topological formulas of symbolic network functions for linear active networks" Proc.IEEE Int.Symp. on Circuit Theory, pp.95-98, 1973.

[11] W.R.Dunn, Jr.,S.P.Chan, "Analysis of active networks by a subgraph-construction technique", presented at the Asilomar Conf., 1968.

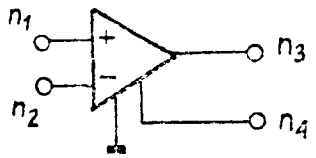
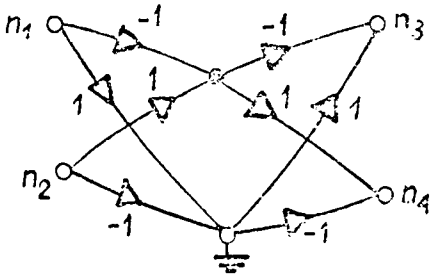
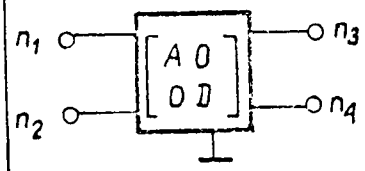
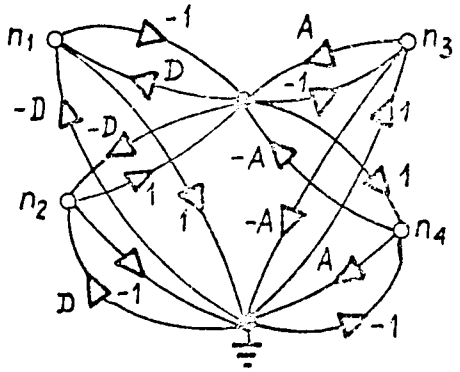
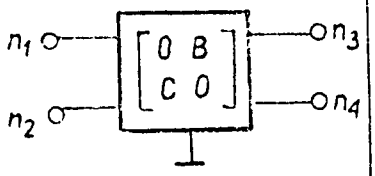
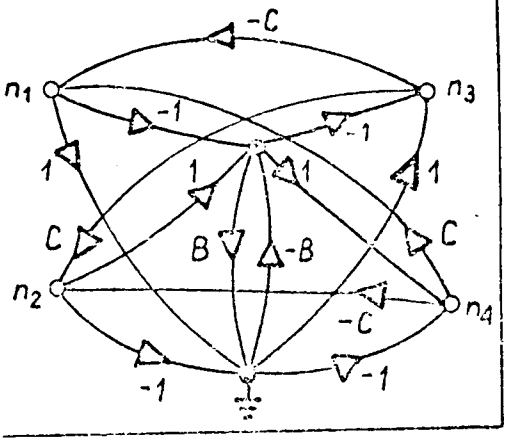
[12] A.Kończykowska, J.Starzyk , "Determination of structural number of a partitioned graph", Part I and II" Arch. Elektrot. z.2, 1975 /in Polish/.

[13] R.R.Mielke, "A new signal flowgraph formulation of symbolic network functions", IEEE Trans. Circuits and Systems, vol. CAS-25, pp 334-340, June 1978.

[14] J.Starzyk, "Signal flow-graph analysis by decomposition method" IEE Journal on Electronic Circuits and Systems, April 1980.

[15] J.Starzyk, E.Sliwa , "Topological analysis by hierarchic decomposition method", Fourth Intern. Symp.on Network Theory, Bled, 1979.

[16] J.Starzyk, E.Sliwa "Hierarchic decomposition method for the topological analysis of electronic network" sent to Intern.Journal of Circuit Theory and Applications.

Element	Formal unistor model
Ideal operational amplifier 	
Converter 	
Inverter 	

Tab.1. Formal unistor models

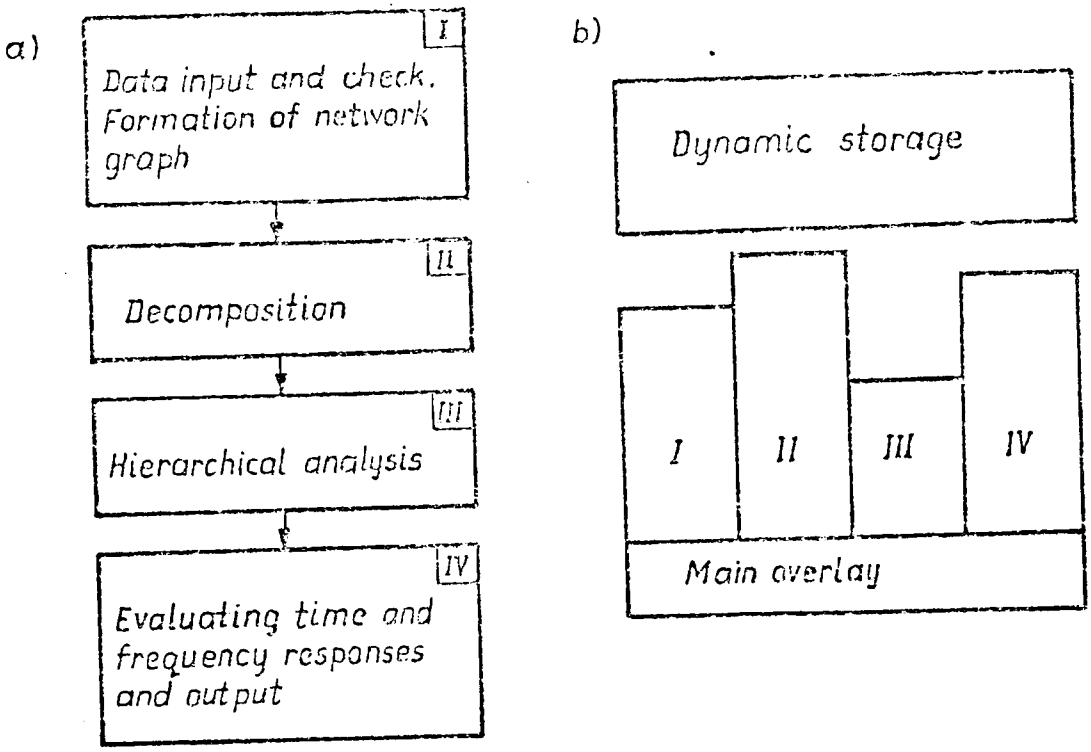


Fig.1. a/ Program structure b/ Memory organisation

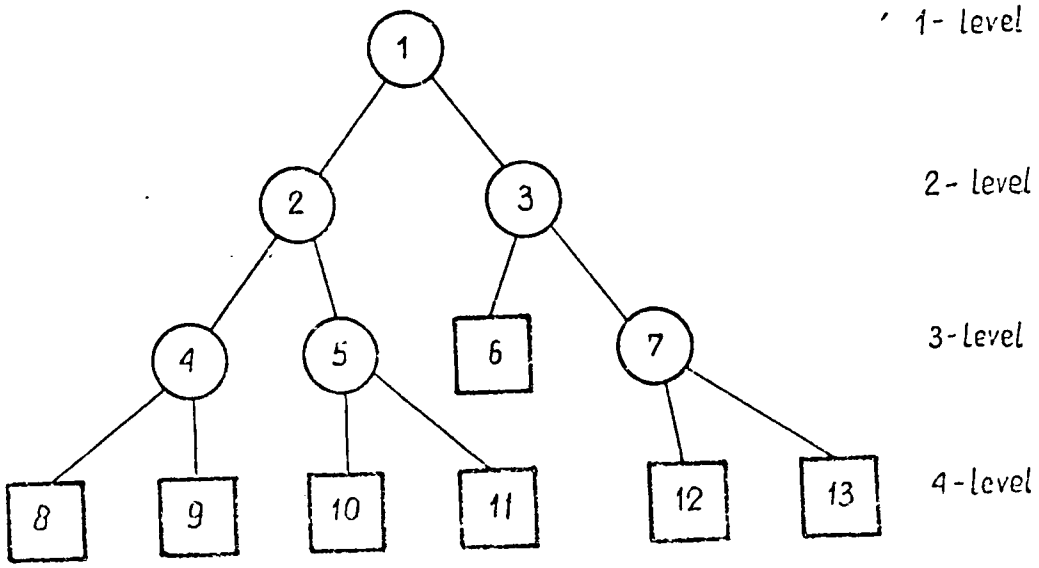


Fig.2. Tree of decomposition.

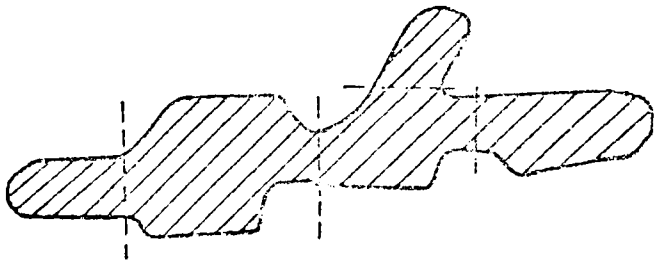


Fig.3. Quasi - optimal network shape for assumed decomposition algorithm

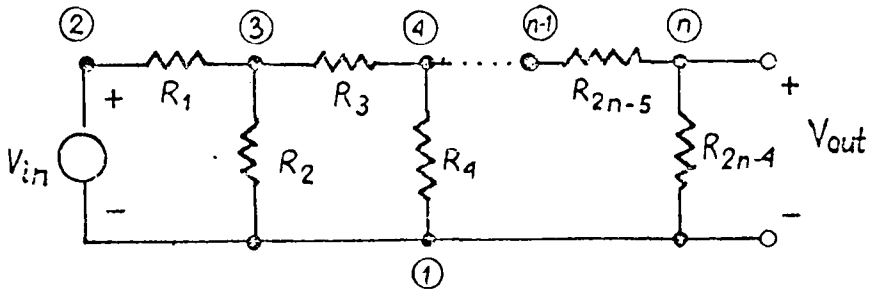


Fig.4. Ladder network

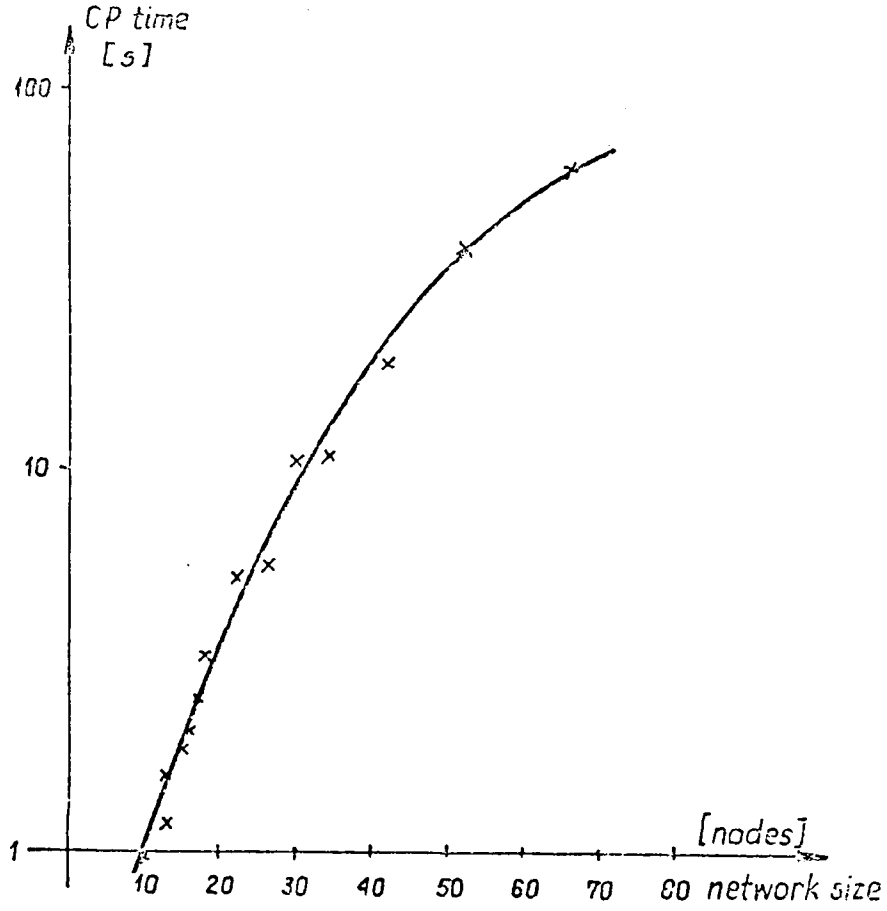


Fig.5. Running time used for entire analysis v.s network size /tested on RC ladders/.

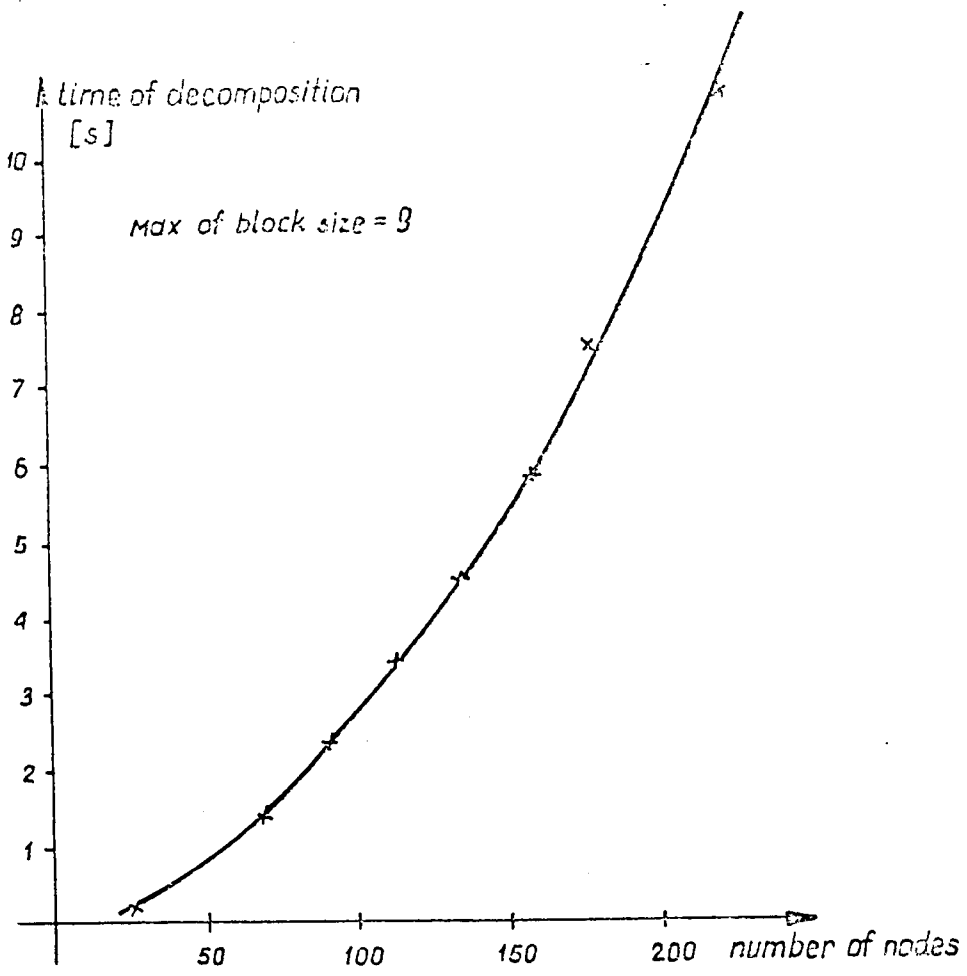
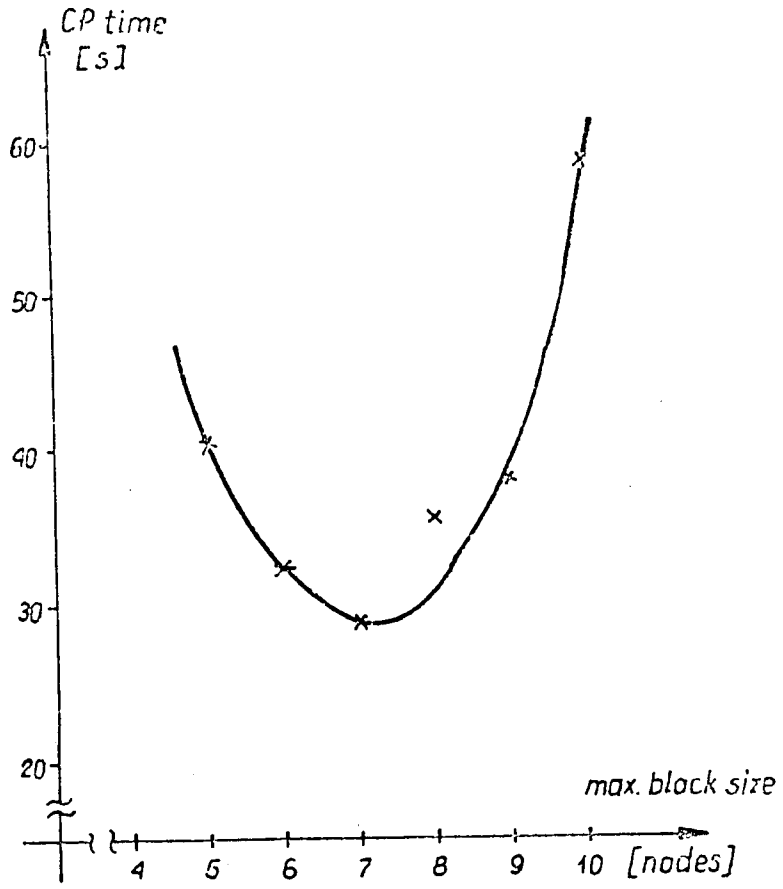


Fig.6. Running time of decomposition as a function of network size.



Rys.7. Running time of hierarchical analysis as a function of proper block size /for the RC ladder of 52 nodes/

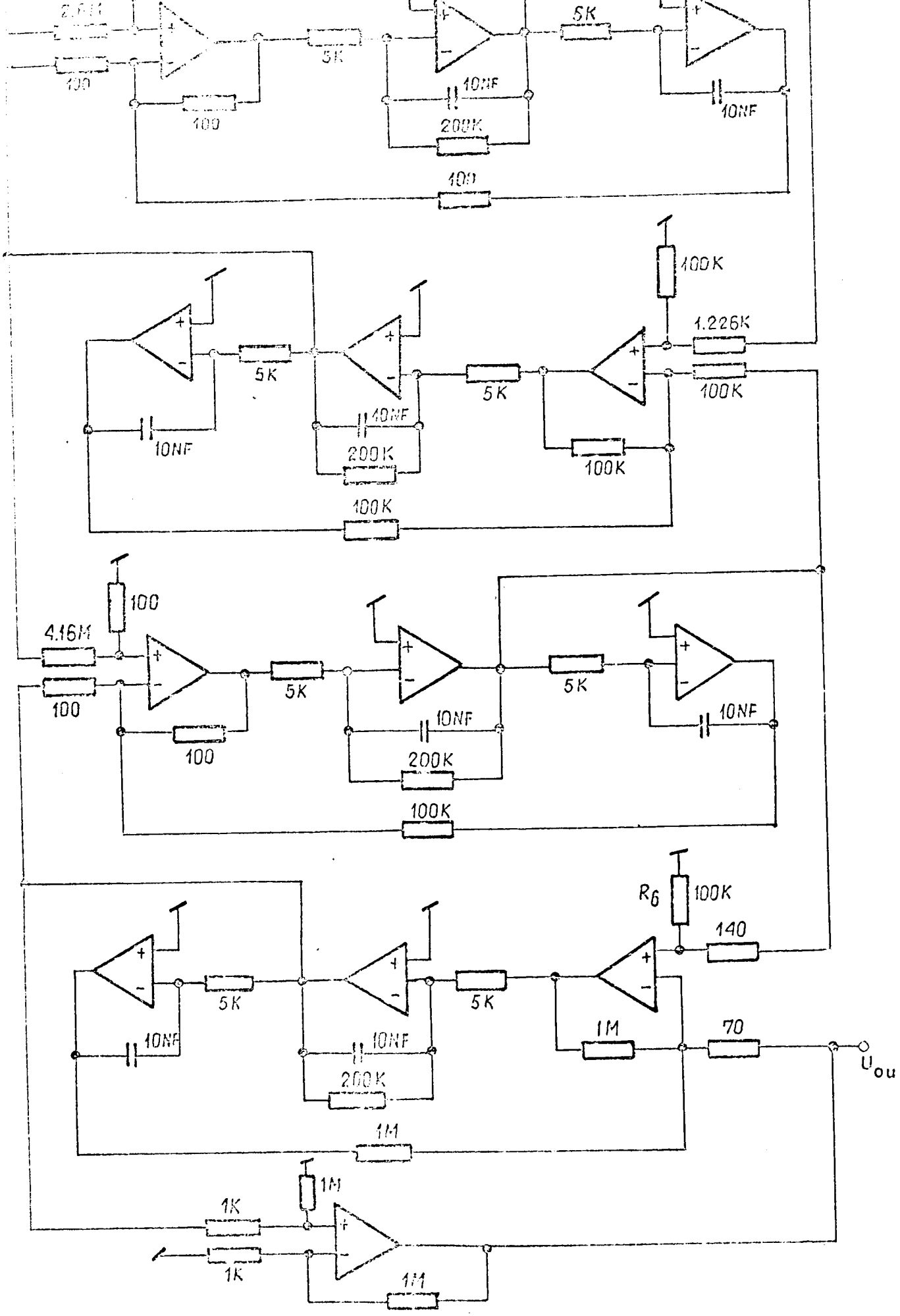


Fig.8. Bandpass filter example.

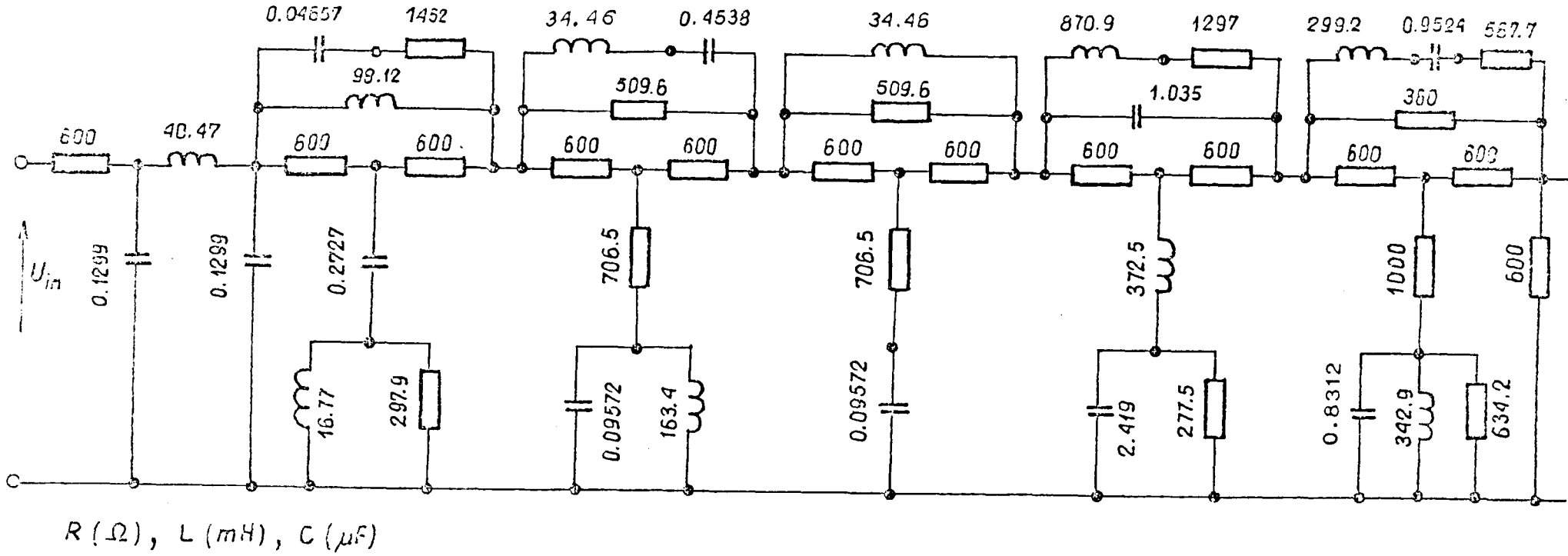


Fig.9. Passive network example.