Control
and Cybernetics
VOL. 15 (1986) No. 3-4

A procedure to obtain the symbolic expression of input-output representations from a signal flow graph

by

ANTONIO LEPSCHY

GIORGIO ROMANIN JACUR*

UMBERTO VIARO

Istituto di Elettrotecnica e di Elettronica
Università di Padova
Padova, Italy

The problem of determining the input-output transferences of a composite system represented by a signal flow graph is considered. Each transference is first expressed as the ratio of two polynomials in terms of numerators and denominators of path and loop transferences. Then, to apply Mason's formula it is necessary: (i) to find all input-output paths, (ii) to find all loops in the graph, and (iii) to detect the loop-loop and loop-path contacts. To solve these problems some operations, that may be carried out on the original graph (i.e. its connection matrix), are suggested. In particular, a new heuristic procedure to find the loops is described. All operations are applied to a graph of reasonable complexity, thus showing the practicality of the method.

1. Introduction and problem statement

This paper deals with the problem of determining the effects of system inputs on one or more outputs when the system under study is composed of interacting parts whose analytic description is known and whose mutual relationships are represented by a directed graph. In particular, linear, time-invariant, finite-dimensional, multivariable systems are considered and it is assumed that the overall system is represented by a signal flow graph, i.e. by a directed graph whose nodes correspond to system characteristic quantities and whose branches are weighted with suitable tranferences (if the original representation is in terms of the usual state equations, then these transferences are either constants or equal to 1/s).

Our objective is to express symbolically the transfer functions relating the applied inputs to the outputs of interest. Of course, the problem could be solved by manipulating the state or input-output representations of the component parts

* also with LADSEB-CNR, Padova, Italy

according to their actual connections. However, this often involves cumbersome procedures. It is therefore very useful to have at our disposal methods leading to a more efficient solution at least with reference to particular classes of systems (even if they may not prove to be equally satisfactory when different classes are considered).

The procedure presented in this paper seems to be particularly suited to computer-assisted or fully automatic application, but may be applied also manually in the simplest cases. Essentially, it is based on the well-known Mason's formula [1, 2, 3] which is rewritten in a way that allows us to obtain directly the corresponding numerator and denominator polynomials from numerators and denominators of branch transferences.

The paper includes proposals for searching paths and loops and for determining automatically their contacts. In this regard, referring for example to the loop search, it may be observed that when the graph connection matrix is not sparse, the number of loops can be very large so that their search is a hard job even with the most efficient techniques. In many practical cases, however, the connection matrix is reasonably sparse and the number of loops is not too high. In this case, a method may be considered effective if the solution of the problem is achieved in an easy way, even though it does not always ensure this result. The method presented in this paper has these characteristics; moreover, its application is always worthwhile because the original problem is decomposed into simpler subproblems for which exhaustive methods can be used more economically.

2. Symbolic form of the transference

According to Mason's formula, the transference $W_{ij}(s)$ relating the output corresponding to node i to the input corresponding to node j may be written in the form:

$$W_{ij}(s)=\frac{\sum_{h=1}^{\pi_{ij}}\left\{P_{ijh}(s)\prod_{k=1}^{\lambda}[1-L_k(s)]\right\}^*}{\left\{\prod_{k=1}^{\lambda}[1-L_k(s)]\right\}^*} \tag{1}$$

where

π_{ij} is the number of paths from j to i,

$P_{ijh}(s)$ is the transference of the h-th path from j to i,

λ is the number of loops in the graph,

$L_k(s)$ is the loop transference of the k-th loop,

and the star indicates that all terms containing products of the type $L_r(s)\,L_s(s)$ or $L_r(s)\,P_s(s)$ corresponding to pairs of touching loops and/or paths are to be cancelled.

$P_{ijh}(s)$ and $L_k(s)$ are rational functions of s:

$$P_{ijh}(s)=\frac{R_{ijh}(s)}{Q_{ijh}(s)}; \quad L_k(s)=\frac{N_k(s)}{M_k(s)} \tag{2}$$

where $R_{ijh}(s)$, $Q_{ijh}(s)$, $N_k(s)$ and $M_k(s)$ are polynomials.

By using the above symbols, the denominator of (1) takes the form:

$$1-\sum\frac{N_k(s)}{M_k(s)}+\sum{}^{*}\frac{N_k(s)N_l(s)}{M_k(s)M_l(s)}-\sum{}^{*}\frac{N_k(s)N_l(s)N_m(s)}{M_k(s)M_l(s)M_m(s)}+\ldots=$$
$$=\frac{\overline{\Pi}M_k(s)-\Sigma N_k(s)\prod_{l\neq k}M_l(s)+\Sigma\{[N_k(s)N_l(s)]^{*}\prod_{m\neq k,\,l}M_m(s)\}-\ldots}{\prod_{k=1}^{\lambda}M_k(s)} \tag{3}$$

with the above-mentioned meaning of the stars.

The right-hand side numerator polynomial of (3) may be rewritten as

$$\Sigma^{*}\prod_{k=1}^{\lambda}F_k(s) \tag{4}$$

where the polynomials $F_k(s)$ in each product correspond to either $-N_k(s)$ or $M_k(s)$ and the summation is over all combinations of λ factors of the type $-N_k(s)$ and $M_k(s)$ related to different values of subscript k, excluding those with two (or more) polynomials $N_k(s)$ related to touching loops.

The numerator of (1) may be written in the form:

$$\frac{\left\{\left[\sum_{h=1}^{\pi_{ij}}R_{ijh}(s)\cdot\prod_{k\neq h}Q_{ijk}(s)\right]\left[\Sigma^{*}\prod_{k=1}^{\lambda}F_k(s)\right]\right\}^{*}}{\left[\prod_{h=1}^{\pi_{ij}}Q_{ijh}(s)\right]\left[\prod_{k=1}^{\lambda}M_k(s)\right]} \tag{5}$$

so that (1) may be expressed as the ratio of two polynomials:

$$W_{ij}(s)=\frac{\left\{\left[\sum_{h=1}^{u_{ij}}R_{ijh}(s)\cdot\prod_{k\neq h}Q_{ijk}(s)\right]\left[\Sigma^{*}\prod_{k=1}^{\lambda}F_k(s)\right]\right\}^{*}}{\left[\prod_{h=1}^{\pi_{ij}}Q_{ijh}(s)\right]\left[\Sigma^{*}\prod_{k=1}^{\lambda}F_k(s)\right]} \tag{6}$$

To compute $W_{ij}(s)$ it is then necessary:

1) to find all paths from node j to node i;
2) to find all loops in the graph;
3) to detect the loop-to-loop and loop-to-path touching conditions;
4) to determine polynomials $R_{ijh}(s)$, $Q_{ijh}(s)$, $N_k(s)$ and $M_k(s)$ related to all paths and loops.

A specific section will be devoted to each of these problems. They will be preceded by a section in which some useful preliminary simplifications of the original graph are suggested.

3. Preliminary manipulation of the graph

Although the procedure described in the following sections could directly be applied to the original signal flow graph, a simple manipulation of this graph often allows us to reduce appreciably the computations involved.

It is straightforward to associate with the given signal flow graph the connection matrix Q whose generic element q_{hk} is equal to 1 if there is a branch from node k to node h and is equal to zero otherwise. Clearly, since Q cannot account for parallel branches (if any), these must first be replaced by single branches whose transference is equal to the sum of the transferences of the original parallel branches. This operation will be repeated whenever a parallel arises from a graph modification at any step of the procedure.

First of all, isolated nodes and isolated selfloops are to be removed since they do not influence the transference (and, in fact, the corresponding loop transference would be a factor both of the numerator and of the denominator of Mason's formula). This operation can directly be performed on matrix Q: an isolated node corresponds to a row and a column of equal index whose elements are all zeros; an isolated selfloop corresponds to a row and a column of equal index with a 1 on the main diagonal and all other elements equal to zero.

In order to carry out further simplifications as well as the next steps of the procedure, it is useful to subdivide the nodes into the following classes:

— input nodes I (sources);
— output nodes O (sinks);
— single-input single-output nodes S;
— convergence nodes C with at least two entering branches and one outgoing branch;
— divergence nodes D with one entering branch and two or more outgoing branches;
— multiple-input multiple-output nodes M with at least two ingoing and two outgoing branches.

For this purpose, then sum m of all the elements of each row and column of Q is computed. Then, each node is classified according to Table I depending on the m values of the row and the column whose indices are equal to that of the node.

TABLE I(a)

row \ column	$m=0$	$m=1$	$m>1$
$m=0$	—	I	I
$m=1$	O	S	D
$m>1$	O	C	M

The different types of node are illustrated in Fig. 1.

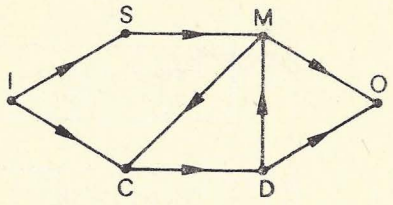

Fig. 1. Types of nodes

Next, nodes I different from j and nodes O different from i are removed together with the branches connected with them; this operation and the consequent re-classification of the remaining nodes is to be repeated until no further branch elimination is possible.

The S nodes are then absorbed by replacing the pair of branches incident to each S node by a single branch with transference equal to the product of the two related transferences. As a consequence, new parallels may arise, as shown in Fig. 2: as already said at the beginning, these parallels should in turn be reduced to single branches. Again, such an operation and the consequent node classification must be repeated till all S nodes have been eliminated. Notice that a situation of the kind depicted in Fig. 2 can easily be detected from matrix Q: indeed, after node 4 has been absorbed, the resulting connection matrix is equal to the corresponding part of the previous connection matrix; so, the presence of a parallel from node 3 to node 5 is revealed by the fact that element q_{53} is already equal to 1 before the absorption of node 4.

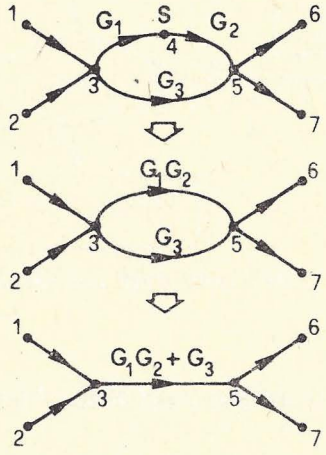

Fig. 2. S-node absorption and reduction of parallel branches

The above operations are summarized in the diagram of Fig. 3.

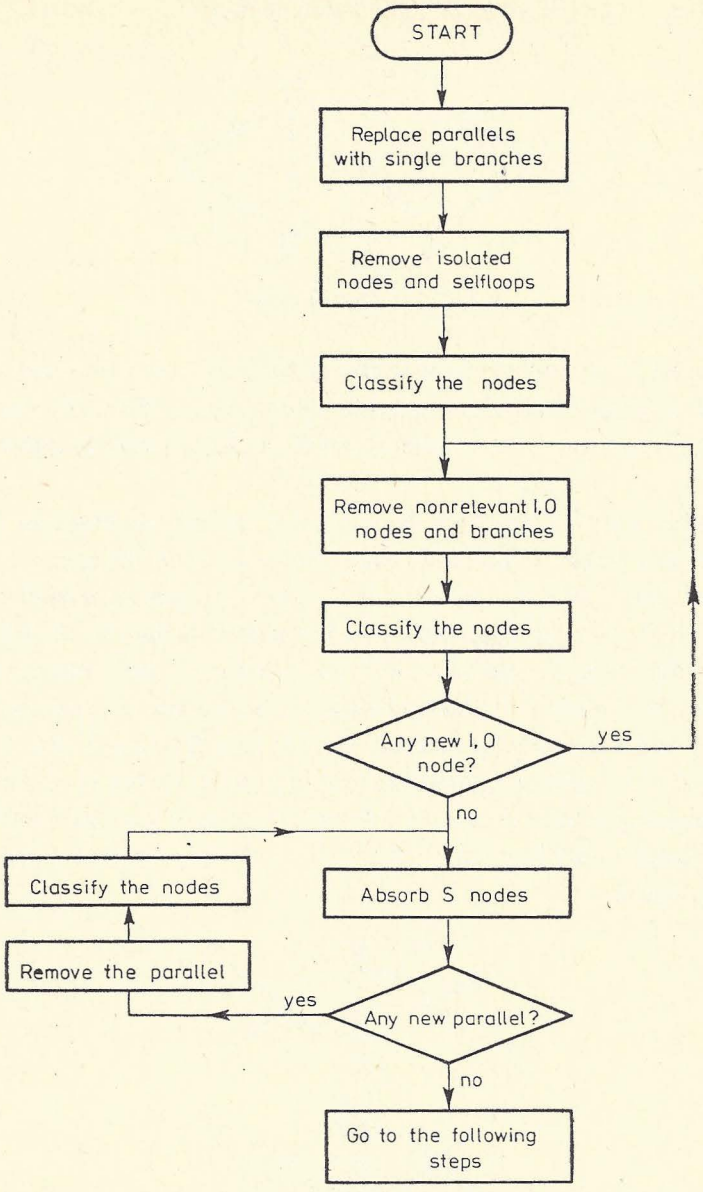

Fig. 3. Preliminary graph simplification

4. Search of the I/O connected subgraph and of the strongly connected subgraphs

The path and loop search is much simplified by considering only the I/O connected subgraph; this comprises the nodes that are reachable from input j and from which output i may be reached. Within this subgraph, the strongly connected

subgraphs (S.C.S.) are then identified, which is particularly useful in the search of the loops since a loop may belong to one and only one S.C.S..

To do this, the reachability matrix Q_R is obtained from Q according to the procedure described in [4], i.e.

$$Q_R=(\mathrm{I}+Q)^{2^k} \tag{7}$$

where k is the lowest integer not less than $\log_2(\nu-1)$ and ν is the number of nodes of the simplified graph. Alternatively, the cascade algorithm of Farbey et al. [5, 6] may be applied in a simplified version according to the Boolean nature of the problem.

Then, the column of Q_R corresponding to input node j and its row corresponding to putput node i are considered; the nodes corresponding to unit entries both in that column and in this row belong to the I/O connected subgraph. All other nodes may be removed from the graph, which entails deleting (or replacing by zeros) the corresponding rows and columns of Q and Q_R; the resulting matrices will be denoted by $\hat{Q}$ and $\hat{Q}_R$.

To determine the strongly connected subgraphs, it is enough to consider that the S.C.S. containing the generic node k is formed by all nodes corresponding to unit entries both in the k-th row and in the k-th column.

5. Search of the input-output paths

For the sake of simplicity, we shall proceed by examples. Consider the graph of Fig. 4a: there are two S.C.S.; one is formed by nodes c, d, e and the branches connecting them, the other is formed by nodes f, g and the related branches. The original graph may be replaced by a graph in which each S.C.S. is condensed to a single node (Fig. 4b). It is easily realized that the paths from j to i belong to one of the following categories:

1) a path consisting of a single branch connecting directly node j with node i;
2) paths formed by branch sequences that do not touch any node belonging to an S. S.C.S. (cf., for example, path j, a, b, i, in Fig. 4);
3) paths formed by branch sequences that, although they touch some S.C.S., do not include any branch of such S.C.S. (cf., for example, path j, c, a, b, i of Fig. 4a);
4) paths containing branches belonging to some S.C.S. (cf., for example, path j, c, d, f, g, i of Fig. 4a).

This classification is clearly exhaustive. A path of type 1 is readily found since it corresponds to the fact that entry q_{ij} of $\hat{Q}$ is equal to 1. The search of the paths of types 2, 3 and 4 may be carried out, in a first phase, on the condensed graph, which is in general much simpler than the original one. Each path of the condensed graph not touching any condensed node corresponds to a single path of type 2 formed by the same branches. For the paths touching condensed nodes a more

detailed analysis is necessary; in fact: (i) a branch of such paths may correspond to distinct branches of the original graph (for instance, the branch from j to c' in Fig. 4b corresponds to the branches from j to c and from j to d); (ii) within a condensed S.C.S. more distinct paths may exist that connect the node where the input-output path enters the S.C.S. with the node from which such path leaves the S.C.S.; if these two nodes coincide (cf. path j, c, a, b, i of Fig. 4a) for all touched S.C.S., then the corresponding path is of type 3 and no search is necessary within the S.C.S.;

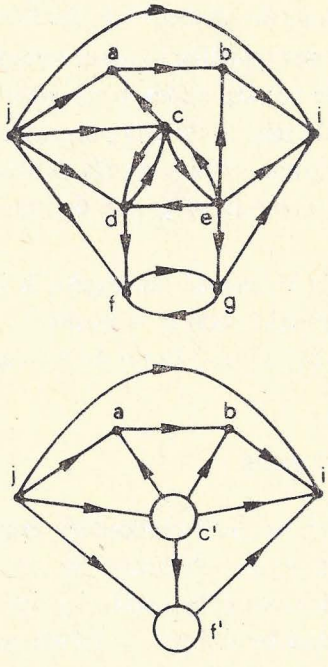

Fig. 4. Graph condensation for path search

if, on the contrary, the considered terminal nodes are different for at least one S.C.S., it is necessary to search for the subpaths connecting these nodes within the S.C.S..

In conclusion, the suggested procedure may be summarized as follows:

(i) determine the paths from j to i in the condensed graph;
(ii) find the condensed nodes 'crossed' by these paths, i.e. such that in the original graph the entrace node and the departure node of the S.C.S. are different;
(iii) find the subpaths connecting the nodes of interest within the crossed S.C.S.;
(iv) determine the input-output paths by combining the paths of the condensed graph with the internal subpaths of the crossed S.C.S..

In this way, the search is carried out on structures that are simpler than the original one, which often requires a reduced amount of computation. According to the complexity of each subproblem, one of the methods described in [7, 8, 9] as well as other methods may be used.

6. Search of the loops

The search may conveniently be restricted to each of the strongly connected subgraphs.

Before applying exhaustive methods, it is suggested to use a heuristic procedure, called the Base Frame method, described by the present authors in [4]. This method is of easy application and often leads to the complete solution of the problem; even if this is not the case, it simplifies the original problem thus facilitating the application of general methods like the one described in [9].

By recalling the node classification of section 3, we first observe that after absorbing the S nodes, an S.C.S. may contain only C, D and M nodes or may reduce to a selfloop. Now, by indicating the subgraph branches with pairs of letters where the first denotes the node from which the considered branch departs and the second the node reached by the branch, a Base Frame (B. F.) is a connected subgraph formed by branches of the types CC, CM, CD, MD, DD. An example of a B. F. within an S.C.S. is given in Fig. 5 where the B. F. branches are represented by solid lines and the others by dashed lines.

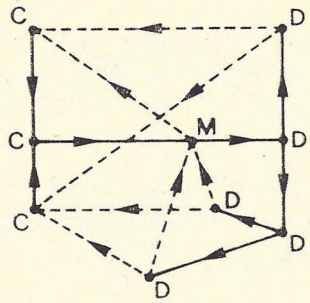

Fig. 5. Example of base frame (solid lines)

If all nodes of the S.C.S. are M nodes then no B. F. branch exists; on the other hand, more separate B. F. may be identified within the S.C.S..

The most general structure of a B. F. is of the type depicted in Fig. 5: it is formed by a convergent arborescence (with only C nodes) followed by a divergent arborescence (with only D nodes); they are connected either by a CD branch or by an M node. Clearly, a B. F. may well reduce to a single arborescence or to a single CD branch.

Although the search of the B. F. may be carried out directly on the connection matrix corresponding to the considered S.C.S. (which is obtained from $\hat{Q}$ by deleting the rows and columns corresponding to the other nodes), for ease of exposition we shall refer only to operations on the graph.

To form a B. F. we may start from a CD branch or an M node (if any). In the first case all DD branches connected with the D node of the selected branch and all CC branches connected with its C node are added to the CD branch; this ope-

ration is repeated for the new terminal C and D nodes of the already determined part of the B. F.. In the second case, all CM and MD branches incident to the chosen M node are looked for and the procedure develops as in the previous case.

After all B. F. have been found, the remaining branches are analysed and those incident to pairs of nodes of the same B. F. are identified. It is easily seen that each of these branches forms one and only one loop with the B. F. branches connecting the considered pair of nodes and will therefore be called 'closing branch'. The part of the loop within the B. F. is easily found by following the route indicated by the arrows from the node of arrival of the closing branch to the first M or D node and by going in the direction opposite to the arrows from the node of departure of the closing branch to the terminal M or D node of the first subpath. Notice that the path within the B. F. is unique and that the procedure outlined above does not involve any choice between alternatives.

The loops corresponding to the closing branches are recorded and the closing branches are removed from the subgraph. Then the nodes are classified again, the new S nodes, if any, are absorbed and the resulting parallel branches, if any, are reduced to single branches (which may require a new classification of their departure and arrival nodes), a new B. F. is searched and so on.

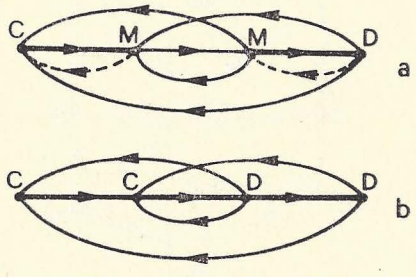

Fig. 6. Extension of the B. F. (bold lines) after elimination of the dashed closing branches

The procedure, which is exemplified in Fig. 6, is iterated until:

1) all branches have been included in a loop (as closing branches or B. F. branches);
2) the graph reduces to a selfloop;
3) no new loops may be found even if not all branches have been included in a loop.

In the last case it is necessary to use a general exhaustive method like the one described in [9].

The suggested procedure is summarized by the flow chart of Fig. 7. The method may fail when no B. F. exists or no closing branch is found. These situations, however, are rare and, anyway, a limited effort is required before abandoning the attempt and using a different method. In all other cases the B. F. method seems to be worthwile since either it leads to the complete solution of the problem in an easy way or it simplifies the original graph thus facilitating appreciably the search of the remaining loops by means of other methods.

Some extensions of the B. F. method will be illustrated in the Appendix.

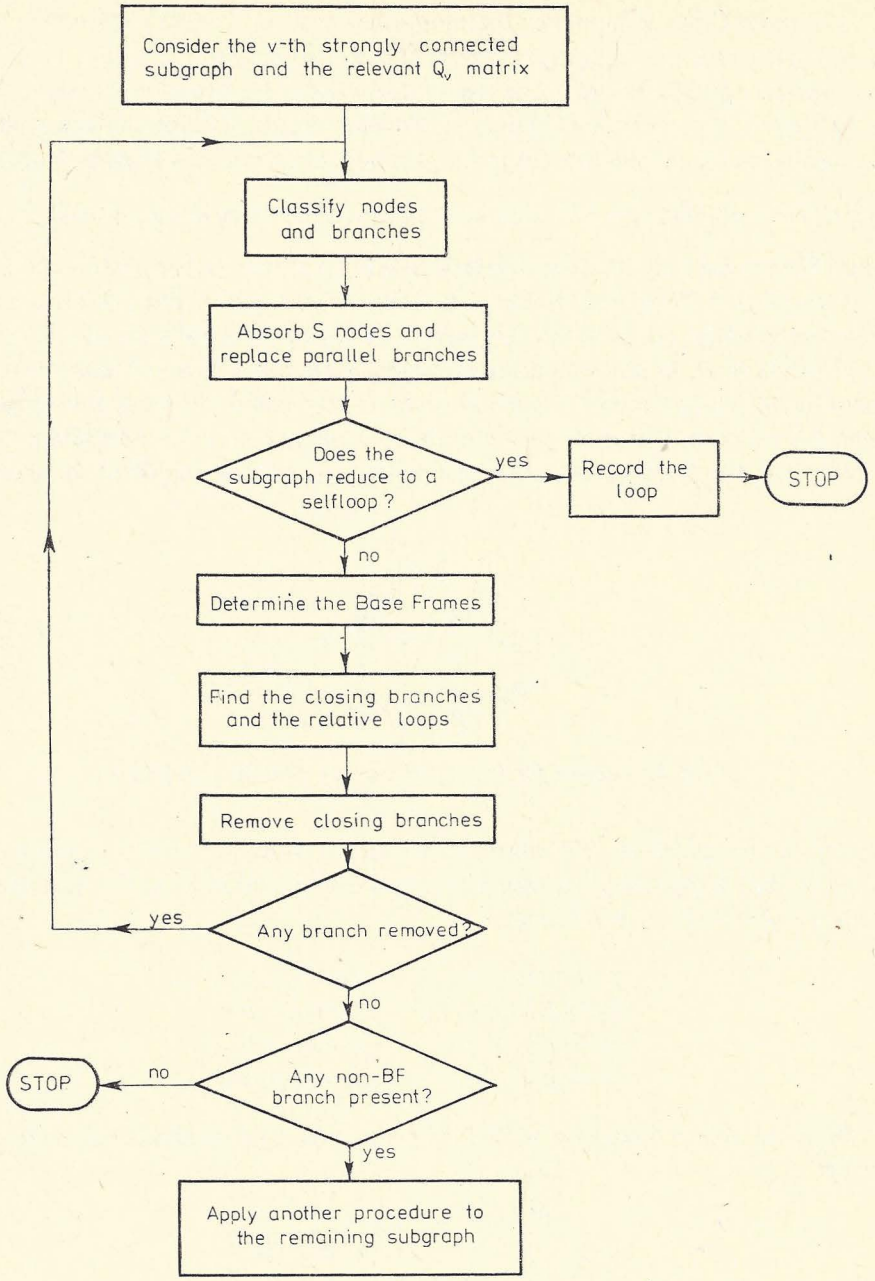

Fig. 7. The Base Frame method

7. Detection of the contacts

To apply Mason's formula it is necessary to determine which loop-loop or loop-path pairs have at least one common node (touching condition). To this purpose it is suggested to carry out some elementary algebraic operations.

Notice first that a loop may touch another loop or an I/O path only at nodes of strongly connected subgraphs. Let n be the total number of nodes of all S.C.S ($n<v$) and, as previously said, λ be the total number of loops and π_{ij} the total number of I/O paths. A rectangular $(\lambda+\pi_{ij})\times n$ Boolean matrix A is formed whose element a_{pq} is equal to 1 if node q belongs to the p-th path or loop and is equal to 0 otherwise. The Boolean product $B=AA^T$ is then computed whose element b_{pq} equals $\sum_{r=1}^{n} a_{pr}\, a_{rp}$, where the products and the sum are Boolean. B is a square $(\lambda+\pi_{ij})\times(\lambda+\pi_{ij})$ matrix; b_{pq} is equal to 1 if and only if the p-th loop or path touches the q-th loop or path; therefore B will be called the contact matrix. Of course, it is not necessary to compute all elements of B; in particular, the elements of the main diagonal and those corresponding to path-path contacts are not interesting; moreover, since B is symmetric, only the relevant elements above the main diagonal need be computed. Notice, finally, that the Boolean nature of the operations can be exploited to reduce the

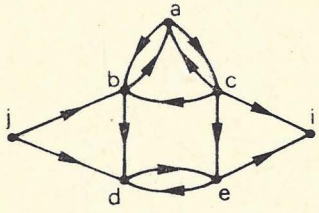

Fig. 8. Example for the analysis of the touching conditions

amount of computation. The above procedure is illustrated with reference to the graph of Fig. 8. By using the methods suggested in sections 5 and 6, the following paths p_k and loops l_k are found:

$$\begin{array}{ll} p_1=(j,b,a,c,i) & l_5=(a,b,a) \\ p_2=(j,b,a,c,e,i) & l_6=(a,c,a) \\ p_3=(j,b,d,e,i) & l_7=(a,c,b,a) \\ p_4=(j,d,e,i) & l_8=(d,e,d) \end{array}$$

Since the nodes belonging to the strongly connected subgraph are a, b, c, d, e, matrix A is

$$\begin{array}{c|ccccc} & a & b & c & d & e \\ \hline p_1 & 1 & 1 & 1 & 0 & 0 \\ p_2 & 1 & 1 & 1 & 0 & 1 \\ p_3 & 0 & 1 & 0 & 1 & 0 \\ p_4 & 0 & 0 & 0 & 1 & 1 \\ l_5 & 1 & 1 & 0 & 0 & 0 \\ l_6 & 1 & 0 & 1 & 0 & 0 \\ l_7 & 1 & 1 & 1 & 0 & 0 \\ l_8 & 0 & 0 & 0 & 1 & 1 \end{array} \qquad (8)$$

Then the contact matrix B is given by:

	p_1	p_2	p_3	p_4	l_5	l_6	l_7	l_8
p_1	·	·	·	·	1	1	1	0
p_2	·	·	·	·	1	1	1	1
p_3	·	·	·	·	1	0	1	1
p_4	·	·	·	·	0	0	0	1
l_5	·	·	·	·	·	1	1	0
l_6	·	·	·	·	·	·	1	0
l_7	·	·	·	·	·	·	·	0
l_8	·	·	·	·	·	·	·	·

(9)

where only the entries of interest are reported.

8. Evaluation of the transference

The application of (6) requires the previous identification of: (i) the I/O paths, (ii) the loops, and (iii) the touching conditions. This has been done in the preceding sections. The evaluation of $W_{ij}(s)$ involves the following steps;

1) For any path from j to i, the numerator $R_{ijh}(s)$ of the transference $P_{ijh}(s)$ ($h=1, 2, \ldots, \pi_{ij}$) is expressed as the product of the numerators of the relevant branch transferences; the same is done for the denominator $Q_{ijh}(s)$ of $P_{ijh}(s)$.
2) The numerators $M_k(s)$ and denominators $N_k(s)$ of each loop transference $L_k(s)$ are similarly formed.
3) By taking into account the contact matrix B, expression

$$\Sigma^* \prod_{k=1}^{\lambda} F_k(s) \tag{4}$$

is computed, where $F_k(s)$ equals either $-N_k(s)$ or $M_k(s)$ and the products with at least two factors $[-N_k(s)]$ corresponding to touching loops are excluded.
4) The denominator of (6) is formed by multiplying expression (4) by the product of the denominators of all path transferences: $\prod_{h=1}^{\pi_{ij}} Q_{ijh}(s)$.
5) Each numerator $R_{ijh}(s)$ is multiplied by $\prod_{k \neq h} Q_{ijk}(s)$ and by the addenda of (4) not containing factors $[-N_l(s)]$ such that $b_{hl}=1$.
6) The numerator of (6) is formed by adding the terms determined at step 5.

The above procedure will be illustrated with reference to the already considered example of Fig. 8. For the sake of simplicity we neglect steps 1 and 2, and let $P_h(s)=R_h(s)/Q_h(s)$, $h=1, 2, 3, 4$, and $L_h(s)=N_h(s)/M_h(s)$, $h=5, 6, 7, 8$.

By taking into account matrix (9), expression (4) becomes

$$M_5M_6M_7M_8 - N_5M_6M_7M_8 - M_5N_6M_7M_8 - M_5M_6N_7M_8 - \\ - M_5M_6M_7N_8 + N_5M_6M_7N_8 + M_5N_6M_7N_8 + M_5M_6N_7N_8 \tag{10}$$

The denominator of the I/O transference equals the product of (10) and $Q_1Q_2Q_3Q_4$.

By referring again to matrix (9) for determining the path-loop contacts, we may write the numerator of the I/O transference in the form:

$$\begin{aligned} &R_1Q_2Q_3Q_4M_5M_6M_7M_8 - R_1Q_2Q_3Q_4M_5M_6M_7N_8 + \\ &+Q_1R_2Q_3Q_4M_5M_6M_7M_8 + Q_1Q_2R_3Q_4M_5M_6M_7M_8 + \\ &-Q_1Q_2R_3Q_4M_5N_6M_7M_8 + Q_1Q_2Q_3R_4M_5M_6M_7M_8 + \\ &-Q_1Q_2Q_3R_4N_5M_6M_7M_8 - Q_1Q_2Q_3R_4M_5N_6M_7M_8 + \\ &-Q_1Q_2Q_3R_4M_5M_6N_7M_8 \end{aligned} \tag{11}$$

9. An illustrative example

The main phases of the suggested procedure will be illustrated in the sequel with regard to a meaningful example. For the sake of simplicity, only the graphs resulting from the various matrix manipulations will be reported without entering into the details of the operational aspects.

The original signal flow graph of the considered example is shown in Fig. 9 together with the classification of the nodes. The input and output nodes of the transference of interest are assumed to be node 1 and node 20, respectively.

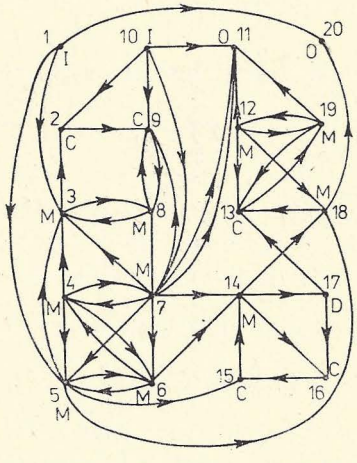

Fig. 9. Original graph and first node classification

First of all, the I nodes different from node 1 (i.e. node 10) and the O nodes different from node 20 (i.e. node 11) as well as the branches connected with them, are eliminated and the S nodes (i.e. node 2) are absorbed. The resulting graph is reported in Fig. 10, where the pair of dashed-line branches incident to the S node 2 are replaced by a solid-line branch from node 3 to node 9.

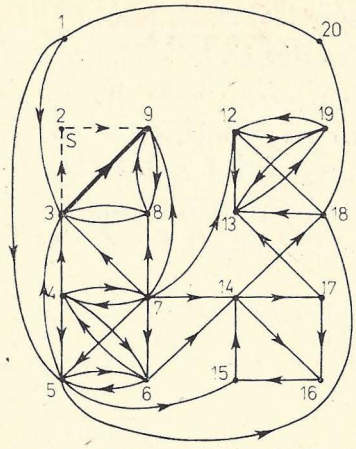

Fig. 10. Preliminary graph simplification

Next, the parts not belonging to the I/O connected subgraph (in the specific case, nodes 3, 8, 9 and the branches connected with them) are removed. This result is shown in Fig. 11.

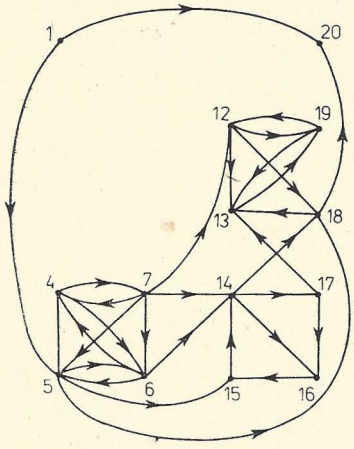

Fig. 11. I/0 connected subgraph

The strongly connected subgraphs are then identified: they are represented in Fig. 12a. After this, the graph is condensed by replacing each strongly connected subgraph by a supernode (cf. Fig. 12b where the 'identity' of the nodes within each supernode has been preserved). First, the path search is carried out on the acyclic condensed graph; then the relevant subpaths inside each S.C.S. are determined: in the case considered, there is only one subpath for each pair of terminal nodes.

In conclusion, the I/O paths are formed by the following sequences of nodes (for completeness also the sequences relative to the condensed graph are reported):

$p_1=(1, 20)$

$p_2=(1, A_5, C_{18}, 20)=(1, 5, 18, 20)$

$p_3=(1, A_5, B_{15}, B_{14}, C_{18}, 20)=(1, 5, 15, 14, 18, 20)$
$p_4=(1, A_5, B_{15}, B_{17}, C_{13}, C_{18}, 20)=(1, 5, 15, 14, 17, 13, 19, 12, 18, 20)$
$p_5=(1, A_5, A_6, B_{14}, C_{18}, 20)=(1, 5, 6, 14, 18, 20)$
$p_6=(1, A_5, A_6, B_{14}, B_{17}, C_{13}, C_{18}, 20)=(1, 5, 6, 14, 17, 13, 19, 12, 18, 20)$
$p_7=(1, A_5, A_7, C_{12}, C_{18}, 20)=(1, 5, 6, 4, 7, 12, 18, 20)$
$p_8=(1, A_5, A_7, B_{14}, C_{18}, 20)=(1, 5, 6, 4, 7, 14, 18, 20)$
$p_9=(1, A_5, A_7, B_{14}, B_{17}, C_{13}, C_{18}, 20)=(1, 5, 6, 4, 7, 14, 17, 13, 19, 12, 18, 20)$

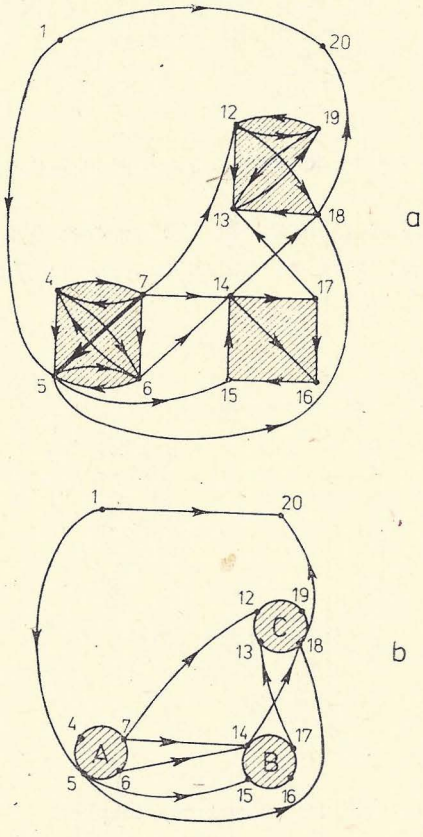

Fig. 12. Strongly connected subgraphs (shaded) and corresponding condensation

In order to find the loops, the B. F. method is applied separately to each S.C.S.. The procedure is illustrated in Fig. 13: the B. F. are represented by bold lines. Concerning subgraphs B and C, the B. F. method leads immediately to the complete solution of the problem since all non-B. F. branches close a loop. (Notice that in the case of subgraph B, the S nodes 15 and 17 must be absorbed beforehand). Concerning subgraph A, the initial B. F. allows us to find two loops; after the corresponding closing branches have been removed, the B. F. may be extended, and the remaining four loops may be found.

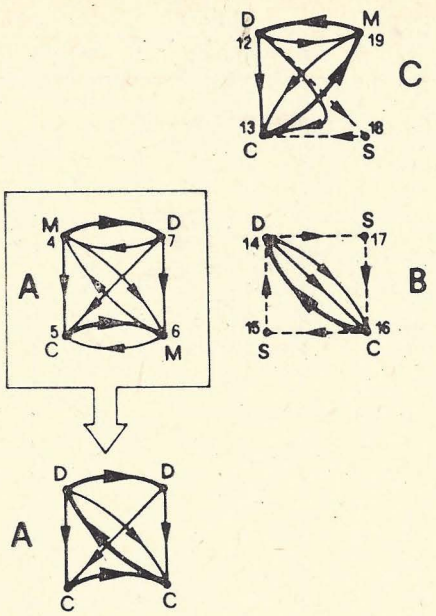

Fig. 13. Loop search using the B. F. method. The B. F. branches are in bold lines. The pairs of dashed lines incident to S nodes have been replaced by solid-line branches

The loops are therefore the following:

subgraph A: $l_{10}=(4, 7, 4)$
$l_{11}=(5, 6, 5)$
$l_{12}=(4, 6, 4)$
$l_{13}=(4, 5, 6, 4)$
$l_{14}=(4, 7, 6, 4)$
$l_{15}=(4, 7, 5, 6, 4)$

subgraph B: $l_{16}=(14, 16, 15, 14)$
$l_{17}=(14, 17, 16, 15, 14)$

subgraph C: $l_{18}=(12, 19, 12)$
$l_{19}=(13, 19, 13)$
$l_{20}=(12, 13, 19, 12)$
$l_{21}=(12, 18, 13, 19, 12)$

Matrix A, as defined in section 7, is reported in Table I. By multiplying A by its transpose, we find the contact matrix B which allows us to determine the touching conditions. The interesting part of matrix B is given in Table II. Expression (6) of the I/O transference may finally be determined as already exemplified in section 8.

10. Conclusions

Some simple operations have been suggested to facilitate the determination of the input-output transferences of a composite system represented by a signal flow graph.

Tab. 1

	3	4	5	6	7	8	9	12	13	14	15	16	17	18	19
p_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p_2	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
p_3	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0
p_4	0	0	1	0	0	0	0	1	1	1	1	0	1	1	1
p_5	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0
p_6	0	0	1	1	0	0	0	1	1	1	0	0	1	1	1
p_7	0	1	1	1	1	0	0	1	0	0	0	0	0	1	0
p_8	0	1	1	1	1	0	0	0	0	1	0	0	0	1	0
p_9	0	1	1	1	1	0	0	1	1	1	0	0	1	1	1
l_{10}	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
l_{11}	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
l_{12}	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
l_{13}	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
l_{14}	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
l_{15}	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
l_{16}	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
l_{17}	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
l_{18}	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
l_{19}	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
l_{20}	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
l_{21}	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1

Tab. 2

				l_{10}	l_{11}	l_{12}	l_{13}	l_{14}	l_{15}	l_{16}	l_{17}	l_{18}	l_{19}	l_{20}	l_{21}
p_1	.	.	.	0	0	0	0	0	0	0	0	0	0	0	0
p_2	.	.	.	0	1	0	1	0	1	0	0	0	0	0	1
p_3	.	.	.	0	1	0	1	0	1	1	1	0	0	0	1
p_4	.	.	.	0	1	0	1	0	1	1	1	1	1	1	1
p_5	.	.	.	0	1	1	1	1	1	1	1	0	0	0	1
p_6	.	.	.	0	1	1	1	1	1	1	1	1	1	1	1
p_7	.	.	.	1	1	1	1	1	1	0	0	1	0	1	1
p_8	.	.	.	1	1	1	1	1	1	1	1	0	0	0	1
p_9	.	.	.	1	1	1	1	1	1	1	1	1	1	1	1
l_{10}	.	.	.	.	0	1	1	1	1	0	0	0	0	0	0
l_{11}	.	.	.	.	.	1	1	1	1	0	0	0	0	0	0
l_{12}	.	.	.	.	.	.	1	1	1	0	0	0	0	0	0
l_{13}	.	.	.	.	.	.	.	1	1	0	0	0	0	0	0
l_{14}	.	.	.	.	.	.	.	.	1	0	0	0	0	0	0
l_{15}	.	.	.	.	.	.	.	.	.	0	0	0	0	0	0
l_{16}	.	.	.	.	.	.	.	.	.	.	1	0	0	0	0
l_{17}	.	.	.	.	.	.	.	.	.	.	.	0	0	0	0
l_{18}	.	.	.	.	.	.	.	.	.	.	.	.	1	1	1
l_{19}	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1
l_{20}	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1
l_{21}	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

The underlying idea has been that of decomposing the original problem into easier subproblems thus reducing the amount of computation.

In particular, a heuristic method, called the Base Frame method, has been described which in most practical cases appreciably simplifies the loop search.

All suggested operations have been applied to a signal flow graph of reasonable complexity.

Appendix

Extension of the B.F. method

Some improved versions of the B. F. method may be successful even in case of failure of the procedure described in section 6.

One of these consists in finding a set of branches that cannot belong to the same loop as is the case for the branches entering (leaving) the same node. The B. F. method is then applied successively to the graphs obtained from the original by removing all but one branches of the considered set. In this way either an M node becomes a D (C) node or a C (D) node becomes an S node, thus allowing the extension of the previous B. F.. Observe that a loop may be detected more times. The efficiency of this procedure depends on the choice of the mentioned of branches; a valid criterion is that of referring to the node with the largest number of entering (outgoing) branches. Alternatively, a two-branch loop may be looked for; this is easily done with reference to matrix $Q=[q_{hk}]$ by checking the condition $q_{hk}\, q_{kh}=1$; then the B. F. is applied to both the graphs obtained by removing either of the loop branches.

As an example of application of the above variants let us consider the simple but complete graph of Fig. A1a. Fig. A1b illustrates the first variant with reference to the branches entering node 1 (loop 3, 2, 3 is found two times). Fig. A1c shows the operations corresponding to the second variant with reference to the two-branch loop 1, 3, 1 (now, loops 1, 2, 1 and 2, 3, 2 are found twice).

According to a different approach, all nodes of each remaining B. F. are condensed to a supernode and the resulting parallel branches are replaced by a single branch. The B. F. method may then be applied to the condensed graph. Each loop of the condensed graph (which is formed by a path within a B. F. of such graph and by a closing branch) may correspond to more loops of the original graph: these, in fact, contain only one of the parallel branches that have been replaced by a single branch in the condensation process. Therefore, the number of loops of the original graph corresponding to the same loop of the condensed graph is given by the product of the numbers of branches of each condensed parallel. Also this variant may fail to solve completely the problem, but usually leads to an advantageous simplification of the graph which will finally be analysed using a general method.

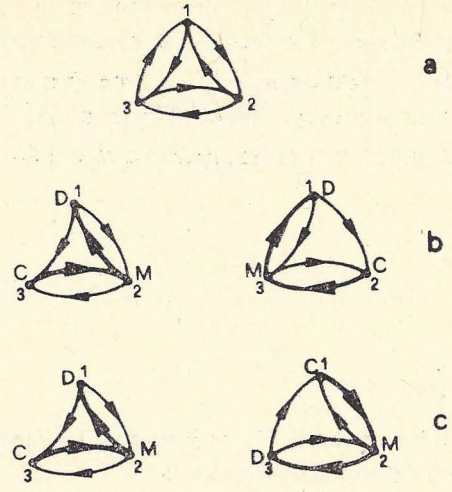

Fig. A1

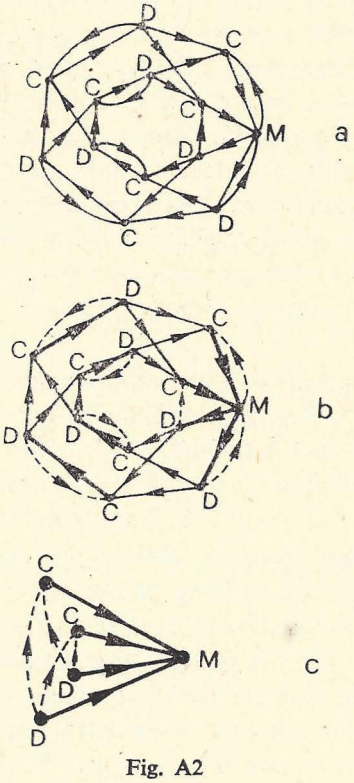

Fig. A2

The procedure is exemplified in Fig. A2. In particular, Fig. A2a represents the initial strongly connected subgraph; Fig. A2b shows the B. F. branches (bold lines) and the closing branches (dashed lines). Seven loops are identified with the B. F. method of section 6, but the procedure stops before the complete solution of the

problem. In Fig. A2c the condensed graph with the relevant B. F. (bold lines) is reported; again, the closing branches are represented by dashed lines. In this case, the second application of the Base Frame method is completely successful.

References

[1] MASON S. J. Feedback Theory: Some Properties of Signal Flow Graphs. *Proc. I.R.E.*, **9** (1953) 9, 1144–1156.
[2] MASON S. J. Feedback Theory: Further Properties of Signal Flow Graphs. *Proc. I.R.E.*, **44** (1956) 7, 920–926.
[3] MASON S. J., ZIMMERMANN H. J. Electronic Circuits, Signals and Systems. New York, J. Wiley, 1960.
[4] LEPSCHY A., ROMANIN JACUR G., VIARO U. Derivation of the Characteristic Polynomial of a Multivariable System from its Signal Flow Graph. Proc. Symp. Application of Multivariable System Techniques, The Institute of Measurement and Control, Plymouth, 31 Oct.–2 Nov., 1984.
[5] FARBEY B. A. et al. The Cascade Algorithm for Finding All Shortes Distances in a Directed Graph. *Management Science*, **14** (1967) 1, 19–28.
[6] HU T. C. A Decomposition Algorithm for Shortest Paths in a Network. *Operations Research*, **16** (1968) 1, 91–102.
[7] BELLERT S., WOZNIACKI H. Analiza i synteza układów elektrycznych. Metoda liczb strukturalnych. Warszawa, Wydawn. Naukowo-Techniczne, 1968.
[8] BOMBI F., FORNASINI E., LEPSCHY A. Un Procedimento per il Calcolo della Transferenza di un Grafo di Flusso di Segnale con la Formula di Mason. *Alta Frequenza*, **44** (1975) 7, 370–375.
[9] READ R. C., TARJAN R. E. Bounds on Backtrack Algorithms for Listing Cycles, Path, and Spanning Trees. *Networks*, **5** (1975), 237–252.

Procedura otrzymywania wyrażeń symbolicznych reprezentujących zależności wejścia-wyjścia w grafie przepływowym

W pracy rozważono zagadnienie określania wag krawędzi grafu przepływowego reprezentującego badany system. Każda taka waga jest najpierw określana jako iloraz dwóch wielomianów określających odpowiednio wagi ścieżek i cykli w grafie. Następnie, przy stosowaniu formuły Mason'a należy: (1) znaleźć wszystkie ścieżki od „wejścia" do „wyjścia", (2) znaleźć wszystkie cykle, (3) wykryć wszystkie połączenia pomiędzy cyklami oraz pomiędzy cyklami i ścieżkami. Podano sposób rozwiązania powyższych zadań, opisując między innymi nową heurystyczną metodę poszukiwania cykli w grafie.

Процедура получения символических выражений, отображающих зависимости входа-выхода в поточном графе

В работе рассматривается вопрос определения весов ребер поточного графа, отображающего исследуемую систему. Каждый вес определяется, в первую очередь, как частное двух многочленов, отображающих соответствующе весы путей и циклы в графе. Затем, используя формулу Масона следует: (1) найти все пути от „входа" до „выхода", (2) найти все циклы, (3) обнаружить все соединения между циклами, а также между циклами и путями. Дается способ решения выше указанных задач, описывая, в том числе, новый эвристический метод поиска циклов в графе.