SIAM J. SCI. STAT. COMPUT.
Vol. 9, No. 2, March 1988

THE CONSISTENT INITIALIZATION OF DIFFERENTIAL-ALGEBRAIC SYSTEMS*

CONSTANTINOS C. PANTELIDES†

Abstract. The initial values for variables in mixed Differential-Algebraic (DAE) systems must satisfy not only the original equations in the system but also their differentials with respect to time. Whether or not this additional requirement actually imposes extra constraints on the initial values depends on the particular problem. This paper derives a criterion for determining whether differentiation of a subset of the equations of a nonlinear DAE system provides further constraints to be satisfied by the initial values. A graph-theoretical algorithm is proposed to locate those subsets of the system equations which need to be differentiated.

Key words. differential-algebraic equations, initialization, assignment problem

AMS(MOS) subject classification. 65L05

1. Introduction. The differential-algebraic (DAE) systems considered here are of the general form

$$f(x, \dot{x}, y, t) = 0 \tag{1}$$

where $x, \dot{x} \in R^n$, $y \in R^m$, $f: G \subseteq R^n \times R^n \times R^m \times R \to R^{n+m}$.

For a set of initial conditions $(x_0, \dot{x}_0, y_0)$ to be consistent, it must satisfy system (1) at an initial time t_0:

$$f(x_0, \dot{x}_0, y_0, t_0) = 0. \tag{2}$$

Note that here the term "initial conditions" is used to refer to the vector $(x_0, \dot{x}_0, y_0)$ rather than simply to (x_0, y_0). This is because in the general case, the information available on the condition of system (1) at initial time t_0 may actually involve elements of vector $\dot{x}_0$. For instance, starting from an equilibrium position can be expressed as $\dot{x}_0 = 0$, with x_0 and y_0 determined by solving (2). Also no assumption is made in this paper regarding linearity of (1) with respect to $\dot{x}$.

Condition (2), however, is not always sufficient for consistency. Differentiating some or even all of the original equations produces new equations which must also be satisfied by the initial conditions. This need not necessarily constrain the vector $(x_0, \dot{x}_0, y_0)$ further: differentiation can also introduce new variables (time-derivatives of $\dot{x}$ and y) and it may well be the case that the new equations can be satisfied for all possible values of the initial conditions and appropriate choices of values for the new variables. Thus, in this case no useful information is generated by differentiation.

The following two simple examples should help illustrate the above argument:

Example 1. The original DAE system is:

$$\dot{x} = x + y, \qquad 0 = x + 2y + a(t)$$

where $a(t)$ is a given continuous and differentiable function of time. Here no useful information can be obtained by differentiating either or both equations in the system.

* Received by the editors September 23, 1985; accepted for publication (in revised form) March 13, 1987.

† Department of Chemical Engineering, Imperial College of Science and Technology, London SW7 2BY, England.

Consider, for instance, the differential of the second equation:

$$0 = \dot{x} + 2\dot{y} + \dot{a}(t).$$

This can be satisfied for all initial values of the variable $\dot{x}_0$ by choosing a value of $-(\dot{x}_0 + \dot{a}(t_0))/2$ for the newly introduced variable $\dot{y}_0$. It therefore suffices that the initial condition vector $(x_0, \dot{x}_0, y_0)$ simply satisfy the original system of equations at time t_0.

Example 2. The DAE system is:

$$\dot{x}_1 + \dot{x}_2 = a(t), \qquad x_1 + x_2^2 = b(t)$$

where $a(t)$ and $b(t)$ are continuous and differentiable functions of time. The initial conditions $(x_{1_0}, x_{2_0}, \dot{x}_{1_0}, \dot{x}_{2_0})$ must also satisfy the equation obtained by differentiating the second equation with respect to time:

$$\dot{x}_1 + 2x_2\dot{x}_2 = \dot{b}(t).$$

This paper proposes an algorithm for analyzing the structure of the system equations and determining the minimal subset, differentiation of which may yield useful information in the sense that it imposes further constraints on the vector of initial conditions.

The next section establishes a criterion regarding the desirability of differentiating an equation subset. Section 3 introduces the graph-theoretical concepts that form the basis of the algorithm. The latter is fully described in § 4. Some examples of the application of the algorithm are given in § 5.

It should be noted here that the problem of determining consistent initial conditions has been associated in the past with DAE systems of index greater than unity (see Petzold [7], Lötstedt and Petzold [6] and Gear and Petzold [3] for a definition of index and discussion of its implications). Algorithms have been proposed to eliminate the problem by reducing the index to zero. This corresponds to converting the DAE system to a system of ordinary differential equations (ODE) and is achieved by differentiating all algebraic equations for a sufficient number of times. After each differentiation, all time derivatives of existing differential variables are immediately replaced by their equivalent expressions given by the system differential equations. Such indiscriminate differentiation is not only impractical for large systems but it may often be quite unnecessary, since the solution of DAE systems of index 1 is, in principle, no more difficult than the solution of ODE systems. Furthermore, as this paper aims to show, problems are caused by equation *subsets*, differentiation associated difficulties. Finally, the initialization of general DAE systems of index 1 can also pose problems (Example 2 is, in fact, such a system).

2. Differentiation of system subsets. In this section we consider the differentiation with respect to time of a subset of the original system (1). We define the new vector $z = (\dot{x}, y)$ and rewrite (1) as

$$f(x, z, t) = 0. \tag{3}$$

Let the subset of interest be

$$\bar{f}(\bar{x}, \bar{z}, t) = 0 \tag{4}$$

consisting of k equations, with $\bar{x} \in R^q$ and $\bar{z} \in R^l$.

Assuming that $(\bar{f}_{\dot{x}}: \bar{f}_{\bar{z}})$ has full row rank, k, and that (4) is differentiable, we differentiate (4) with respect to time to obtain:

$$\bar{f}_{\dot{x}}\dot{\bar{x}} + \bar{f}_{\bar{z}}\dot{\bar{z}} + \bar{f}_t = 0. \tag{5}$$

Let the row rank of $\bar{f}_{\bar{z}}$ be r; then

$$r \leqq \min(k, l). \tag{6}$$

Then by appropriate row operations (e.g., Gaussian elimination), (5) can be reduced to the form

$$A\dot{\bar{x}} + B\dot{\bar{z}} = a \tag{7a}$$

and

$$C\dot{\bar{x}} = b \tag{7b}$$

where $A \in R^r \times R^q$, $B \in R^r \times R^l$, $C \in R^{k-r} \times R^q$ are matrices and $a \in R^r$, $b \in R^{k-r}$ are vectors, all functions of $(\bar{x}, \bar{z}, t)$ in general.

Differentiation has introduced the new variables $\dot{\bar{z}}$ which do not occur in the original system (1); it has also created the new equations (7). We note that, because of (6), l is always at least as large as r and one can therefore partition vector $\bar{z}$ into $\bar{z}_1 \in R^r$ and $\bar{z}_2 \in R^{l-r}$, and matrix B into $B_1 \in R^r \times R^r$ and $B_2 \in R^r \times R^{l-r}$ such that B_1 is nonsingular. Solving the resulting equations, one obtains from (7a) an explicit expression for $\dot{\bar{z}}_1$:

$$\dot{\bar{z}}_1 = B_1^{-1}a - B_1^{-1}(A\dot{\bar{x}} + B_2\dot{\bar{z}}_2). \tag{8}$$

Now for any set of values of the variables $(\bar{x}, \dot{\bar{x}}, \bar{y})$, one can always find values for the new variables $\dot{\bar{z}}$ such that (7a) is satisfied. No useful information concerning the original variable set is contained in (7a).

However, since by assumption Rank $(\bar{f}_{\dot{x}}: \bar{f}_{\bar{z}}) = k$, matrix C in (7b) has full row rank, $k - r$. Thus (7b) constitutes $(k - r)$ new equations which the original variable set must satisfy together with the original set (1).

Therefore, the "latent" equations that must be satisfied by a set of consistent initial conditions can be generated by locating all subsets (4) of k equations such that

$$\text{Rank}\left(\frac{\partial \bar{f}}{\partial \bar{z}}\right) < k. \tag{9}$$

Since a system of N equations contains $2^N - 1$ nonempty subsets, clearly a systematic method is required to generate only those subsets with property (9). Here we propose a graph-theoretical algorithm which analyzes the structural properties of the given DAE system. The algorithm is based on the observation that, since r is bounded from above by the cardinality l of $\bar{z}$, a *sufficient* condition for a subset (4) to have property (9) is

$$l < k. \tag{10}$$

A subset of equations that satisfies criterion (10) is called *structurally singular* with respect to the variable subset $\{\bar{z}\}$. A structurally singular subset is called *minimally structurally singular* (MSS) if none of its proper subsets is structurally singular. A system of equations is called structurally singular with respect to a certain set of

variables if it contains a structurally singular subset with respect to the same set of variables.

3. An algorithm for locating MSS subsets. Criterion (10) can be simply interpreted to mean that one has to locate all subsets of k equations containing fewer than k distinct members of $\{\dot{x}, y\}$ so that the number of new equations generated upon differentiation of the subset exceeds the number of new variables.

The algorithm makes use of the following fundamental theorem of combinatorics due to Hall [4]:

THEOREM 3.1. (Systems of Distinct Representatives (SDR)). *Let* $V = \{V_1, V_2, \cdots, V_m\}$ *be a set of objects and* $S = \{S_1, S_2, \cdots, S_n\}$ *a set of subsets of* V. *Then each element of* S *can be assigned a different element of* V *if and only if every collection of* $k(\leqq n)$ *elements of* S *contains at least* k *distinct elements of* V.

In the present application of the SDR theorem, set V is identified to be $\{\dot{x}, y\}$ while each element of S corresponds to one equation in $\{f\}$ and is the subset of V occurring in this equation. From condition (10), it is clear that we are interested in collections of equations (i.e., subsets of (1)) which do *not* satisfy the necessary and sufficient condition for the existence of an assignment. More significantly, existing algorithms for constructing such assignments can be readily modified to locate the required equation subsets.

The algorithm to be presented here is best described in a graph-theoretical context. The reader is assumed to be acquainted with the basic concepts of graph theory such as nodes, edges, paths and bipartite graphs (see e.g., Harary [5]).

For the purposes of the algorithm the equations $\{f\}$ and the variables $\{\dot{x}, y\}$ are represented as two disjoint sets of nodes in a bipartite graph; the two sets will be referred to as the "E-nodes" and the "V-nodes," respectively. Edge $(i-j)$ exists if equation i contains variable j.

An *assignment* is a set of edges $(i-j)$ such that no node i or j appears in more than one edge in the set. Edges in the assignment are called *matching edges*. A node is *exposed* if it does not appear in any matching edge. An *augmenting path* is a path with exposed nodes at both ends and alternating nonmatching and matching edges between them; in the trivial case, the path may just consist of only one (nonmatching) edge between two exposed nodes. An assignment is *complete* if it leaves no E-node exposed; otherwise it is called a *partial* assignment.

Algorithms for constructing complete assignments generally start with a partial assignment (often the empty set). They then attempt to construct an augmenting path from an exposed E-node to an exposed V-node. If such a path is found, then the matching edges in the path are replaced in the assignment by the nonmatching edges; the number of edges in the assignment is increased by one (hence the term "augmenting path"); the process is known as *reassignment*.

Very efficient procedures have been devised to construct augmenting paths in bipartite graphs. One of them, based on a depth-first search algorithm (Duff [1]), is outlined as Algorithm 3.2 using a pseudocode notation. In order to invoke the recursive procedure AUGMENTPATH, one initially designates all nodes in the graph as "uncoloured" and sets flag PATHFOUND to FALSE. ASSIGN (j) is assumed to contain the equation to which variable j is assigned in the current partial assignment (zero if the corresponding V-node is still exposed). Assuming that E-node i is exposed, the procedure invocation

AUGMENTPATH (i, PATHFOUND)

returns PATHFOUND = TRUE if an augmenting path emanating from i exists and

FALSE otherwise. Vector ASSIGN contains the new partial assignment, while all E- and V-nodes visited by the algorithm are designated as "coloured."

ALGORITHM 3.2. Construction of Augmenting Paths.
Procedure AUGMENTPATH (i, PATHFOUND)
(1) Colour i
(2) If a V-node j exists such that edge $(i-j)$ exists
and ASSIGN $(j)=0$ then:
(2a) Set PATHFOUND = TRUE
(2b) Set ASSIGN $(j)=i$
(2c) Return
(3) For every j such that edge $(i-j)$ exists
and j is uncoloured do:
(3a) Colour j
(3b) Set $k=$ ASSIGN (j)
(3c) AUGMENTPATH (k, PATHFOUND)
(3d) If PATHFOUND then
(3d-1) Set ASSIGN $(j)=i$
(3d-2) Return
(4) Return

We prove the following property of the algorithm:

LEMMA 3.3. *If the algorithm returns flag* PATHFOUND *with value* FALSE, *then the coloured subset of equations is minimally structurally singular with respect to* $\{\dot{x}, y\}$.

Proof. If no augmenting path is found and a certain E-node i is coloured, then all V-nodes corresponding to variables occurring in this equation are also coloured, as a result of step (3) of the procedure. Thus the set of coloured nodes corresponds to a subset of the system equations and all variables $\{\dot{x}, y\}$ occurring in this subset. Furthermore, the number of coloured E-nodes exceeds the number of the coloured V-nodes by exactly one. This can be easily deduced by observing that, apart from the very first invocation of AUGMENTPATH, colouring of a V-node j(step (3a)) is immediately followed by colouring E-node k to which j is assigned (steps (3b), (3c) and (1)).

It should now be obvious from the above discussion that the coloured subset of E-nodes corresponds to a subset of the system equations that satisfy criterion (10) and is, therefore, structurally singular.

Furthermore, each E-node lies, by construction, at the end of a path that contains alternating nonmatching and matching edges and is reached through the variable which is assigned to it. Its deletion would leave this variable unassigned and the path leading to it would be a proper augmenting path allowing a reassignment. The existence of an assignment in the remaining subset would then indicate that the number of coloured V-nodes is at least equal to the number of coloured E-nodes and criterion (10) cannot be satisfied. The original coloured subset is, therefore, minimally structurally singular. QED

4. Description of structural algorithm. Using the AUGMENTPATH procedure to locate the minimally structurally singular subsets in a given DAE system, a complete algorithm can be constructed to determine all necessary (in the structural sense) differentiations of system equations.

The original system (1) is rewritten in terms of the vector of all unknowns

$$X=(x, \dot{x}, y)\in G\subseteq R^{2n+m} \tag{11}$$

in the form

$$F(X, t) = 0, \qquad F: G \times R \to R^{n+m}. \tag{12}$$

Relationships between variables and their derivatives with respect to time are expressed through a Variable Association List A defined as:

$$A(j) = k \quad \text{if } X_k = dX_j/dt, \qquad 1 \leqq j, k \leqq 2n+m,$$
$$0 \quad \text{otherwise} \tag{13}$$

and initialized accordingly. Similarly relations between equations are stored in an Equation Association List B defined as:

$$B(i) = l \quad \text{if the } l\text{th equation has been created by differentiating equation } i,$$
$$= 0 \quad \text{otherwise.} \tag{14}$$

List B is initialized to all zeros.

A full description of the algorithm in pseudocode notation follows (Algorithm 4.1). Upon successful termination, N and M will contain the total number of equations and variables, respectively, in the enlarged system containing (1) and all other necessary equation differentials. The two association lists A and B will have also been updated to reflect the relationships between variables and equations in the enlarged system.

ALGORITHM 4.1. Detection of Subsets to be Differentiated.

Given the initial number of equations, $N = n+m$ and variables, $M = 2n+m$
the system bipartite graph $\{f\}$----$\{X\}$,
the variable association list, $A(\,.\,)$

(1) Initialization:
ASSIGN $(j) = 0$, $j = 1(1)M$,
$B(i) = 0$, $i = 1(1)N$.
(2) Set $N' = N$
(3) For $k = 1$ *to* N' do
(3a) Set $i = k$
(3b) Repeat
(3b-1) Delete all V-nodes with $A(\,.\,) \pm 0$ and all their incident edges from the graph
(3b-2) Designate all nodes as "uncoloured"
(3b-3) Set PATHFOUND = FALSE
(3b-4) AUGMENTPATH (i, PATHFOUND)
(3b-5) If PATHFOUND = FALSE then
(i) For every coloured V-node j do
. Set $M = M+1$
. Create new V-node M
. Set $A(j) = M$
(ii) For every coloured E-node l do
. Set $N = N+1$
. Create new E-node N
. Create edges from E-node N to all V-nodes j and $A(j)$ such that edge $(l-j)$ exists.
. Set $B(l) = N$
(iii) For every coloured V-node j set
ASSIGN $(A(j)) = B($ASSIGN $(j))$
(iv) Set $i = B(i)$

(3c) Until PATHFOUND
(4) End

Note that at step (3b-5), PATHFOUND is FALSE, if a minimally structurally singular subset is detected. Step (3b-5)(i) creates additional variable nodes to represent the time-derivatives of the variables in the MSS. Step (3b-5)(ii) creates additional equation nodes resulting from differentiating the equations in the MSS. Note that it is assumed that the equation differentials contain all variables occurring in the original equations *and* the variable time-derivatives. Clearly this is not always correct: if, for instance, a variable occurs linearly in an equation, then it does not occur in the equation time-differential (although its derivative does). However this is not important, since such occurrences are deleted anyway as a result of step (3b-1) when loop (3b) is repeated.

Step (3b-5)(iii) assigns each variable derivative to the differential of the equation to which the variable was originally assigned.

Because of the existence of a Repeat-Until loop (steps (3b)–(3c)), questions arise as to whether the algorithm terminates for a general DAE system. This problem is dealt with next.

DEFINITION. Given a DAE system (1), the corresponding extended system is:

$$\begin{aligned} &f(x, \dot{x}, y, t) = 0, \\ &h_i(x_i, \dot{x}_i) = 0, \qquad i = 1(1)n. \end{aligned} \tag{15}$$

Here each h_i is a new equation relating x_i to $\dot{x}_i$. Since the algorithms given in this paper depend purely on the structure of system (1), it is not necessary to define the exact form of the new equations h; in practice, they could be thought of as difference formulae representing the relationships between variables x_i and $\dot{x}_i$; however, the results presented here are independent of any such assumption.

DEFINITION. The DAE system (1) is said to be structurally *inconsistent* if it can become structurally singular with respect to *all* occurring variables by the addition of the time differentials of a (possible empty) subset of its equations.

The following theorem links the termination of the algorithm to the structural properties of the extended system:

THEOREM 4.2. *Algorithm* 4.1 *terminates if and only if the extended system* (15) *is structurally nonsingular. Furthermore, if* (15) *is structurally singular, then the* DAE *system* (1) *is structurally inconsistent.*

Proof. Part A. Assume that (15) is structurally nonsingular. Here it is shown that differentiation of a subset of equations which is MSS (with respect to variables with $A(\,.\,) = 0$) renders this subset structurally nonsingular, while other subsets which were structurally nonsingular before the differentiation remain so after it. Consequently the algorithm must terminate since there is only a finite number of subsets in the system.

Suppose that the algorithm detects an MSS subset while considering the ith system equation (i.e., procedure AUGMENTPATH returns a FALSE value for flag PATHFOUND). If the coloured E-nodes are denoted by $\{\bar{f}\}$, then the most general form for the MSS subset is:

$$\bar{f}(\bar{x}_1, \bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}, t) = 0 \tag{16}$$

where $\{\bar{x}_1\}$, $\{\bar{x}_2\}$, $\{\bar{x}_3\}$ are disjoint subsets of $\{x\}$, and $\{\bar{y}\}$ is a subset of $\{y\}$.

We denote the bipartite graph representing (16) as a pair of node sets:

$$\{\bar{f}\}\text{----}\{\bar{x}_1, \bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\} \tag{17}$$

or, after the deletion of V-nodes with $A(\,.\,) \neq 0$,

(18) $$\{\bar{f}\}\text{----}\{\dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}.$$

It is the nodes appearing in (18) that form the MSS subset and are coloured by the augmenting path procedure.

Upon differentiation, a new set of E-nodes $\{\dot{\bar{f}}\}$ is created together with a set of new V-nodes $\{\ddot{\bar{x}}_1, \ddot{\bar{x}}_3, \dot{\bar{y}}\}$. Before the next application of the AUGMENTPATH procedure, V-nodes with $A(\,.\,)\neq 0$, namely $(\dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y})$ are deleted. As a result, E-nodes $\{\bar{f}\}$ become totally isolated and are thus effectively removed from further consideration by the algorithm, being replaced by their differentials:

(19) $$\{\dot{\bar{f}}\}\text{----}\{\dot{\bar{x}}_2, \ddot{\bar{x}}_1, \ddot{\bar{x}}_3, \dot{\bar{y}}\}.$$

We note that, apart from a renaming of nodes, (19) has exactly the same structure as

(20) $$\{\bar{f}\}\text{----}\{\bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}.$$

Now the bipartite graph (20) must be structurally nonsingular. For assume that it is not; then it must be MSS since (18) is MSS. Thus

(21) $$C\{\bar{f}\} > C\{\bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}$$

where $C\{\,.\,\}$ is the cardinality of a given set. Then the bipartite graph

(22) $$\{\bar{f}, \bar{h}_1\}\text{----}\{\bar{x}_1, \bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}$$

where $\{\bar{h}_1\}$ are the h-equations corresponding to $\{\bar{x}_1\}$, is also structurally singular. However (22) represents part of the extended system (15) which was assumed to be structurally nonsingular. A contradiction arises demonstrating that (20) (and hence (19)) are structurally nonsingular.

Finally consider some other subset $\{\bar{f}'\}$ which is E-node disjoint from $\{\bar{f}\}$. Suppose that $\{\bar{f}'\}$ was structurally nonsingular before the differentiation of $\{\bar{f}\}$; then it remains so after the differentiation: $\{\bar{f}'\}$ could only become singular if at least one of its equations were assigned to one of the variables $\{\dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}$ deleted following the differentiation of $\{\bar{f}\}$ but in that case, that equation would have been coloured by the AUGMENTPATH procedure and would be part of $\{\bar{f}\}$ itself.

Part B. Assume that (15) is structurally singular. Then, by Theorem 3.1, there exists a subset of (15) which contains more equations than variables. This subset must contain some of the $\{f\}$ equations since no subset consisting entirely of h-type equations can be structurally singular.

The most general form for a subset of (15) is

(23) $$\bar{f}(\bar{x}_1, \bar{x}_2, \ddot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}, t) = 0, \qquad \bar{h}(\bar{x}_4, \dot{\bar{x}}_4) = 0$$

where $\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{y}$ are as for (16) and $\{\bar{x}_4\}$ is another subset of $\{x\}$, not necessarily disjoint from $\{\bar{x}_1, \bar{x}_2, \bar{x}_3\}$.

If $\{\bar{x}_4\}$ is not the same as $\{\bar{x}_1\}$, then one can remove all h-type equations corresponding to the set difference $\{\bar{x}_4\}-\{\bar{x}_1\}$ without affecting the structural singularity of (23). Also one can add the h-type equations corresponding to the set difference $\{\bar{x}_1\}-\{\bar{x}_4\}$ to produce the subset

(24) $$\bar{f}(\bar{x}_1, \bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}, t) = 0, \qquad \bar{h}(\bar{x}_1, \dot{\bar{x}}_1) = 0$$

which is still structurally singular. Thus,

$$C\{\bar{f}, \bar{h}\} > C\{\bar{x}_1, \bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}$$

or

$$C\{\bar{f}\} > C\{\bar{x}_2, \dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}, \tag{25a}$$

$$\geqq C\{\dot{\bar{x}}_1, \dot{\bar{x}}_3, \bar{y}\}. \tag{25b}$$

The algorithm will detect the subset $\{\bar{f}\}$ for differentiation because of (25b); however, the resulting subset will still be structurally singular because of (19), (20) and (25a). Further differentiations will produce subsets with exactly the same structure. The algorithm will keep differentiating ad infinitum.

DAE systems which result in structurally singular extended systems can be shown to be structurally inconsistent. Let the cardinalities of sets $\{\bar{x}_1\}$, $\{\bar{x}_2\}$, $\{\bar{x}_3\}$, $\{\bar{y}\}$, $\{\bar{f}\}$, and $\{\bar{h}\}$ in (24) be n_1, n_2, n_3, n_y, k and n_1, respectively. If

$$k > 2n_1 + n_2 + n_3 + n_y \tag{26}$$

then it will in general be impossible to find values of $x(t)$, $y(t)$ to satisfy equations (16) at any given time $t \geqq t_0$.

Assuming that (26) is not true, the structural singularity of subset (24) implies that

$$k > n_1 + n_2 + n_3 + n_y. \tag{27}$$

Differentiating the f-type equations in (24), one generates k new equations and $(n_1 + n_2 + n_3 + n_y)$ new variables, namely $\ddot{\bar{x}}_1$, $\ddot{\bar{x}}_2$, $\ddot{\bar{x}}_3$, $\dot{\bar{y}}$. Repeating this process of differentiation for a sufficient number of times L with

$$L > \frac{2n_1 + n_2 + n_3 + n_y - k}{k - n_1 - n_2 - n_3 - n_y}, \tag{28}$$

a set of equations can be formed from $\{\bar{f}\}$ and all its time differentials, which contains more equations than variables and is, therefore, structurally inconsistent. QED

It should be noted that the structural nonsingularity of the extended system (15) can easily be checked a priori (and independently of Algorithm 4.1) by determining the maximum transversal of (15) (Duff [1]). Theorem 4.2 then indicates that Algorithm 4.1 should only be applied to DAE systems, the corresponding extended systems of which have a maximum transversal of cardinality $2n + m$.

Inconsistent DAE systems, the equations of which are functionally independent, have no solutions, since it is impossible to find a set of consistent initial conditions. Consequently Theorem 4.2 implies that the algorithm terminates for all well-posed DAE systems. Furthermore, systems with nonsingular extended systems are, in principle, solvable by *implicit* ODE methods since such methods attempt to solve problems by performing the system extension indicated by (15). It is interesting to note that *explicit* ODE methods correspond to system extensions of the form

$$\begin{aligned} &f(x, \dot{x}, y, t) = 0, \\ &h_i(x_i) = 0, \qquad i = 1(1)n. \end{aligned} \tag{29}$$

As a result, only systems for which

$$\text{Rank}\,(f_{\dot{x}} : f_y) = n + m \tag{30}$$

can be solved by explicit methods. Equation (30) represents the subclass of index-1 DAE systems which do not contain structurally singular subsets of the kind located by Algorithm 4.1.

Upon termination of the algorithm one is confronted by a (generally underdetermined) set of N equations in M variables. In order to calculate a consistent set of

initial conditions, one must provide initial values for a subset of $(M-N)$ of the variables and then solve the resulting square system of algebraic equations for the remaining N variables. This is only possible if this system is structurally nonsingular with respect to the variables left unspecified. For a given selection of variables, this can be checked immediately by forming the corresponding bipartite graph and applying procedure AUGMENTPATH (Algorithm 3.2) to each one of the E-nodes. If no augmenting path can be found for a certain E-node, the system is SS and another subset of $(M-N)$ variables must be selected. For large systems this trial-and-error selection process may be often simplified by decomposition techniques (Sargent [8]). The rectangular set of N equations in M unknowns is first reordered to a block-triangular form with all diagonal blocks being square with the exception of the last one. Only variables belonging to this last block may be selected for initialization, since all others are already determined by the solution of the square blocks to which they belong. The required block triangularization can be readily performed by an efficient graph-theoretical algorithm due to Tarjan [9].

Very recently, Duff and Gear [2] developed an algorithm for determining whether the index of a class of DAE systems exceeds a value of two, a problem which is closely related to the one examined here. Their algorithm is also based on structural concepts, and can exhibit nonpolynomial computational complexity for certain problems. However, we believe that Algorithm 4.1 of this paper is of greater applicability since it does not place any restriction on the form of the DAE system considered and it can indeed analyze systems of any index. Furthermore, its complexity is only polynomial in the size of the problem, and comparable to the complexity of algorithms for the determination of structural rank. This can be seen from the fact that the algorithm only considers each equation (original or derived) exactly once, attempting to construct an augmenting path emanating from it (step (3b-4)). Since each invocation of routine AUGMENTPATH can at most examine each edge once, the complexity of the algorithm does not exceed $N.t$, where N and t are, respectively, the number of equations and edges (nonzeros) in the *final* system.

5. Some examples of the application of the algorithm. In this section Algorithm 4.1 is applied to two DAE systems corresponding to nonsingular extended systems. A third example, corresponding to a structurally singular extended system is also presented to demonstrate both the fact that the algorithm does not terminate and the structural inconsistency of the system.

Example 3. Fixed-Length Pendulum. The equations of motion of a pendulum of fixed length L in Cartesian coordinates are (Lötstedt and Petzold [6]):

$$\begin{aligned} \ddot{x} &= Tx, \\ \ddot{y} &= Ty - g, \\ 0 &= x^2 + y^2 - L^2 \end{aligned} \tag{31}$$

where T is the (unknown) tension in the pendulum bar and g is the gravity constant. Through the definition of auxiliary variables w and z, (31) can be transformed to the first-order DAE system:

$$\begin{aligned} \dot{x} &= w, \\ \dot{y} &= z, \\ \dot{w} &= Tx, \\ \dot{z} &= Ty - g, \\ 0 &= x^2 + y^2 - L^2. \end{aligned} \tag{32}$$

In this example, initially $M = 9$ and $N = 5$; vector X is

$$X = (x, y, w, z, \dot{x}, \dot{y}, \dot{w}, \dot{z}, T)$$

with the variable association list

$$A = (5, 6, 7, 8, 0, 0, 0, 0, 0).$$

The bipartite graph representing (32) is shown in Fig. 1(a). Its form after deletion of V-nodes with $A(\,.\,) \neq 0$ (i.e. x, y, w, and z) is shown in Fig. 1(b).

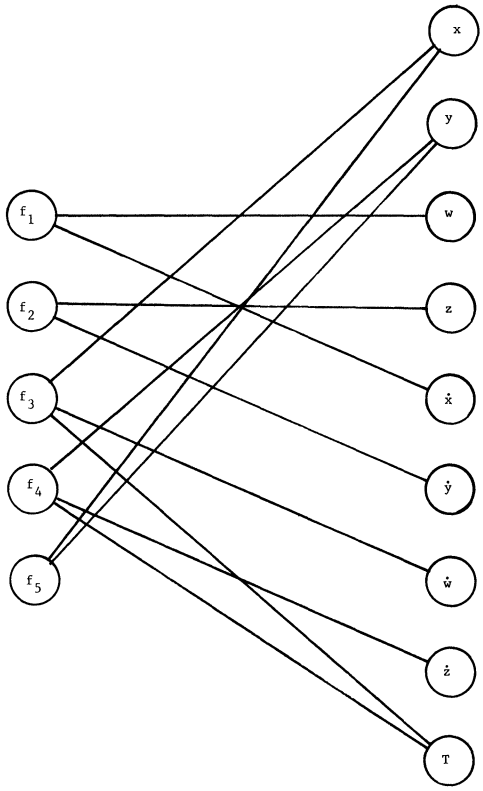

FIG. 1(a). *Fixed-length pendulum.*

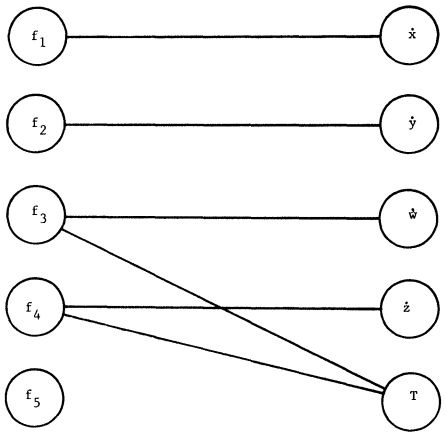

FIG. 1(b)

When Algorithm 4.1 is applied to the graph in Fig. 1(b), procedure AUGMENTPATH returns a TRUE value for the flag PATHFOUND for the first four equations ($k = 1, 2, 3$, and 4). Thus no differentiation is necessary so far. The (trivial) augmenting paths determined by the procedure are

$$(f_1, \dot{x}), (f_2, \dot{y}), (f_3, \dot{w}), (f_4, \dot{z})$$

corresponding to vector ASSIGN becoming

$$\text{ASSIGN} = (0, 0, 0, 0, 1, 2, 3, 4, 0).$$

However the fifth invocation of AUGMENTPATH returns a value of FALSE for PATHFOUND. No V-nodes are coloured, but E-node f_5 is; f_5 is therefore differentiated (step (3b-5)(ii)). The number of equations, N, is increased to six and a new E-node is created. The equation association list becomes

$$B = (0, 0, 0, 0, 6, 0).$$

Procedure AUGMENTPATH is now applied again to the bipartite graph shown in Fig. 1(c). The depth-first search procedure of Algorithm 3.2 constructs paths ($f_6 -- \dot{x} -- f_1$) and ($f_6 -- \dot{y} -- f_2$) without being able to complete an augmenting path. The coloured V-nodes $\dot{x}$ and $\dot{y}$ are differentiated (step (3b-5)(i) of Algorithm 4.1) to produce the new V-nodes $\ddot{x}$ and $\ddot{y}$; M is increased to 11 and list A becomes

$$A = (5, 6, 7, 8, 10, 11, 0, 0, 0, 0, 0)$$

with variables assigned according to

$$\text{ASSIGN} = (0, 0, 0, 0, 1, 2, 3, 4, 0, 7, 8).$$

Also E-nodes f_1, f_2 and f_6 are differentiated to produce new E-nodes f_7, f_8, and f_9. List B becomes

$$B = (7, 8, 0, 0, 6, 9, 0, 0, 0).$$

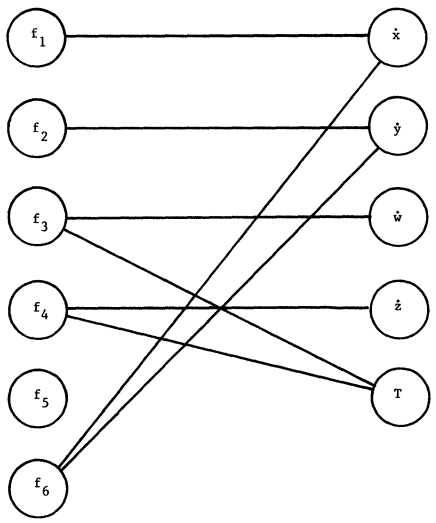

FIG. 1(c)

After deletion of nodes with $A(\,.\,)\,0$, the bipartite graph is shown in Fig. 1(d). Applying the AUGMENTPATH procedure to E-node f_9 finally succeeds in producing the augmenting path

$$(f_9 -- \ddot{x} -- f_7 -- \dot{w} -- f_3 -- T).$$

After reassignment, vector ASSIGN is

$$(0, 0, 0, 0, 1, 2, 7, 4, 3, 9, 8).$$

No more differentiations are required. The final system dimensions are $M = 11$ and $N = 9$; thus two variables out of the set $\{x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, w, \dot{w}, z, \dot{z}, T\}$ can be given arbitrary initial values, subject to the constraint that the system of 9 equations must be nonsingular with respect to the remaining 9 variables (see discussion at the end of § 4).

Example 4. Exothermic Reactor Model. The DAE system considered in this example is derived from the equations describing a chemical reactor. A simple first-order isomerization reaction takes place and the heat generated is removed from the system through an external cooling circuit. The relevant equations are:

$$\begin{aligned} \dot{C} &= K_1(C_0 - C) - R, \\ \dot{T} &= K_1(T_0 - T) + K_2 R - K_3(T - T_c), \\ 0 &= R - K_3 \exp(-K_4/T) C. \end{aligned} \tag{33a}$$

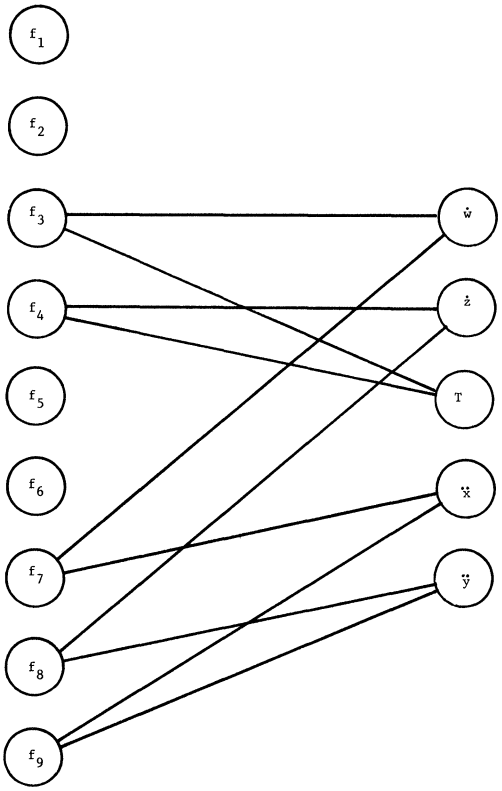

FIG. 1(d)

Here C_0 and T_0 are the feed reactant concentration and feed temperature (assumed known); C and T are the corresponding quantities in the product. R is the reaction rate (per unit volume). T_c is the temperature of the cooling medium, which can be varied. K_1, K_2, K_3 and K_4 are constants.

If T_c is known, the initialization of system (33a) is straightforward, requiring that the variables $\{C, \dot{C}, T, \dot{T}, R\}$ only satisfy the equations in (33a). However the related problem of determining the variation of the cooling medium temperature necessary to achieve a given, twice-differentiable variation with time, $u(t)$ of the product concentration is more interesting. In order to investigate this problem, the specification is added to (33a) as an additional equation:

$$0 = C(t) - u(t). \tag{33b}$$

The bipartite graph describing this system is shown in Fig. 2(a). The system dimensions are $M = 6$ and $N = 4$, with the variable vector $X = (C, T, \dot{C}, \dot{T}, R, T_c)$ and the variable association list $A = (3, 4, 0, 0, 0, 0)$.

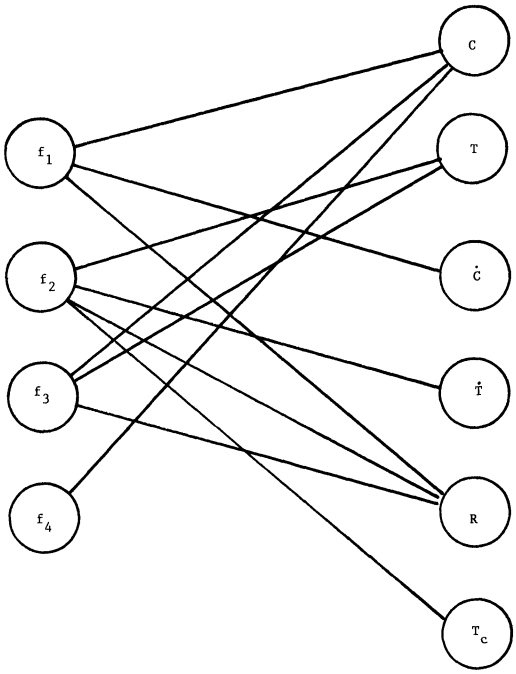

FIG. 2(a). *Exothermic reactor model.*

After deletion of V-nodes (C, T), the bipartite graph becomes as shown in Fig. 2(b). Application of procedure AUGMENTPATH to the first three E-nodes succeeds in finding augmenting paths, and vector ASSIGN becomes

$$\text{ASSIGN} = (0, 0, 1, 2, 3, 0).$$

However application of AUGMENTPATH to the isolated E-node 4 fails to determine a suitable path. The node is coloured, and, therefore, differentiated at step (3b-5)(ii) of Algorithm 4.1. List A and vector ASSIGN remain unchanged, but list B changes to

$$B = (0, 0, 0, 5, 0).$$

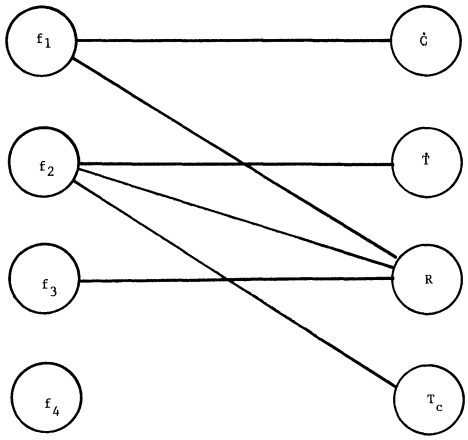

FIG. 2(b)

The new bipartite graph is shown in Fig. 2(c). Applying AUGMENTPATH to E-node 5, traces the path

$$(f_5 -- \dot{C} -- f_1 -- R -- f_3).$$

Again no augmenting path can be constructed. New V-nodes $\ddot{C}$ and $\dot{R}$ are formed (step (3b-5)(ii)) and linked to the new E-nodes f_6, f_7, and f_8 produced by differentiating f_1, f_3, and f_5 (step (3b-5)(iii)). After step (3b-5)(iii), the current assignment becomes

$$\text{ASSIGN} = (0, 0, 1, 2, 3, 0, 6, 7).$$

Lists A and B become

$$A = (3, 4, 7, 0, 8, 0, 0, 0),$$

$$B = (6, 0, 7, 5, 8, 0, 0, 0).$$

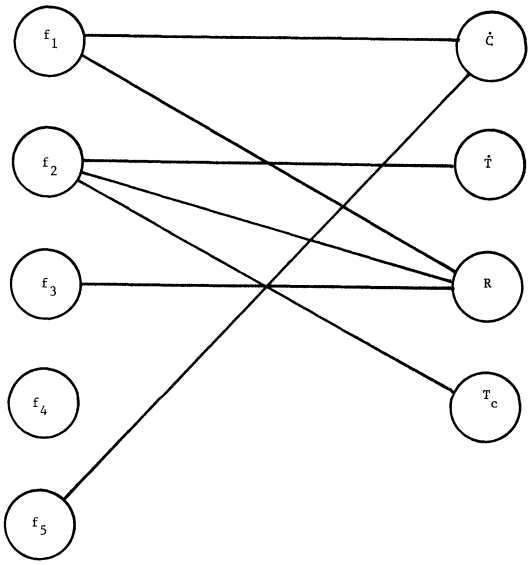

FIG. 2(c)

After deletion of V-nodes $(\dot{C}, R)$, AUGMENTPATH is applied to node f_8 in the bipartite graph of Fig. 2(d). An augmenting path is found, namely

$$(f_8 -- \ddot{C} -- f_6 -- \dot{R} -- f_7 -- \dot{T} -- f_2 -- T_c).$$

Reassignment along this path results in

$$\text{ASSIGN} = (0, 0, 1, 7, 3, 2, 8, 6).$$

The final system dimensions are $M = 8$ and $N = 8$. Thus no variables can be given an arbitrary initial value!

Example 5. Uncontrollable DAE system. The simple DAE system considered here is

$$\begin{aligned} 0 &= f_1(x, u_1, u_2), \\ 0 &= f_2(x, \dot{x}, y_1), \\ 0 &= f_3(x, y_2). \end{aligned} \tag{34}$$

In a control context, y_1 and y_2 can be thought of as desired system outputs; u_1 and u_2 are (unknown) system inputs which are required to yield outputs $y_1(t)$ and $y_2(t)$. Variable $x(t)$ is also an unknown.

Here the vector of unknowns is $X = (x, \dot{x}, u_1, u_2)$. The corresponding bipartite graph before and after deletion of V-node x is shown in Figs. 3(a) and 3(b). Application of AUGMENTPATH to E-nodes f_1 and f_2 results in the assignment

$$\text{ASSIGN} = (0, 2, 1, 0).$$

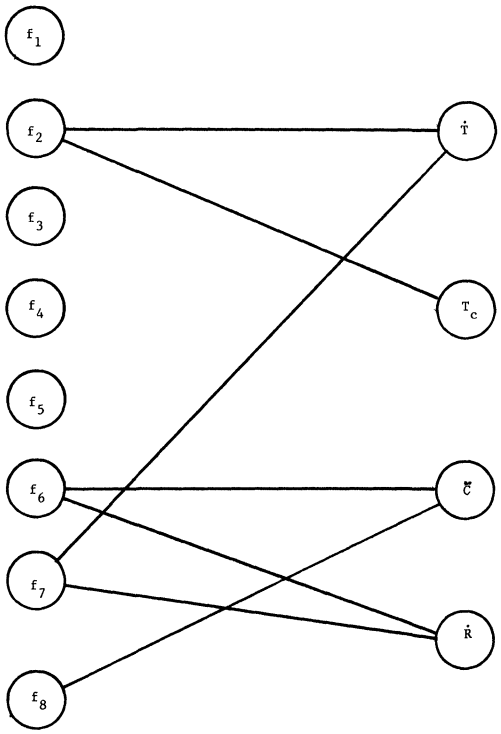

FIG. 2(d)

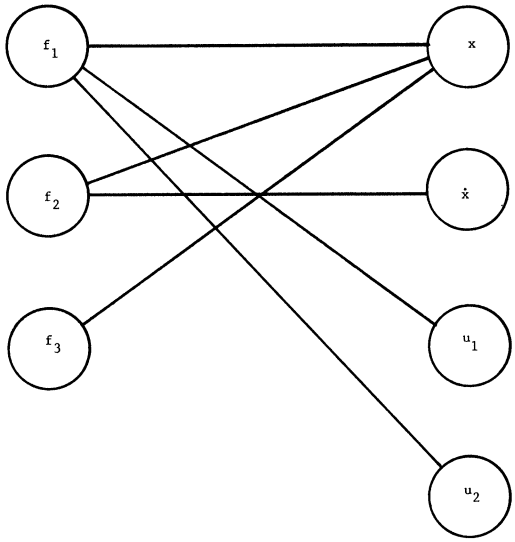

FIG. 3(a). *Uncontrollable DAE system.*

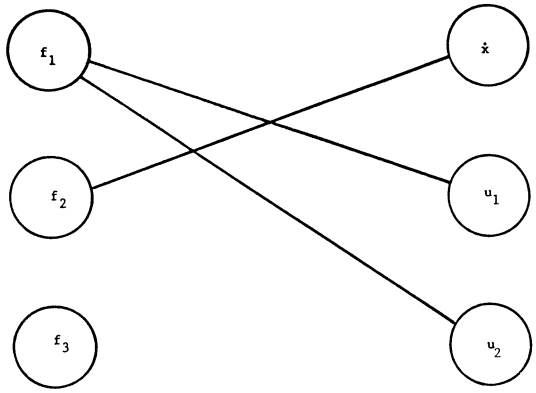

FIG. 3(b)

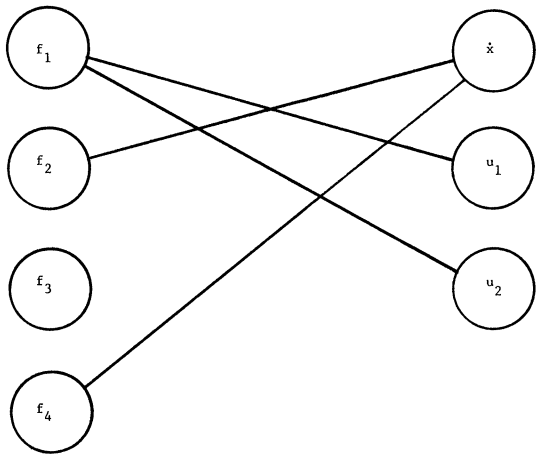

FIG. 3(c)

E-node f_3 is isolated; it is therefore coloured and differentiated to yield the new E-node f_4 (see Fig. 3(c)). Applying AUGMENTPATH to f_4 again fails to produce an augmenting path after colouring E-nodes f_2 and f_4 and V-node $\dot{x}$.

Differentiation of equations f_2 and f_4, and deletion of V-node $\dot{x}$ produces the bipartite graph of Fig. 3(d). However the subgraph

$$\{f_5, f_6\} -- \{\ddot{x}\}$$

in Fig. 3(d) is essentially the same as subgraph

$$\{f_2, f_4\} -- \{\dot{x}\}$$

in Fig. 3(c). As a result the algorithm will keep colouring and differentiating the same subgraph ad infinitum. This is not surprising since the extended system corresponding to (34) is structurally singular, containing the structurally singular subset

$$\begin{aligned} f_2(x, \dot{x}, y_1) &= 0, \\ f_3(x, y_2) &= 0, \\ h(x, \dot{x}) &= 0. \end{aligned} \tag{35}$$

Furthermore, system (34) is structurally inconsistent: equations f_2 and f_3 and their differentials with respect to time form a system of four equations in the three variables x, $\dot{x}$ and $\ddot{x}$. If f_2 and f_3 are independent, then no solution exists and the required control objectives $y_1(t)$ and $y_2(t)$ are unachievable with the available controls $u_1(t)$ and $u_2(t)$.

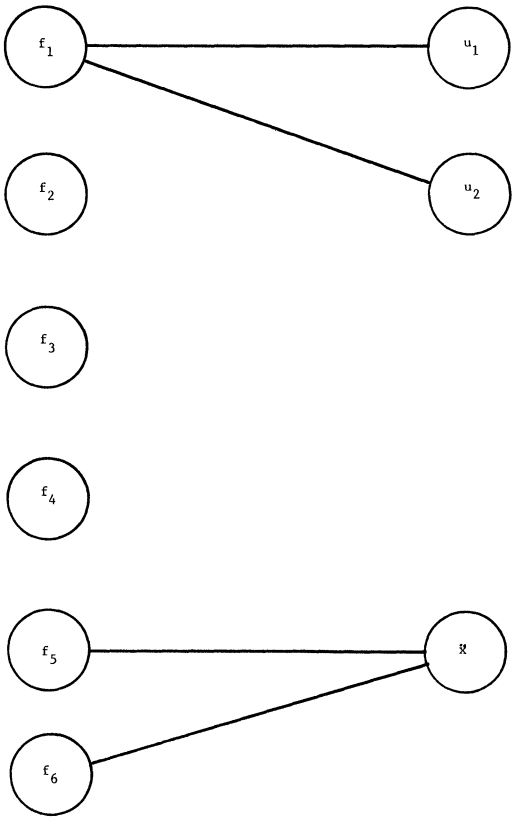

FIG. 3(d)

6. Some concluding remarks. The algorithm presented in this paper identifies the minimal subset of equations, differentiation of which is necessary for consistent initialization. This, coupled with its low computational complexity, should make it especially useful for the analysis and solution of large DAE systems.

The algorithm is entirely graph-theoretical and requires no arithmetic operations; it does not, therefore, suffer from the problems associated with numerical algorithms such as rounding errors. However the price paid for this advantage is that an equation subset that ought to be differentiated may escape detection, due to the sufficient but not necessary nature of criterion (10).

For instance, consider the linear system

$$\begin{aligned} \dot{x}_1 &= x_1 + 2y_1 + 3y_2, \\ 0 &= x_1 + y_1 + y_2 + 1, \\ 0 &= 2x_1 + y_1 + y_2. \end{aligned} \tag{36}$$

If subset $\bar{f}$ of the system equations is taken to consist of the last two equations ($k=2$), then with $\bar{z}=(y_1, y_2)^T$ we have

$$\frac{\partial \bar{f}}{\partial \bar{z}} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

i.e., rank $(\partial \bar{f}/\partial \bar{z}) = 1 < 2$ (see (9)).

Clearly this subset should be differentiated; however, the structural criterion (10) is not satisfied since the cardinality, l of $\bar{z}$ is also 2. In such cases the algorithm terminates without detecting all the equation subsets which ought to be differentiated.

Finally the algorithm is not only useful for the initialization of DAE systems but also for transforming them to a form amenable to solution by existing codes with the minimum possible distortion of the structure of the original system. Upon termination of the algorithm one can discard all but the highest order differential of each equation; these, together with the relationships between variables expressed by the Variable Association List constructed by the algorithm, form a DAE system for which the matrix $[f_{\dot{x}} f_y]$ is structurally nonsingular. In general, this system can then be solved by ODE codes (Gear and Petzold [3]).

REFERENCES

[1] I. S. DUFF, *On algorithms for obtaining a maximum transversal*, ACM Trans. Math. Software, 7 (1981), pp. 315-330.

[2] I. S. DUFF AND C. W. GEAR, *Computing the structural index*, Technical Memorandum 50, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1985.

[3] C. W. GEAR AND L. R. PETZOLD, ODE *methods for the solution of differential-algebraic systems*, SIAM J. Numer. Anal., 21 (1984), pp. 716-728.

[4] P. HALL, *On representatives of subsets*, J. London Math. Soc., 10 (1935), pp. 26-30.

[5] F. HARARY, *Graph Theory*, Addison-Wesley, Reading, MA, 1969.

[6] P. LÖTSTEDT AND L. R. PETZOLD, *Numerical solution of nonlinear differential equations with algebraic constraints*, Report SAND83-8877, Sandia National Laboratories, Albuquerque, NM, 1983.

[7] L. R. PETZOLD, *Differential algebraic equations are not* ODE's, this Journal, 3 (1982), pp. 367-384.

[8] R. W. H. SARGENT, *The decomposition of systems of procedures and algebraic equations*, in Numerical Analysis—Proceedings, Dundee, G. A. Watson, 1977, Lecture Notes in Mathematics 630, Springer-Verlag, Berlin, New York, Heidelberg, 1978.

[9] R. E. TARJAN, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146-160.