Carnegie Mellon University
Research Showcase @ CMU

Department of Chemical Engineering | Carnegie Institute of Technology

1979

Diagnosis procedures from fault trees

Thomas L. Teague
Carnegie Mellon University

Gary J. Powers

Follow this and additional works at: http://repository.cmu.edu/cheme

Published In
.

This Technical Report is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase @ CMU. It has been accepted for inclusion in Department of Chemical Engineering by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

DIAGNOSIS PROCEDURES FROM FAULT TREES

by

Thomas L. Teague and Gary J. Powers

DRC-06-6-79

May 1979

* Dept. of Chemical Engineering
Carnegie-Melion University
Pittsburgh, PA 15213

Accepted for publication in IEEE Transactions on Reliability
to be published in late 1979

SUMMARY AND CONCLUSIONS

A method is presented demonstrating the use of Fault Tree Analysis to produce diagnostic procedures for chemical processing systems. Fault trees are generated for important process variables and alarms. Diagnostic test procedures are subsequently constructed, effectively using the cause and effect relationships implied in the cause and effect digraph models and fault trees. A priori'estimates of failure rates are combined with real time data to determine test ordering, thereby producing efficient procedures. System faults can be identified in detail, while maintaining flexibility in the depth of detail. Diagnostic procedures can be developed for multiple events which occur due to a common cause. Inconsistencies which sometimes arise from unfamiliar patterns of multiple events are addressed. Algorithms are presented and demonstrated manually on a simple chemical system. The implementation of these algorithms would provide a feasible means to generate the diagnostic procedures automatically resulting in greater accuracy and completeness than can be obtained manually. The resulting diagnostic procedures permit operations personnel to rapidly and accurately diagnose process failures and assess the relative hazard state of the process.

UNIVERSITY LIBRARIES
CA^SE-MELLGN UNIVERSITY
PITTSBURGH. PENNSYLVANIA 15213

INTRODUCTION

In an operating chemical plant there are many process variables that require monitoring to ensure that product quality and process reliability are maintained. At the same time the plant must be operated safely to reduce the personal and economic risks associated with events such as fires, explosions, and release of toxic chemicals. When certain process upsets or failures occur these objectives may occasionally be in direct conflict and it is left to the operator and/or plant engineer to decide upon the best course of action in a particular situation. Often the time required to make this decision may be critical.

As chemical plants have increased in complexity, they have also increased in the degree and sophistication of the instrumentation used. Crucial components of a system are controlled automatically to stay within an operating range while emergency shutdown systems are provided when certain critical process variables exceed a safe range. However, there remains a substantial grey area when a process is approaching an unsafe state through a series of process failures. At this point the process operator must take appropriate corrective actions to avoid the hazard state. The reliable and rapid detection and correction of these

process failures is important to the maintenance of process integrity.

During a process upset, the amount of information to be analyzed may be overwhelming or indicative of two or more seemingly conflicting process states. The operator at this point needs assistance in diagnosing the causes of the process upset and in evaluating the relative hazard state of the process.

Presently, to diagnose a process upset, the operator relies on his own experience and intuition, simple mental cause-consequence models of the process, and prewritten operating instructions. In cases where the operator is unfamiliar with the process and/or has difficulty in assessing the process state, a diagnostic aid would serve a valuable purpose.

Once a successful diagnosis has been made there is the problem of deciding the appropriate actions to take to counteract the failures. To aid the operator in this decision some estimate of the consequences of continuing plant operation at full or reduced levels need to be weighed against the consequences of complete plant shutdown. We shall address primarily the issue of a successful diagnosis and show how the information gained in the diagnosis procedure assists in making these operating decisions.

PREVIOUS DEVELOPMENTS

Formal diagnosis of process failures has its genesis in the area of alarm analysis. Pioneering research efforts in alarm analysis applied to nuclear plants were done by Kay [3]i Patterson [8], and Melbourne [15]. Recent increased usage of process computers for monitoring and control have stimulated the interest in at least partially automating alarm analysis and failure diagnosis.

The alarm analysis technique was developed as an aid to the operator -- particularly in nuclear systems. Nuclear systems usually have hundreds of process alarms. Often many of these alarms occur simultaneously, hampering the operator's task of fault diagnosis from the standpoint of information overload. Alarm analysis interprets the sequence and types of alarms to discover a smaller set of "primary" alarms. This reduces the number of alarms which the operator must interpret and therefore assists the operator in fault diagnosis.

Failure diagnosis generally resolves the faults in more detail. Instead of identifying primal alarms, components and system failures are identified. These failures are the specific causes of the alarm or event pattern that is observed. Failure diagnosis provides the additional

advantage of explaining "inconsistent" alarm patterns.

Inconsistent alarm patterns arise when a certain pattern which "normally" occurs only partially occurs. For example, the operator notices that the pump indicator comes on when the pump is turned on, but the flowmeter in the outlet line indicates no flow. Which is the operator to* believe? Failure modes which inactivate causative relationships in the process can lead to a confusing situation and potentially disastrous results. It is crucial that diagnosis procedures be able to resolve such conflicting information since it is this situation which leads to higher probabilities of no action or slow action on the part of the operator to correct the process disturbance.

Failure diagnosis usually takes the form of a diagnosis * tree (troubleshooting chart) or checklists. The automotive and electronics industries currently use diagnosis trees. These charts are produced manually and are a costly, but indispensible, aid to the automotive and electronics technician. For chemical plants diagnosis trees are not widely used although their usefulness is acknowledged.

Diagnostic procedures such as those described above represent in a compact form the following information.

1. Engineering knowledge of cause and effect relationships existing in the process.

2. A priori estimates of failure rates of causative failure events.

3. The acquisition of real time data which determines the sequence of tests necessary to isolate the faults.

There are two major diagnostic goals for engineering systems. One approach is to isolate faulty system components while the system is in a non-operational mode. The faulty component(s) are isolated by a series of alterations of the system configuration, testing the response of the system to these changes. Examples of this "off-line" approach are most maintenance procedures and mechanical system and electrical system diagnosis procedures.

The second approach -- the approach required for chemical and other processing systems -- is isolation of local causes while the system is "on-line". The diagnostic goal of such systems is not only fault isolation but also prevention of the system from moving to an undesired state.

There are several criteria that need to be met to produce diagnosis procedures of acceptable quality. They are:

1. A rapid and reliable method of generating and updating diagnosis procedures at reasonable cost.

2« A method which would encompass most engineering systems of interest, e.g., continuous or sequential, chemical, nuclear, mechanical or electrical systems.

3« Efficient ordering of tests so that the number of diagnostic tests is adequately low.

4. Efficient use of a priori estimates of failure rates together with real time data acquisition.

5. Identify the common cause(s) of multiple events which occur simultaneously.

6. Resolve inconsistencies in real time input data.

Table 1 summarizes some of the recent research efforts In alarm analysis and fault diagnosis. The work presented in this paper is given also for comparison.

Andow and Lees [1] have demonstrated alarm analysis for chemical and nuclear systems. The work which they have described concerns a method for automatically generating alarm trees from process unit models. These alarm trees are suitable for subsequent on-line alarm analysis on a process computer. The work addresses one of the major criticisms of alarm analysis -- that of manual generation of the alarm trees which serve as the basis for on-line alarm analysis. The advantages of automation -- improved accuracy, ease of updating, and completeness are introduced into the alarm analysis method. One drawback is the non-inclusion of failure modes which may alter the normal relation-ships implied in the alarm trees. This results in insufficient attention to the issue of inconsistent alarm patterns.

Grumbach and Pfeiff [2] have outlined conceptually a software system which is designed to serve as an interactive operation aid. The diagnosis or disturbance analysis consists of identifying "event chains" which lead to intermediate and final events. These event chains are stored in the form of "event matrices" which are analyzed using real time operating data. In the diagnosis effort both

primal causes (alarms) and probable final events are predicted. This is, in effect, an application of Failure Modes and Effects Analysis (FMEA) discussed elsewhere ([7]$_9$[12]). Since event chains are determined manually there may be some question as to the completeness and accuracy of the analysis* Provisions are made, however, for the operator to update the event matrices to reflect operating experience. Therefore the system can "learn" as experience is gained.

Pieper and Pinkus [9] developed a computer program for the U.S. Air Force for electrical systems and demonstrated the feasibility of automatic generation of troubleshooting charts for electrical systems. Their approach consisted of generating tests for upstream components and all non-redundant combinations of upstream switch positions. These tests are subsequently evaluated using an information gain per unit cost ratio. Test sequencing is arranged so that as nearly as possible half of the system is isolated. Major problems include the inability to handle the combinatorial problems as the number of upstream switches increased to over ten and failure to sufficently resolve feedback loops.

Lambert and Yadigaroglu [4] have used Fault Tree Analysis to generate diagnosis checklists. These checklists are derived by ranking primal events by probabilistic importance and checking each primal event for occurrence* The sequence of the checklist is updated to reflect the knowledge gained of the failures that have occurred. Diagnosis is complete when a minimal cutset of the fault tree has been diagnosed as having occurred. They used their method to identify minimal cutsets with a maximum of two components which caused a single top event. The issues of multiple top events and inconsistent real time data were not discussed.

Most of the recent research has focused on obtaining automatic methods of analysis and diagnosis tree generation. Particularly important are the issues of a good modeling basis i completenessi multiple occurrences due to common causes, and inconsistencies in real time input data. We believe that Fault Tree Analysis can be used to address each of these issues.

APPLICATION OF FAULT TREE ANALYSIS

The technique of Fault Tree Analysis (FTA) provides information relevant to how chemical processes fail. The development of a set of fault trees for a chemical process

has been shown to be a powerful tool for identifying critical components of the system which contribute to the occurrence of hazardous events. Implicit in FTA is the engineering knowledge of cause and effect and a priori probability estimates of failure rates. Efficiently combining the information contained in FTA with real time operating data is necessary for an effective diagnosis procedure.

The FTA technique may be applied to any process variable. In safety analysis hazardous events are usually considered the top event in the fault tree. For diagnosis the top event may be a process alarm which has enabled or an important process variable which is deviating from the norm. The FTA technique asks the question, "How did the top event occur?". This is precisely the diagnosis question and hence may be applied to any top event - hazardous or non-hazardous.

Recent developments by Powers and Lapp ([5],[6],[10]) in the automation of Fault Tree Synthesis enhance the attractiveness of using FTA to generate diagnostic procedures. In a given process there are likely to be a substantial number of alarms and important process variables for which diagnosis trees would be useful. Manual generation of the relatively large number of fault trees necessary for

a complete set of diagnosis procedures is likely to be extremely time consuming and error prone. Therefore, the feasibility of generating diagnosis procedures from FTA is dependent to a large degree on the ability to generate the fault trees automatically.

Computer-aided fault tree synthesis ([6],[10]) offers several ideas which are useful for addressing the diagnosis issues that were cited previously* Modular cause and effect models of process units can be assembled into a system cause and effect model* Such a model includes all known cause and effect behavior* As more knowledge is gained the model can be updated to reflect this knowledge. Properties of certain variables in the process model can be used to identify common causes of multiple events. Particular failure modes can also be used to resolve inconsistent real time data. Fault tree synthesis accounts for failure modes and system interactions. Finally, probability calculations using both a priori estimates of failure rates and real time failure data provide a quantitative base which can be used for ordering the test sequence in the diagnosis.

Powers and Tomkins [11] and Powers and Lapp [6],[10] have described a method for modeling the cause and effect behavior of a chemical system. These models take the form of a directed graph or digraph. Each node in the digraph

represents a process variable or failure event* The directed edges of the graph indicate a causal relationship between the variables by means of a "gain[11] on the edge* For example,

$$V1 \xrightarrow{+1} V2$$

means that a positive deviation in V1 causes a positive-deviation in V2 or that a negative deviation in V1 causes a negative deviation in V2. In a chemical system digraph there can be more than one edge between variables representing different behavior when certain specified conditions'are met* These conditions are usually but not always failure modes which change the normal relationship between VI and ¥2*

The information contained in the system digraph that is particularly important to diagnostic procedures is the identification of loops and common variables in the process* These are indicative of special interactions in the process which must be accounted for in FTA as well as in diagnosis* The loop and common variable analysis of the digraph helps to predict which event patterns occur together due to a common cause* This analysis also aids in resolving inconsistent real time input data due to "inactivation" failure modes (zero gain edges)*

The digraph models are constructed for a chemical system by first choosing a set of top events relevant to diagnosis. These events consist of alarms and important process variables. The selection of these top events reflects the engineering judgement of which variables and alarms are important for rapid diagnosis. Digraphs are constructed from these top events using cause and effect models for process units.

As an example of the application of this modeling technique consider the following system. A mixer in series with a heat exchanger is designed to produce a constant flow while maintaining a steady outlet temperature. Figure 1 contains the flowsheet and numbering scheme. Note that stream 7 normally has a temperature and flow greater than that of stream 1. The exit temperature is controlled by regulating the coolant flow to the exchanger.

For this system, the following top events have been defined.

1. Temperature High Alarm
2. Flow High Alarm
3. Temperature in stream 14 too high
4. Flow in stream 14 too high

A digraph was constructed for each top event. Figure 2

contains the digraph which was obtained by superimposing each of the individual digraphs onto a single digraph.

At this point the fault trees are generated by applying the Lapp-Powers Fault Tree Synthesis algorithm ([6],[10]) to the system digraph* The fault tree for top event 3 - Temperature in stream 14 too high - is found in Figure 3* Probability data were assigned to the primal events in the form of failure rates and detection times. These data were then used to calculate the ranked minimal cutset form of the trees* Probabilities were also calculated for each gate in the fault tree* The top minimal cutsets for this tree are presented in Figure 4* A complete set of fault trees, cutsets and diagnosis procedures are given in Teague [14]*

DIAGNOSTIC PROCEDURES

As mentioned previously there are two major types of diagnostic procedures for engineering systems* Each of these have different diagnostic goals. One approach is to isolate faulty components of a system in a off-line or non-operating mode* The other approach is to isolate the local causes of observed system behavior while the system is in an on-line or operating mode*

For electrical or mechanical systems, where the system is not in an operating mode, the diagnostic goal is to isolate or resolve those components of the system that are faulty* This may be accomplished by changing the system configuration (setting switches, etc-) so as to sucessively reduce the search space until one or two components remain which must be faulty or by devising test patterns which resolve the faulty component* This is the type of approach described previously [9]-

The other major diagnostic approach is one normally encountered in chemical systems* The diagnostic goal for chemical systems is usually hazard prevention and/or reliability considerations in an operating system* The goal in chemical systems is to acquire enough information to make a decision regarding future system status* At wha-t level of operation should the plant be operated? Alternative levels might be:

1* Continued full operation
2. "Hold[11] certain key units at reduced operation
3* Partial plant shutdown
4* Full plant shutdown

The types of system reconfigurations in chemical systems analogous to those used in electrical or static mechanical systems are not usually feasible* Indeed they may cause a

more serious process upset than the original problem* The diagnosis of operating chemical systems must therefore be developed in another way*

The diagnostic procedure in an operating mode must necessarily be an efficient and rapid data gathering procedure- The causative relationships (between observable variables and events) and the relative estimated probability of occurrence of these relationships can serve as a key to direct the diagnosis procedure* Flexibility in the depth of resolution of the diagnosis is a desirable feature for meeting time constraints*

The FTA approach is suitable for achieving the diagnostic goals described above* By utilizing the probability calculations and the cause and effect relationships implied in the fault trees, diagnostic procedures can be deduced in a relatively straightforward manner* All of the basic data required -- system loops, event and gate probabilities! and the local cause and effect relationships -- are present or implicit in the fault tree and the digraph models from which the fault trees are derived*

DIAGNOSTIC PROCEDURES FOR A SINGLE TOP EVENT

Two approaches have been developed for generating diagnostic procedures from Fault Tree Analysis* The first approach is more suited to an off-line mode, full resolution diagnostic goal* This approach operates on the minimal cutset form of the fault tree* The cutsets are arranged in a list with the most probable first* This list can be ranked based on a priori estimates ?nd/or be reranked periodically by a process computer which can measure in real time which of the events are currently true* Primal events occurring in the most probable cutset are verified as to whether they have occurred* This information is used to rerank the ordering of cutsets so that all cutsets containing true events (probability ~ 1*0) are shifted towards the top of the list* Similarly false events (probability ~ 0*0) are shifted to the bottom of the list* Note that the probabilities cannot be set exactly equal to 1*0 or 0*0* This is due to errors which may be present while executing the diagnostic tests* If the diagnosis procedure is explicitly stated in the operating instructions, then the fault tree could be modified to include this procedure and specifically account for these errors* Diagnosis is complete when all events in a cutset are proven to be true* This approach is similar to that presented by Lambert and

Yadigaraglu [4].

The second approach is more suited to the operation mode- In this method the fault tree is used directly- A search of the gates and events which are inputs to the current gate of consideration is made- At OR gates each input to the gate is tested until a true input is found- When a true gate is discovered, it becomes the new current gate- Therefore , a depthwise search is made on true gates* At AND gates all the inputs are assumed to be true to satisfy the implied logic* At this point a decision is made as to whether to continue the search. Since the causative chain of events (fault propagation path) is clear at any point, flexibility in depth of resolution can be obtained* As mentioned above the test ordering is determined by estimated probabilities of failures and/or real time measurements.

To more fully illustrate these concepts, algorithms for each approach are presented. These algorithms were applied to the mixer/heat exchanger example to illustrate the end result.

Note that the use of the words resolve, resolved or resolution as used in the text or figures refer to whether the diagnosis is complete or to what level of detail is reached. This should not be confused with resolving a

Boolean expression.

Cutset Diagnosis Algorithm

1« Rank minimal cutsets by their importance (probability)* (May be a priori or current real time data*)

2* Rank each element in the cutset by some criterion* (This may be importance, probability, ease of testing, etc.)

3* If no more cutsets remain and no cutset has been verified, stop unresolved* If all events in a cutset have been verified, stop with the diagnosis complete* Otherwise, continue*

4« Consider the most important event remaining and verify by a true/false test the occurrence of that event*

5* If true, then rerank all cutsets by recalculating those cutsets which contain the true event, setting that event[1]s probability approximately equal to 1*0*

6. If false, then eliminate all cutsets from consideration which contain the false event as its probability is now approximately equal to 0.0.

7* Repeat from step (3)*

As an example of the cutset diagnosis algorithm, consider the top event "Temperature in stream J4 too high" for the mixer/heat exchanger example (see Figure 4). The resulting diagnosis procedure is presented in Figure 5* Note that, in general, the test ordering is first one component cutsets, then two component, then three component, etc. This is primarily due to the probability ordering. Consider as a' typical combination, cutset number 25 — M10 (-1) AND TRC on manual. The sequence of diagnostic tests is shown with a dashed line in Figure 5« The general characteristics of this method are a rapid identification and full resolution of the combination of events causing the top event. However, the path or chain of events which led to the occurrence of the top event may not be readily apparent without consulting the fault tree structure. Because this procedure identifies faults directly, it is probably more suitable to the diagnostic goal of isolating faulty components of the system.

The algorithm given below uses the fault tree directly to derive the diagnosis procedure. The advantage of doing this is to introduce flexibility in depth of resolution and to present the fault propagation path at all times.

Gate Search Diagnosis Algorithm

1. Start with a top event which is true. Call it the current gate.

2. If all inputs to the current gate are primal, stop. Otherwise continue to step (3). If the stop gate is an OR gate, then any of the inputs are causative. Verify each to resolve. If the stop gate is an AND gate, then all of the inputs are causative.

3. If current gate is an AND gate go to step (4). Current gate is an OR gate. Select most probable input and verify that it is true or false. Take next step based on the result of the input variable test as specified by the following conditions.

    1. Input is a primal event and is true -- stop.

2. Input is a gate and is true -- call this input the current gate and go to step (2).

3« Input is false -- select next most probable input and verify as true or false* Repeat until no more inputs are left*

4. No inputs exist which are true -- implies that the current gate is untrue. Stop with the diagnosis unresolved*

4. Current gate is an AND gate. All inputs are true. For further resolution call each input a top event and begin again at step (1).

The procedure resulting from the gate search algorithm for the top event "Temperature in stream 14 too high[11] is shown in Figure 6.

Clearly the emphasis here is to locate the critical path of the fault propagation through the fault tree. Examining the cutset number 25 -- M10 (-1) AND TRC on manual -- the path in the fault tree (see Figure 3) is followed rather closely by the diagnosis procedure. This particular cutset is shown by a dashed line in Figure 6. An important feature in this algorithm is flexibility in the depth of

resolution obtained. Whenever an AND gate is encountered, a decision can be made whether to continue the diagnosis. When time is a critical factor, this flexibility is quite desirable. At any point in the procedure the cause and effect chain is clear, allowing the operator to stop the procedure when he wishes. In contrast, full resolution is required in the cutset algorithm in order to imply the cause and effect chain. For operating systems, the gate search diagnosis algorithm is superior to the cutset algorithm.

DIAGNOSTIC PROCEDURES FOR MULTIPLE TOP EVENTS

An important issue, especially with regard to complex systems, is the identification of the prime causes which underlie the simultaneous occurrence of two or more top events. When multiple events occur simultaneously, it is usually due to a common cause. (This is the main idea behind alarm analysis)• A significant portion of the possible causes of the individual events can be initially screened by discovering and exploiting the known process interactions. These common causes can be identified by locating those variables which are common to two or more fault trees. These variables, once identified, define the known process interactions. Therefore, certain combinations of top events (or top event patterns) can be directly attributed to these

common causes. Of course, the top event pattern may be explained by other causes occurring simultaneously, but this is less likely. The diagnostic search can be effectively confined to identifying the causes which occur below these common variables in the fault trees. Individual causes for the multiple top events can serve as a backup method should the common variable approach fail to isolate the causes.

An algorithm has been developed for the generation of multiple top event diagnosis procedures. Basically the algorithm does three things:

1. Identify the common variables.

2. If a pattern occurs which has not been previously defined, then develop a procedure to determine the appropriate common variable. This is accomplished by locating inactivating edges (zero gain) in the path(s) between the common variable and the top events. The failures or conditions which cause the inactivating edges are checked to determine the appropriate common variable.

3. Diagnose the known patterns by starting at the common variable(s) and applying a single top event diagnosis algorithm.

The identification of common variables can be accomplished using the system digraph and the fault trees. The primary identification occurs when the system digraph is being constructed. The digraph for the first top event is constructed in the normal way. The subsequent digraphs are constructed until a variable is encountered which has been previously developed (under a different top event). This variable is "marked" as a candidate common variable and is placed on a separate list. The variable is listed with the following information: (1) the variable's associated top events and (2) whether the variable is part of a negative feedback loop.

The list of common variables is further refined by examining the structure of the fault trees. Such considerations as the sign and magnitude of the variable, whether it is on a negative feedback loop and the existence of AND gates above the common variables are used to classify and eliminate some variables from consideration.

The key idea in this common variable analysis is that the diagnosis effort can be reduced for observed patterns by considering first the common causes. Should the common cause not be the true cause, then considering each top event independently would be justified. Common variable analysis also aids in resolving inconsistent real time input data.

This issue is treated in the following section.

INCONSISTENT REAL TIME DATA

An important assumption in the foregoing development is that, for multiple top events, the "patterns" of top events are defined by locating the common variables. There is a possibility that, in real time, a pattern of top events may occur which was not defined by the common variables. An inconsistent(situation arises when an inactivating failure has occurred between the common variable and one of the top events in a known pattern. The resulting real time pattern is not one previously defined but is in fact a subset of some known pattern. For instance, in a hypothetical system with n known patterns, pattern 1 has the top events (1,2,3) and pattern 2 has the top events (2,3,4). Each pattern is due to a different common variable. To visualize this problem refer to Figure 7- Figure 7 is a portion of the system digraph. The top events can be thought of as "causing" the known patterns and are represented by directed edges.

Suppose that the pattern (2,3) occurs. Which common variable should be utilized for the diagnosis? To answer this question, an additional diagnostic procedure must be

executed to determine which of the common variables should be used. This preliminary procedure must determine which of the paths (through top events 1 or 4) is inactive due to a failure mode. Once this is determined, the diagnostic procedures can be developed in the usual manner from the appropriate common variable.

The "unknown pattern[11] portion of the algorithm was developed for use in real time with a process computer. One possibility, for generating these unknown pattern diagnosis procedures a priori, is to permute the known event patterns, inactivating each event and all combinations of inactivated events. This approach also assumes (obviously) that an automatic method for generating the diagnostic procedures would be used to generate the large number of diagnosis trees required.

EVALUATION OF HAZARD FAULT TREES

An important advantage to the fault tree approach is that the probability of hazardous top events can be recomputed utilizing the diagnostic data. In an interactive mode with a process computer the relative hazard state of the process could be evaluated using real time operating and diagnostic data. These data are now real time data -- not a

priori estimates of failure rates. If structured propet recomputing hazardous event probabilities and importance c other failures could be accomplished quickly and easily on the process computer.

FUTURE RESEARCH TOPICS

An important assumption in the method described above is that the diagnosis procedures can be executed by observing key process variables without changing the system configuration. In some cases this may not be possible. Changes in the system configuration! such as bypassing and isolation of certain system components, change the cause and effect relationships in the digraph and therefore the fault trees. Describing these system changes in the form of procedures and representing these procedures with time-dependent digraph modeling techniques [13]? may provide a means to develop these types of diagnosis procedures.

To fully demonstrate the PTA based diagnosis techniques, computer codes need to be implemented for the following applications.

1. A priori generation of diagnostic procedures. This would result in a set of documents for operating use. These documents could be presented either as

off-line written documents or on-line CRT displays.

2. Interactive (semi-automatic) diagnostic procedures suitable for use with a process computer. This approach would use the monitoring capabilities of the process computer to carry out parts of the diagnosis procedures automatically while requesting information as needed from the operator. Presumably the current status of the diagnosis would be presented to the operator as the diagnosis proceeded.

Input data which are necessary for the algorithms would be the system digraph, the diagnosis and hazard fault trees, and a common variables list. Application of these algorithms to more examples would serve to verify their accuracy and to make refinements to the diagnosis algorithms as needed. The practical use of these diagnosis procedures dictates that some form of automatic generation be available. Manual generation of these procedures, even using an organized method such as the one described, would be too time consuming to be used widely.

The use of these diagnostic procedures for the design of alarm systems and sensor placement is another possible area of future research. Most sensors in chemical processes are placed for preventive or control measures. Some sensors, however, are placed specifically for rapid diagnosis reasons. An example of this type of sensor is the placement of sight glasses on individual filter leaves for the discovery of the leaf that is faulty. The availability of diagnostic procedures coupled with hazard state evaluations will allow the designer to determine, a priori, important placements of diagnostic types of sensors and alarms. Similarly, it will be possible to determine which process variables are important to monitor with the process computer. Alarm placement and control panel design will benefit from an assessment of placement and grouping of alarms for diagnosis purposes.

REFERENCES

[1] Andow, P.K. and Lees, F.P., "Process Computer Alarm Analysis: Outline of a Method Based on List Processing", Trans. Instn. Chem. Engrs., jLIt 1975, pp. 195-208

[2] Grumbach, R. and Pfeiff, N.O.M., "Improving Plant Supervision Efficiency by a New Technique for On-line Alarm and Disturbance Analysis", Paper presented at the "Enlarged Halden Program Group Meeting on Computer Control and Fuel Research Related to Safe and Economic Plant Operation", Sandefjord, Norway, June, 1974

[3] Kay, P.CM. "On-Line Computer Alarm Analysis", Ind. Electron., 1, 1966, p. 50-53

[4] Lambert, H.E. and Yadigaroglu, G., "Fault Trees for Diagnosis of System Condition", Nuc. Sci. and Eng., J6L£,1, Jan. 1977, p. 20-34

[5] Lapp, S.A., PhD Thesis, Carnegie-Mellon University, Dept. of Chemical Eng. 1978

[6] Lapp, S.A. and Powers, G.J., "Computer-aided Synthesis of Fault Trees", IEEE Trans, on Reliability, R-26T April 1977, PP- 2-13

[7] Lawley, H.G., Loss Prevention ^ Vol. £. $ A CEP Technical Manual, AIChE, 1974

[8] Patterson, D., "Application of Computerised Alarm Analysis System to a Nuclear Power Station", Proc. I.E.E.,ULSi12, Dec. 1968, p. 1858-1864

[9] Pieper, W.J. and Pinkus, A.L., "Computer Generated Troubleshooting Trees: The Program", Air Force Human Resources Lab. - Report number AFHRL-TR-74-20(II), July,1974

[10] Powers, G.J. and Lapp, S.A., "Computer-aided Fault Tree Synthesis" , CEP, 12,4,April 1976, pp. 89-93

[11] Powers, G.J., and Tompkins, F.C., "Fault Tree Synthesis for Chemical Processes", AIChE J, ZSL$ 2, Mar. 1974, PP.376-387

[12] Society of Automotive Engineers, "Design Analysis Procedure for Failure Modes, Effects and Criticality Analysis", SAE Aerospace Recommended Procedure ARP 926, September 15, 1967

[13] Shaeiwitz, J. , Lapp, S.A., and Powers, G.J., "Fault Tree Analysis of Sequential Systems", I&EC Proc. Des. & Dev., l£i4, Oct. 1977, pp.529-549

[14] Teague, T.L., MS Thesis, Carnegie-Mellon University, Dept. of Chemical Engineering, 1978.

[15] Melbourne, D., "Alarm Analysis and Display at Wyfla Nuclear Power Staion",Proc. I.2.E., 115,11, Nov. 1968, pp. 1726-1732

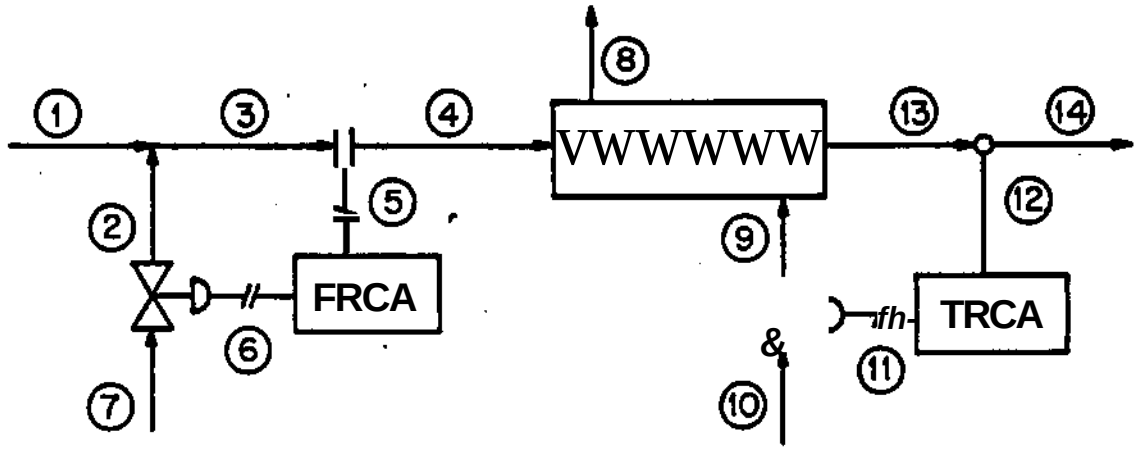

Figure 1. Flowsheet for mixer, heat exchanger system.

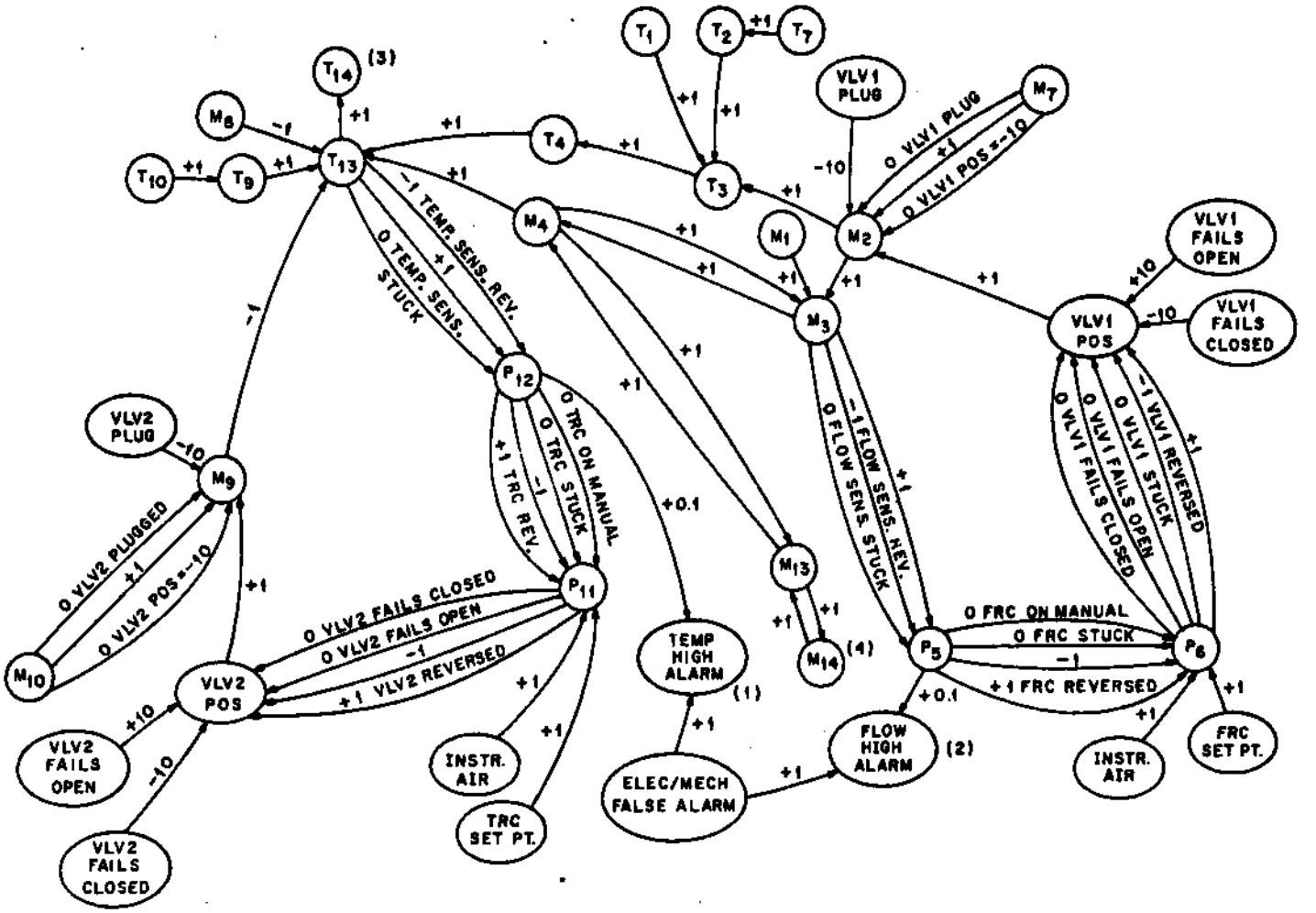

Figure 2. System digraph for mixer, heat exchanger system.

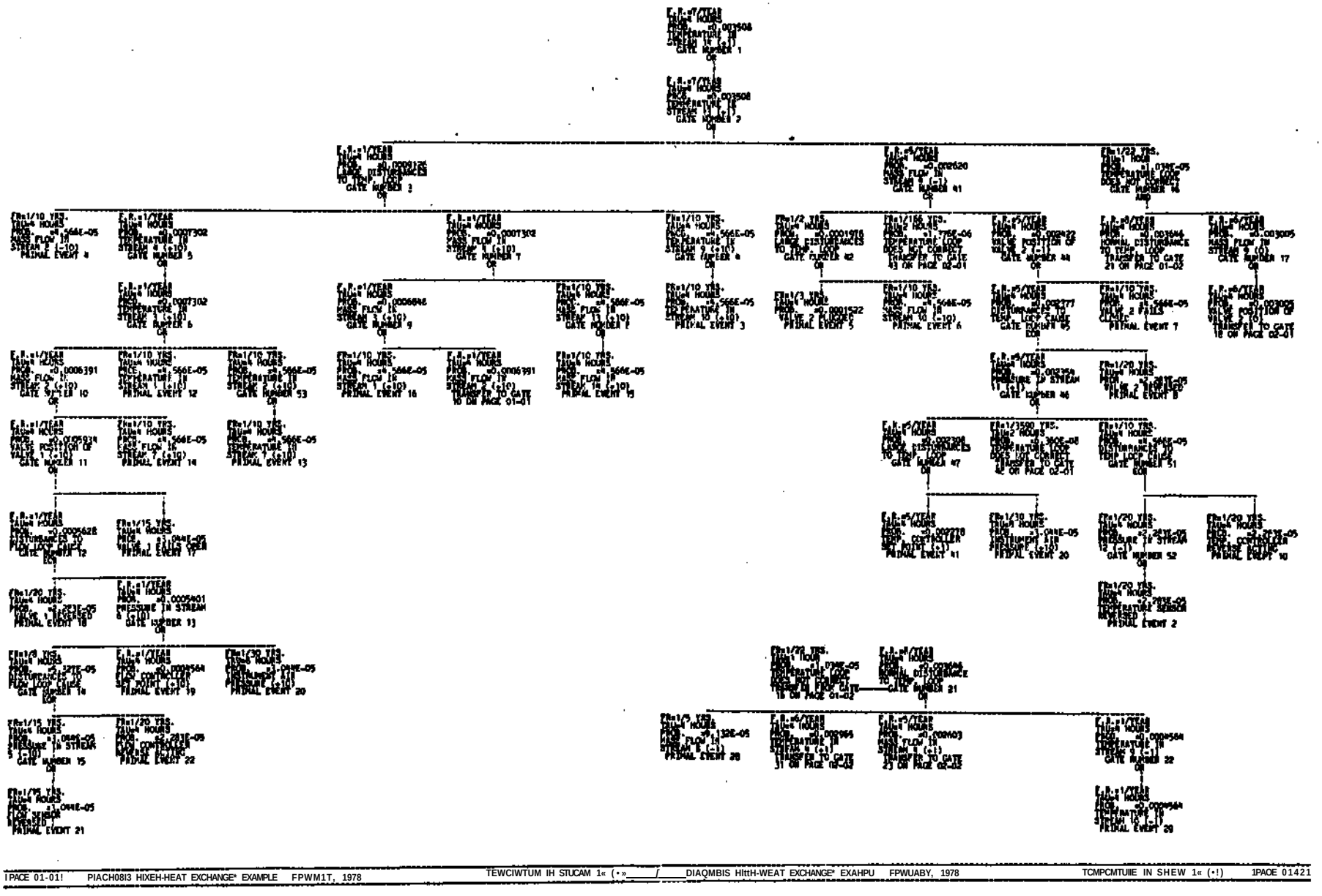

Figure 3. Fault tree for Temperature in Stream 14 Too High,

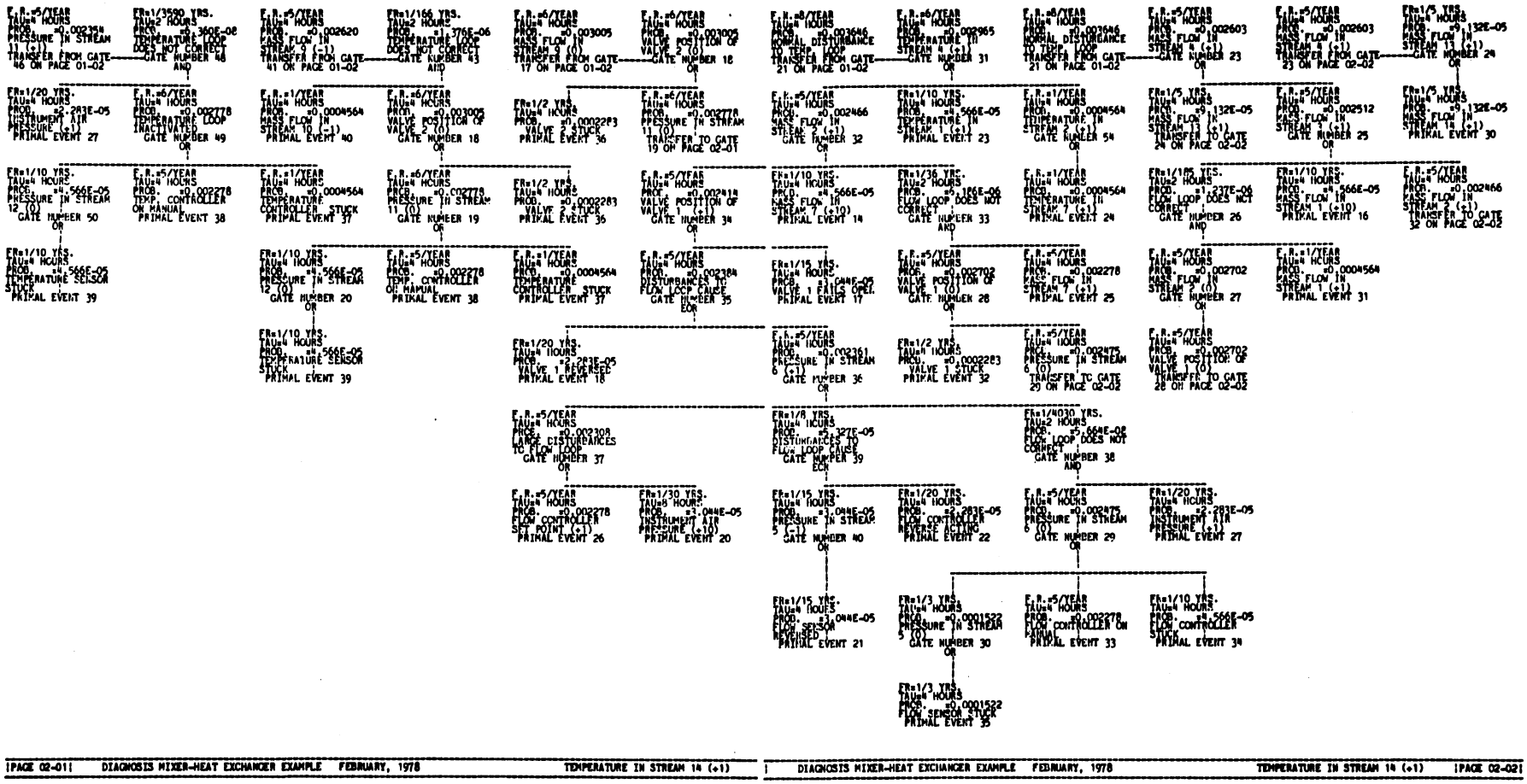

Figure 3. Cont'd.

86 MINIMAL CUT SETS GENERATED.

TOP EVENT PROBABILITY- 3.5163E-03

TOP EVENT RATE OF OCCURRENCE* 1 EVENT/ 1.3E-01 YEARS

M.C.S. NO. 1 RATEsi/ 2.0E-01 YR. PROB.« 2.28E-03
EVENT 41(2.28E-03) TEMPERATURE CONTROLLER SET POINT (•1)

M.C.S. NO. 2 RATE»1/ 1.0E+00 YR. PROB.« 4.56E-04
EVENT 19(4.56E-04) FLOW CONTROLLER SET POINT (+10)

M.C.S. NO. 3 RATEsi/ 3.0E+00 YR. PROB.i 1.52E-04
EVENT 5(1.52E-04) VALVE 2 PLUGGED

M.C.S. NO. 4 RATE=1/ 1.0E+01 YR. PROB.« 4.57E-05
EVENT 13(4.57E-05) TEMPERATURE IN STREAM 7 (+10)

M.C.S. NO. 5 RATE*1/ 1.0E+01 YR. PROB.« 4.57E-05
EVENT 3(4.57E-05) TEMPERATURE IN STREAM 10 (+10)

M.C.S. NO. 6 RATE=1/ 1.0E+01 YR. PROB.« 4.57E-05
EVENT 7(4.57E-05) VALVE 2 FAILS CLOSED

M.C.S. NO. 7 RATEsi/ 1.0E+01 YR. PROB.s 4.57E-05
EVENT 6(4.57E-05) MASS FLOW IN STREAM 10 (-10)

M.C.S. NO. 8 RATE:•1/ 1.0E+01 YR. PROB.s 4.57E-05
EVENT 4(4.57E-05) MASS FLOW IN STREAM 8 (-10)

M.C.S. NO. 9 RATE=1/ 1.0E+01 YR. PROB.« 4.57E-05
EVENT 14(4.57E-05) MASS FLOW IN STREAM 7 (+10)

M.C.S. NO. 10 RATE:•1/ 1.0E+01 YR. PROB.s 4.57E-05
EVENT 12(4.57E-05) TEMPERATURE IN STREAM 1 (+10)

M.C.S. NO. 11 RATEsi/ 1.0E+01 YR. PROB.* 4.57E-05
EVENT 16(4.57E-05) MASS FLOW IN STREAM 1 (+10)

M.C.S. NO. 12 RATE«•1/ 1.0E+01 YR. PROB.s 4.57E-05
EVENT 15(4.57E-05) MASS FLOW IN STREAM 14 (+10)

M.C.S. NO. 13 RATE»1/ 1.5E+01 YR. PROB.*.. 3u.04E-05
EVENT 17(3.04E-05) VALVE 1 FAILS OPEN

M.C.S. NO. 14. RATE*=1/ 1.5E+01 YR. PROB.s 3.04E-05
EVENT 21(3.04E-05) FLOW SENSOR REVERSED

M.C.S. NO. 15 RATE=1/ 2.0E+01 YR. PROB.« 2.28E-05
EVENT 10(2.28E-05) TEMP. CONTROLLER REVERSE ACTING

M.C.S. NO. 16 RATE=1/ 2.0E+01 YR. PROB.s 2.28E-05
EVENT 2(2.28E-05) TEMPERATURE SENSOR REVERSED

M.C.S. NO. 17 RATE»1/ 2.0E+01 YR. PROB.» 2.28E-05
EVENT 8(2.28E-05) VALVE 2 REVERSED

M.C.S. NO. 18 RATE=1/ 2.0E+01 YR. PROB.s 2.28E-05
EVENT 22(2.28E-05) FLOW CONTROLLER REVERSE ACTING

M.C.S. NO. 19 RATE*1/ 2.0E+01 YR. PROB.= 2.28E-05
EVENT 18(2.28E-05) VALVE 1 REVERSED

M.C.S. NO. 20 RATE=1/ 3.0E+01 YR. PROB.s 3-04E-05
EVENT 20(3.04E-05) INSTRUMENT AIR PRESSURE (+10)

M.C.S. NO. 21 RATE-1/ 4.4E+01 YR. PROB.» 5.20E-06
EVENT 26(2.28E-03) FLOW CONTROLLER SET POINT (+1)
EVENT 38(2.28E-03) TEMPERATURE CONTROLLER ON MANUAL

M.C.S. NO. 22 RATE=1/ 2.2E+02 YR. PROB.s 1.04E-06
EVENT 29(4.56E-04) TEMPERATURE IN STREAM 10 (+1)
EVENT 38(2.28E-03) TEMPERATURE CONTROLLER ON MANUAL

M.C.S. NO. 23 RATE-1/ 2.2E+02 YR. PROB.a 1.04E-06
EVENT 24(4.56E-04) TEMPERATURE IN STREAM 7 (+1)
EVENT 38(2.28E-03) TEMPERATURE CONTROLLER ON MANUAL

M.C.S. NO. 24 RATE=1/ 2.2E+02 YR. PROB.s 1.04E-06
EVENT 26(2.28E-03) FLOW CONTROLLER SET POINT (+1)
EVENT 37(4.56E-04) TEMPERATURE CONTROLLER STUCK

M.C.S. NO. 25 RATE-1/ 2.2E+02 YR. PROB.« 1.04E-06
EVENT 38(2.28E-03) TEMPERATURE CONTROLLER ON MANUAL
EVENT 40(4.56E-04) MASS FLOW IN STREAM 10 (-1)

M.C.S. NO. 26 RATE=1/ 4.4E+02 YR. PROB.s 5.20E-07
EVENT 26(2.28E-03) FLOW CONTROLLER SET POINT (+1)
EVENT 36(2.28E-04) VALVE 2 STUCK

M.C.S. NO. 27 RATE=1/ 1.1E+03 YR. PROB.» 2.08E-07
EVENT 24(4.56E-04) TEMPERATURE IN STREAM 7 (+1)
EVENT 37(4.56E-04) TEMPERATURE CONTROLLER STUCK

M.C.S. NO. 28 RATE=1/ 1.1E+03 YR. PROB.» 2.08E-07
EVENT 37(4.56E-04) TEMPERATURE CONTROLLER STUCK
EVENT 40(4.56E-04) MASS FLOW IN STREAM 10 (-1)

M.C.S. NO. 29 RATE-1/ 1.1E+03 YR. PROB.«= 2.08E-07

M.C.S. NO. 30 RATE-1/ 1.1E+03 YR. PROB.<= 2.08E-07

Figure 4. Top 73 80 Mir Pl 01 c80-8 for
Temperature 15 80 03 S HOO> High.

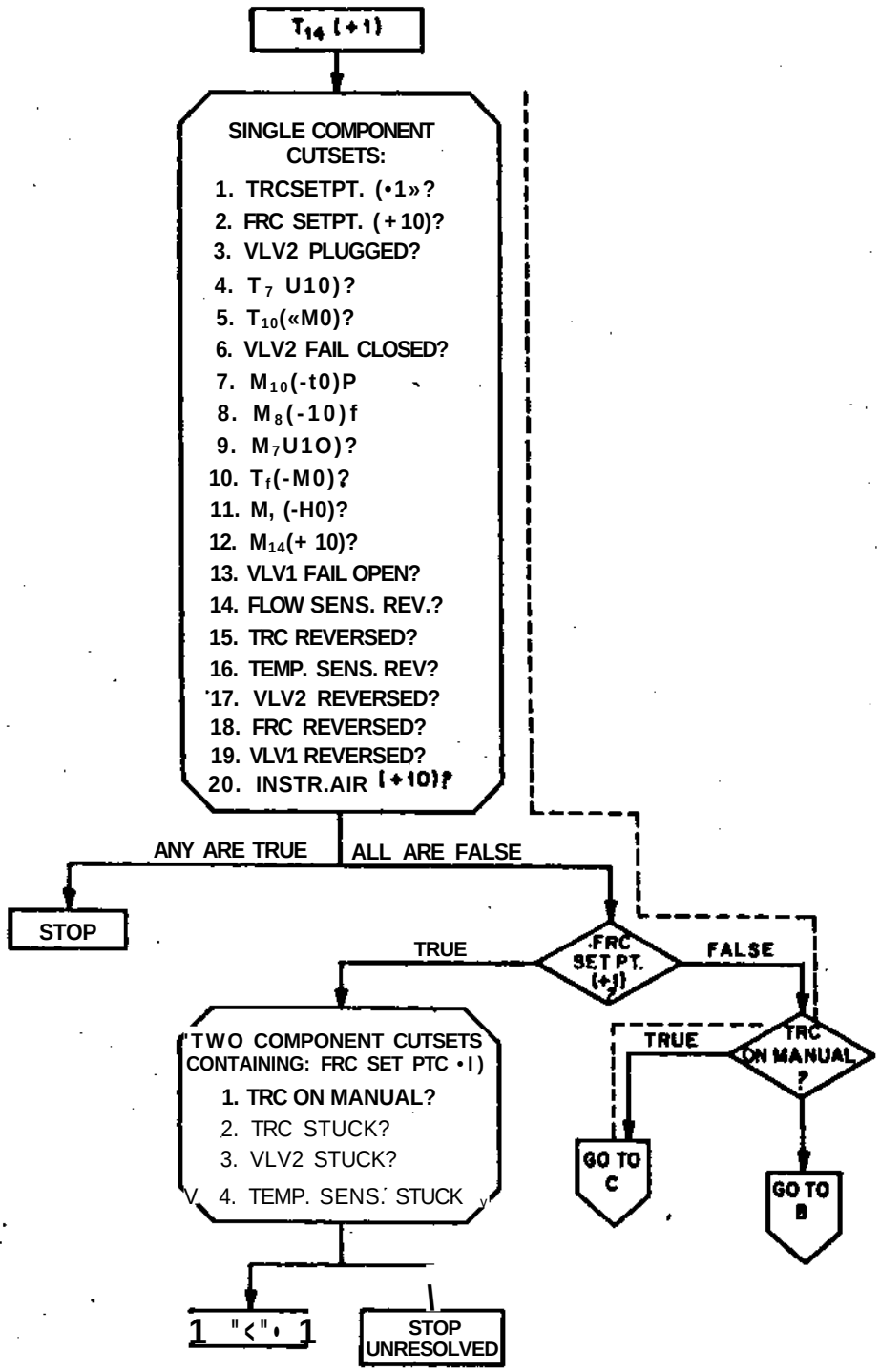

Figure 5. Diagnosis procedure for Temperature in Stream 14 Too High using the Cutset Diagnosis Algorithm.

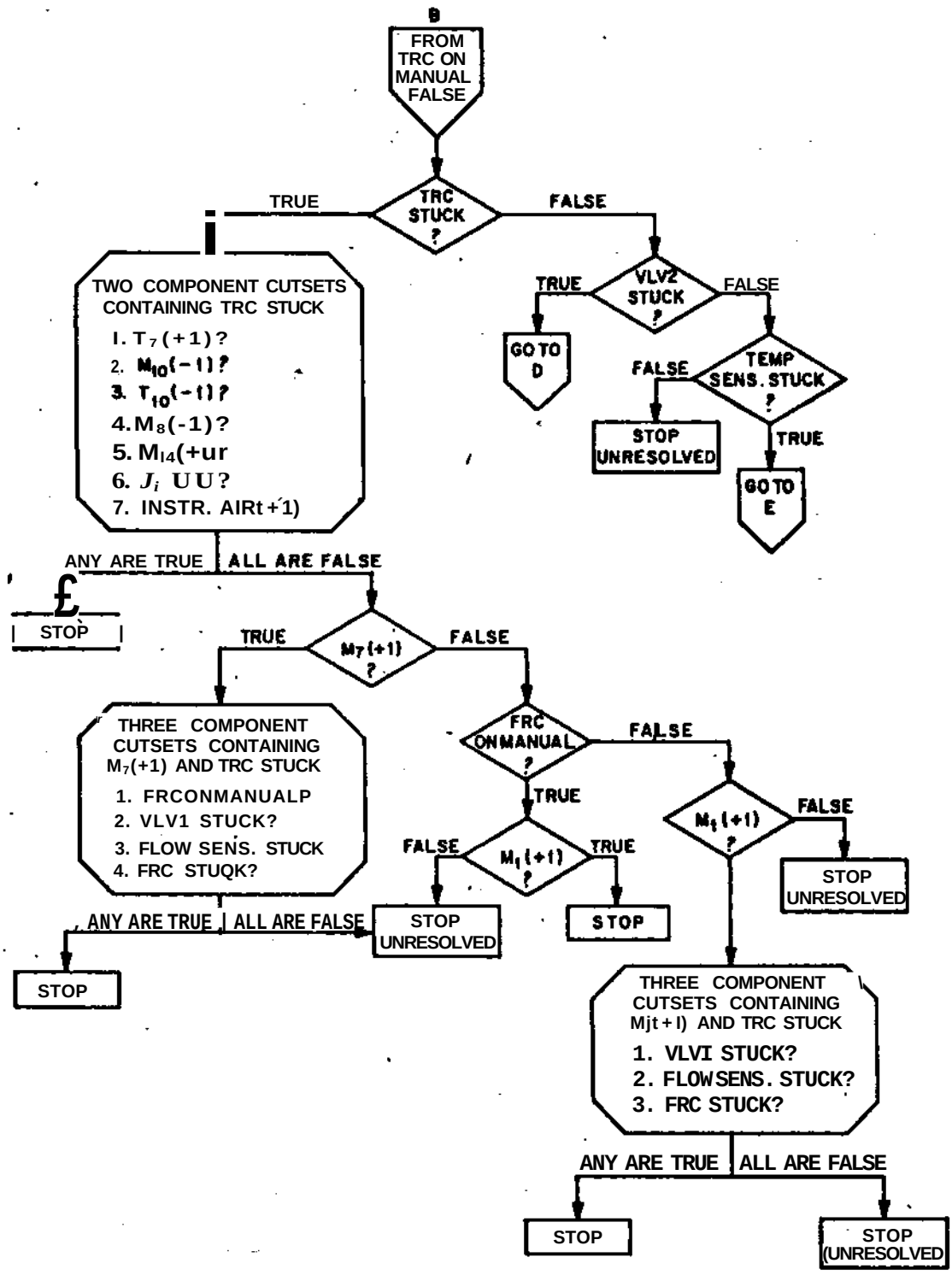

Figure 5. Cont'd.

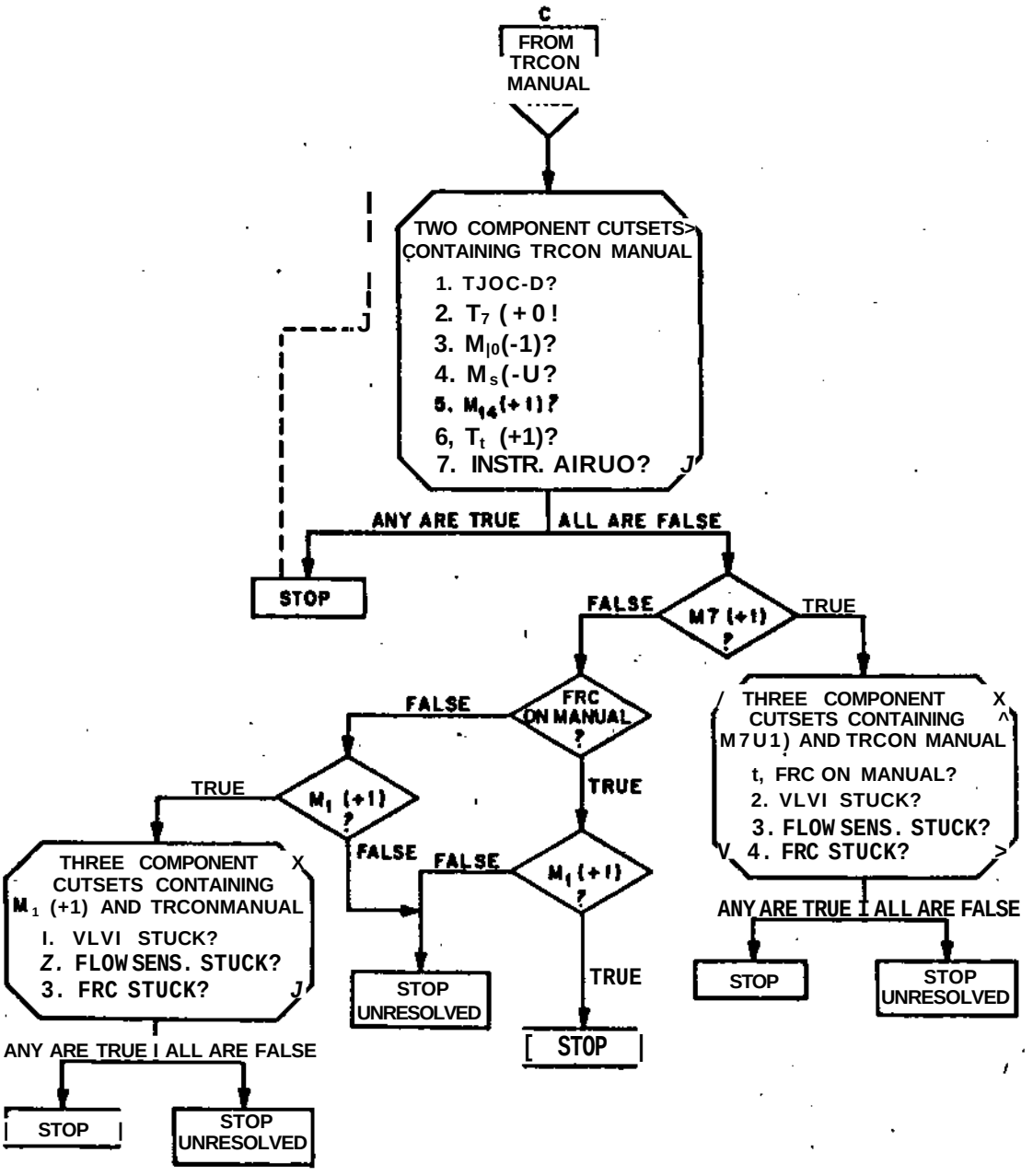

Figure 5. Cont'd.

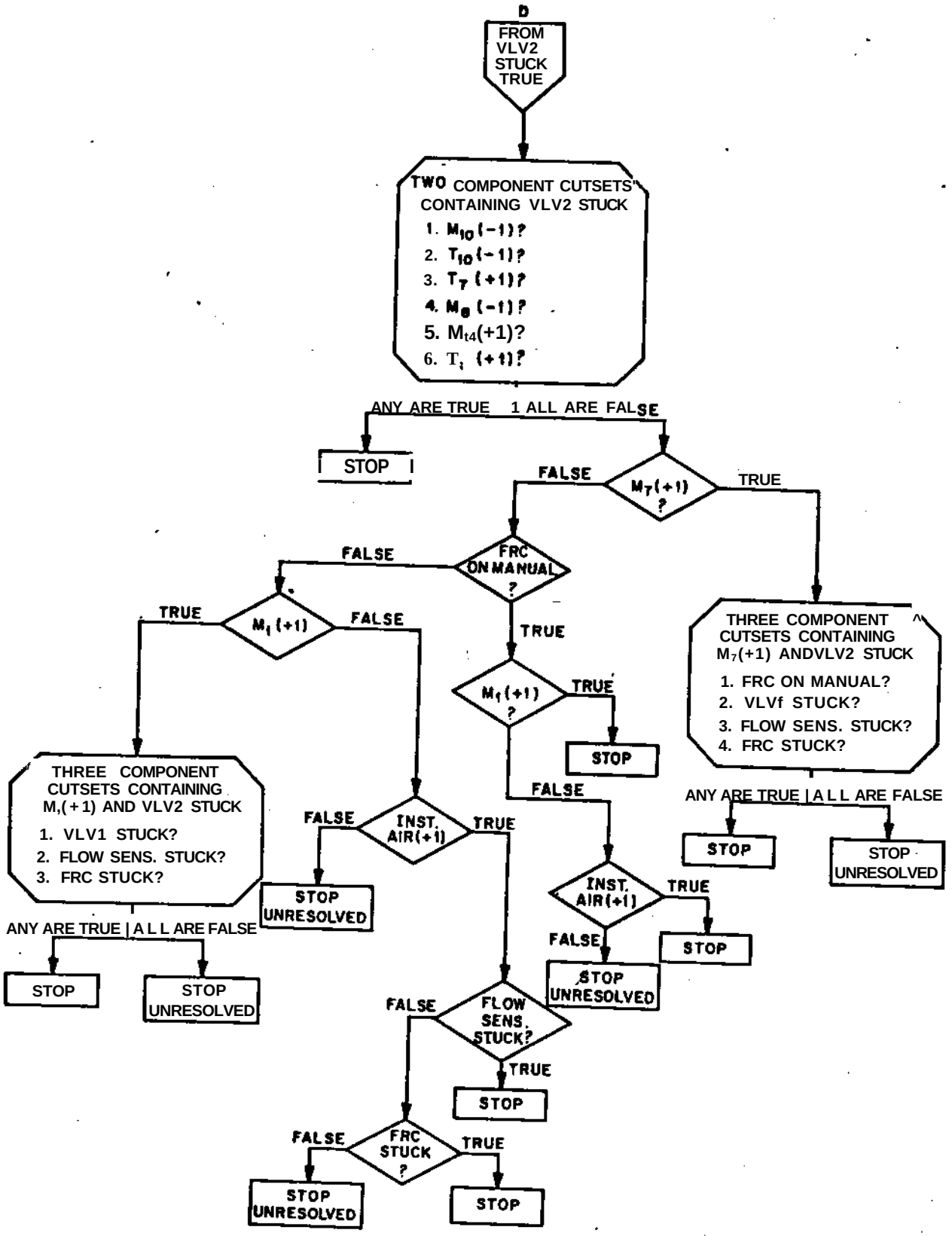

Figure 5, Cont'do

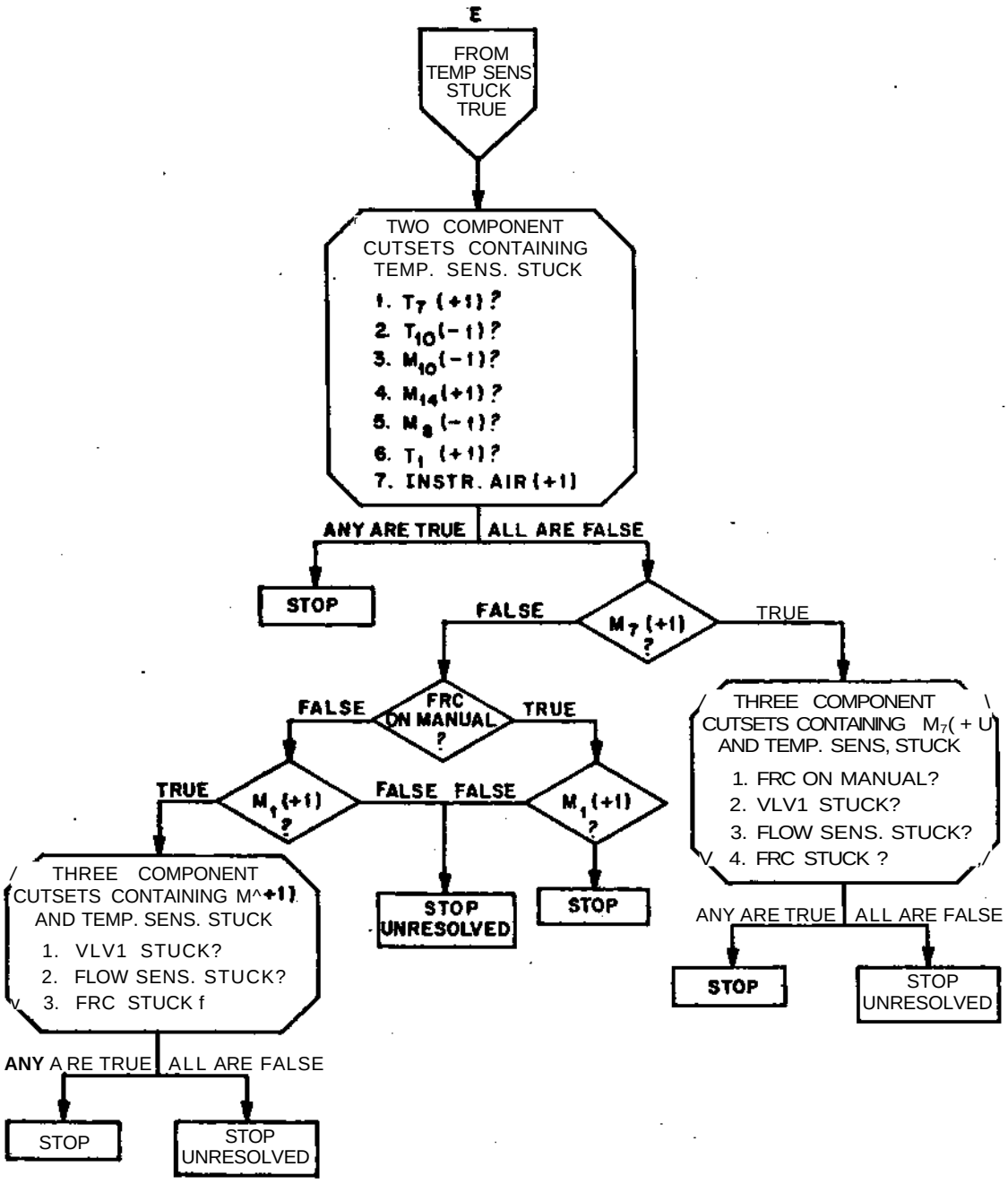

Figure 5. Cont'd.

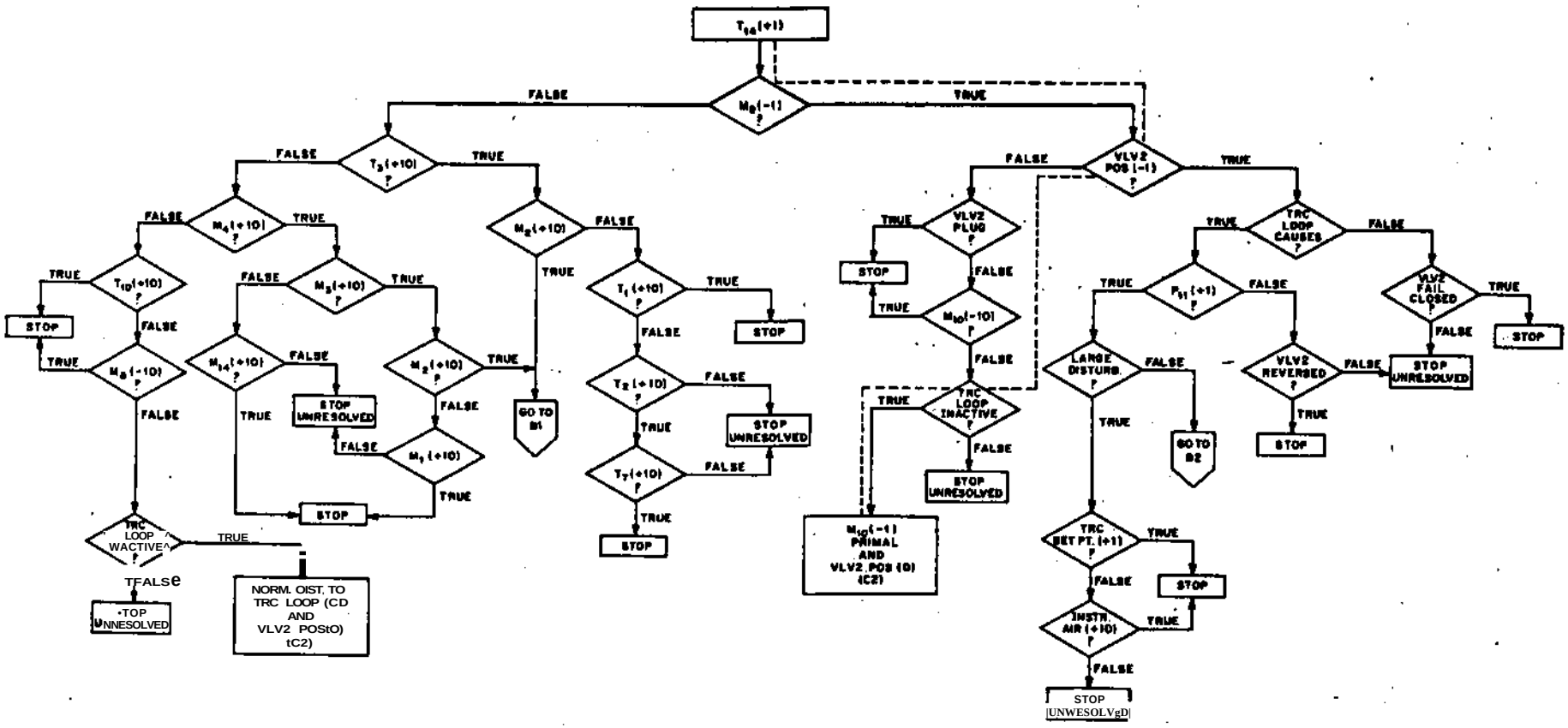

Figure 6. Diagnosis procedure for Temperature in Stream 14 Too High using the Gate Search Diagnosis Algorithm,

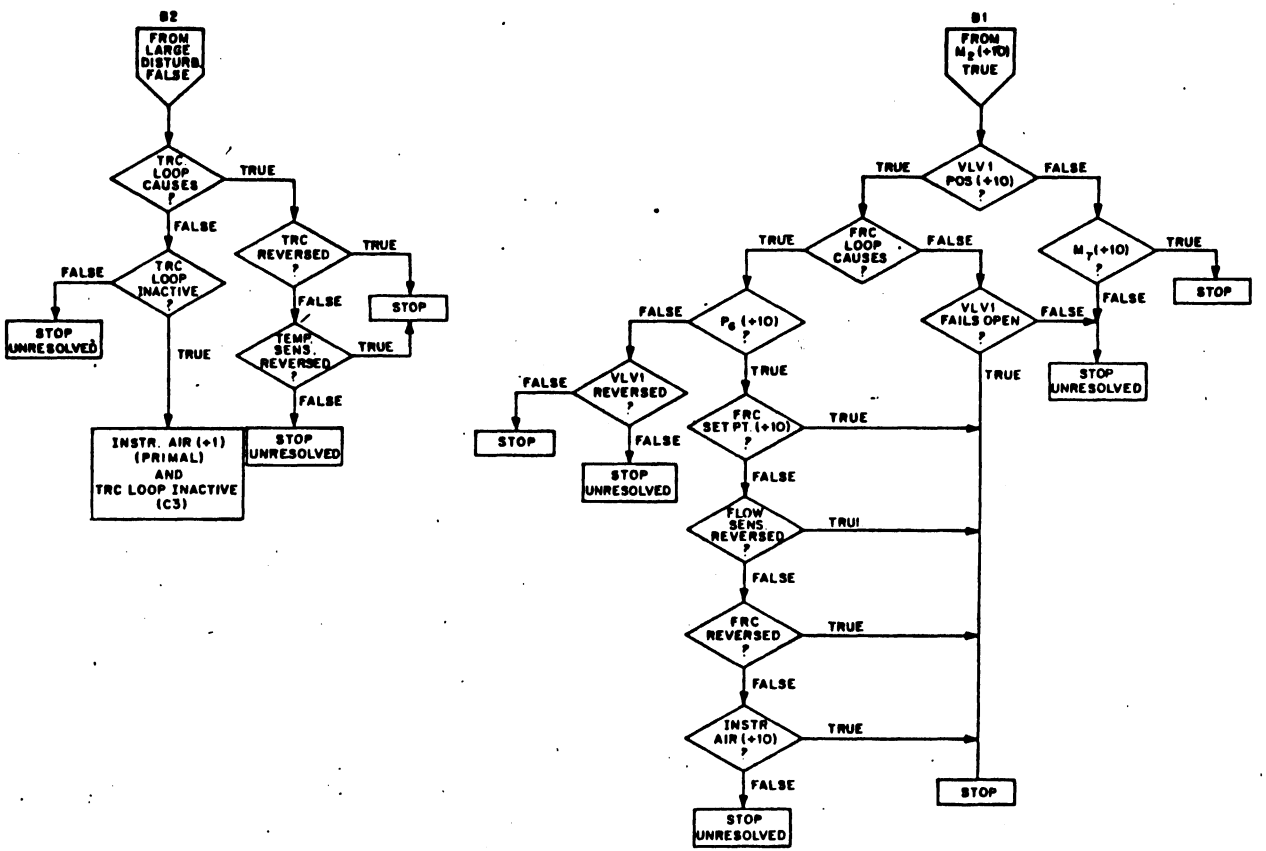

Figure 6. Cont'd.

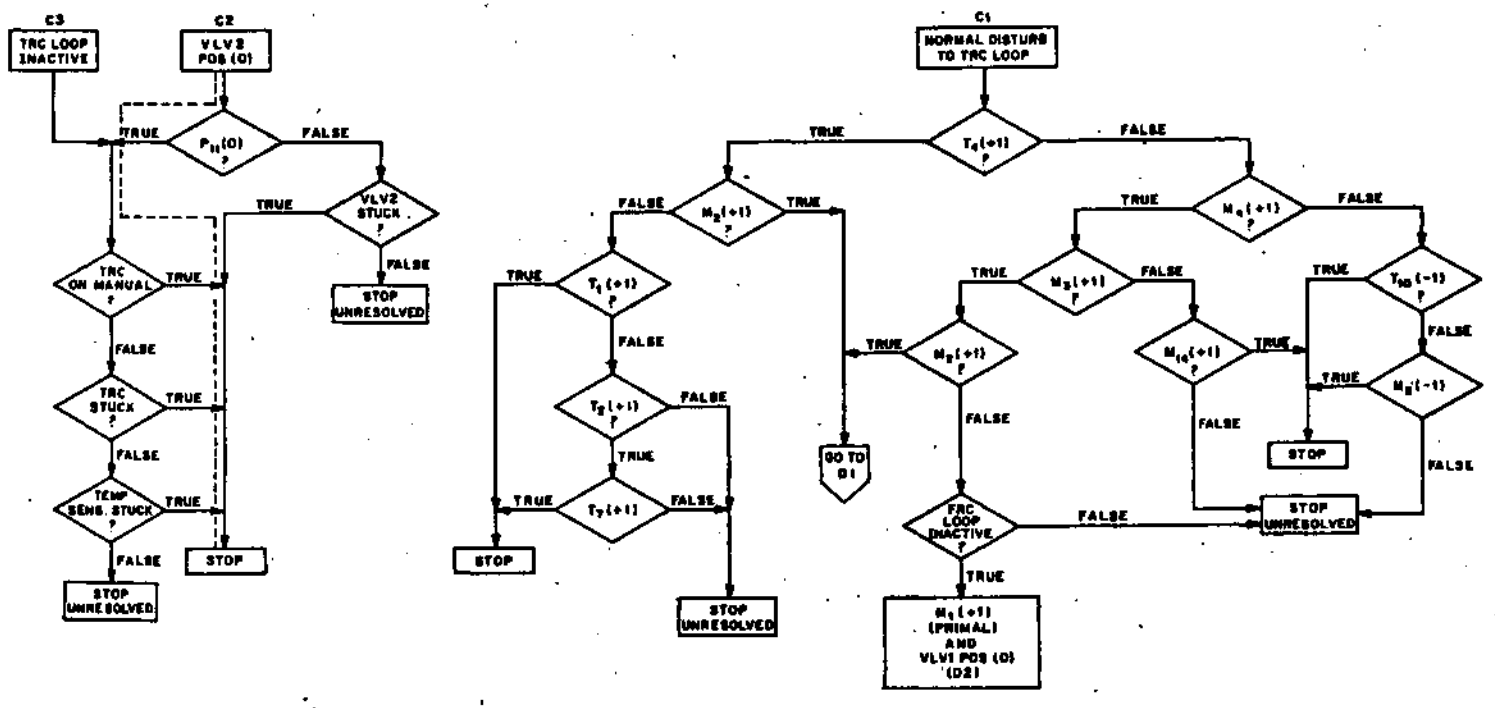

Figure 8. Cont.

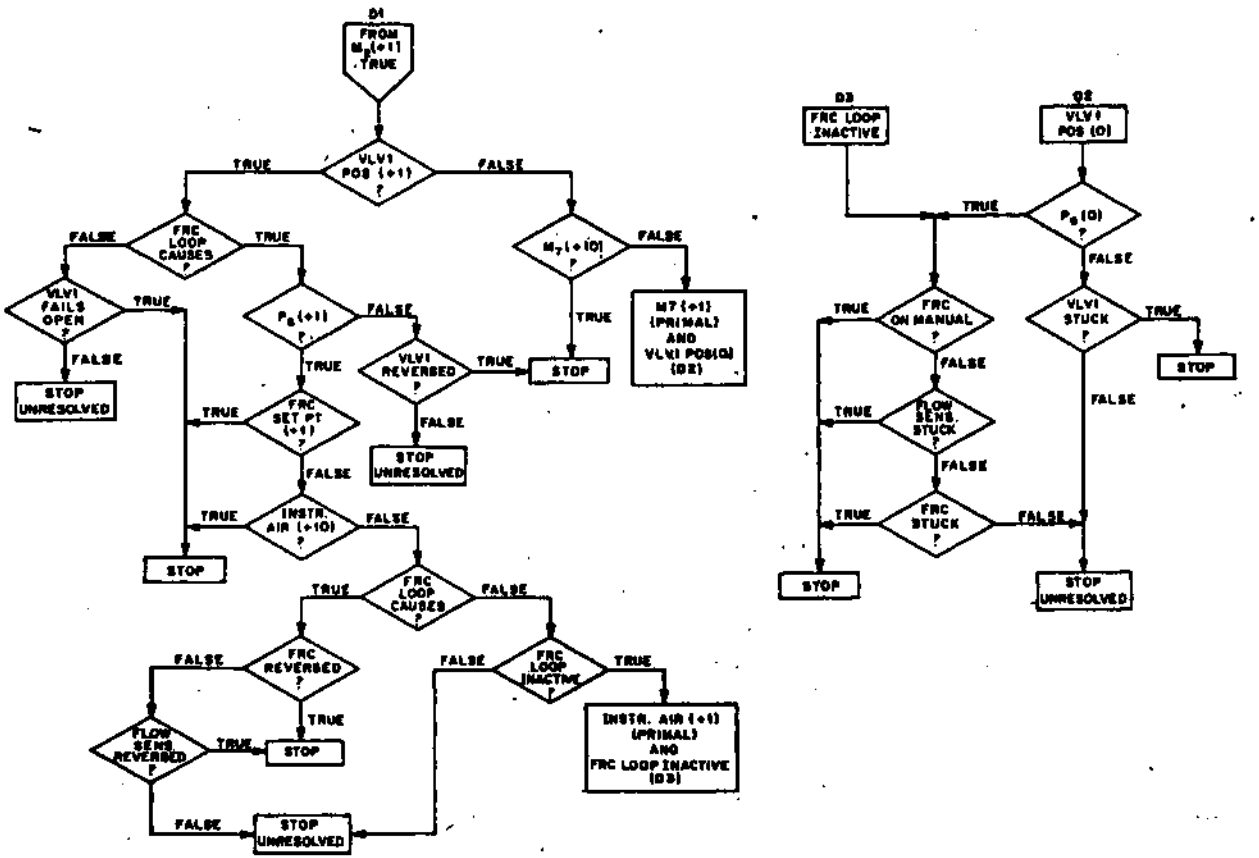
D1
FROM M8(+1) TRUE
VLV1 POS (+1) ?
TRUE
FALSE
FRC LOOP CAUSES ?
VLV1 FAILS OPEN ?
STOP UNRESOLVED
P8(+1) ?
VLV1 REVERSED ?
STOP
M7(+10) ?
M7(+1) (PRIMAL) AND VLV1 POS(0) (D2)
FRC SET PT (+1) ?
INSTR. AIR (+10) ?
FRC REVERSED ?
FLOW SENS. REVERSED ?
FRC LOOP INACTIVE ?
INSTR. AIR (+1) (PRIMAL) AND FRC LOOP INACTIVE (D3)
D3
FRC LOOP INACTIVE
D2
VLV1 POS (0)
P8(0) ?
FRC ON MANUAL ?
FLOW SENS. STUCK ?
FRC STUCK ?
VLV1 STUCK ?

Fig_ 6. D t'd.

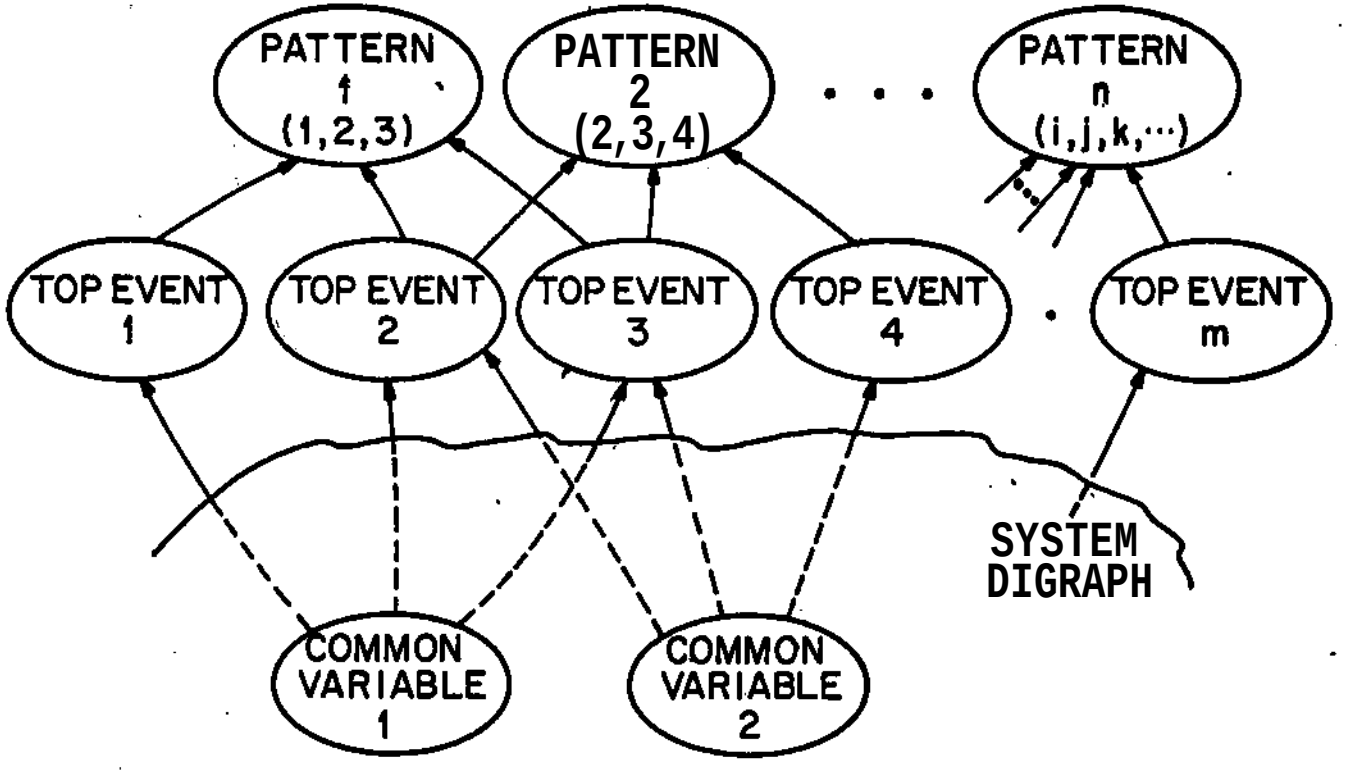

Figure 7. System digraph for multiple top event patterns.

Table 1« Summary of recent research in diagnosis. Ali except Pieper and Pinkus 10 are for chemical/nuclear systems and "on-line" testing. Pieper and Pinkus developed their system for electrical systems and "off-line" testing.

Investigators	Output FormAt	Input Data	Modeling Basis	Multiple Events	Inconsistent Information	Analysis of Secondary Effects	Generation Method	Completeness	Method Tested
Andow and Lees [1]	Off-line alarm trees	Flowsheet, Unit models	Directed graphs from functional models	yes	no	no	Automatic - easily updated	No failure modes	programmed, tested on simple system
Grumbaoh and Pfeiff [33	On-line alarm analysis	Event ohains, Alarms	Event matrices • determined manually	yes	no	yes	Automatio alarm analysis, models manually generated	No failures, No assurance of "all" event chains	no
Pieper and Pinkus [10]	Off-line diagnosis trees	System topography, Teat results	Signal flow from switch positions	no	no	no	Automatio - easily updated	Not all comp-onents were considered "breakable"	programmed, tested on 300 comp. system
Lambert and Yadigaraglu [51	Off-line diagnosis oheokllsts	Minimal outsets, Failure rates, and Test results	Fault trees, Minimal Cutsets	no	no	no	Manual - not easily updated	No outsets of order 3 or greater	Manually tested on simple system
Teague and Powers (this work)	On-line/ Off-line diagnosis troes	Flowsheet, Unit models, Failure rates, and Test results	Digraph models, Fault trees, Minimal Cutsets	yes	yes	yes, for selected events	Algorithms for automatio generation	Limited only by model completeness	Manually tested on simple system