Digraphs and Fault Trees

David J. Allen

Foster Wheeler Development Corporation, Livingston, New Jersey 07039

The principal impediment to the widespread adoption of fault tree analysis as a tool for safety and reliability analysis has been the difficulty in drawing realistic fault trees. The use of digraphs facilitates this task by allowing the analyst to focus upon the definition of the system under study rather than upon the logic of fault tree construction. In this paper the advantages of using digraphs have been presented together with several problems. An approach that avoids the drawing and review of fault trees is recommended.

Introduction

In the past 20 years, public concern and industrial recognition of the costs and consequences of hazard occurrence have engendered the new discipline of hazard prevention. To help guard against hazards that occur with unacceptable consequences or frequency, a variety of techniques, both inductive and deductive, have been devised. These techniques have one common characteristic: an emphasis upon the logical and rigorous examination of a potentially hazardous system. One technique that has been widely recognized as being especially appropriate for the identification of the causes and likelihood of specific hazards is fault tree analysis.

A fault tree is graphic representation of the failure logic of a system—the logical relationship between a specific event and its initiating or causal events. Through the analysis of the fault tree, the causes of the specific event can be determined as "minimal cut-sets" (i.e., as sets of events that are sufficient for the specific event to occur). Although fault tree analysis has been extensively used in nuclear and other energy-related industries, and in the chemical process and aerospace industries, its use has not become as widespread as its earlier proponents had envisaged. Indeed, many safety studies employing fault tree analysis have produced trivial results or have incurred unacceptably high costs. Regardless of how the blame for this state of affairs should be assigned, the basic problem appears to be that the synthesis of fault trees is a tedious and difficult task that requires skills beyond those normally possessed by process engineers or designers. As a result, the preparation of fault trees is time-consuming and is frequently marred by errors that individuals familiar with the system under study could readily identify. This not only wastes effort; it also undermines the credibility of the analysis.

The resolution of these problems requires both the facilitation of the task of fault tree synthesis and the closer involvement of process and other design engineers in this task. This can be achieved through the use of digraphs (directed graphs) to represent the system and of computer programs that will transform the digraph into a fully edited fault tree that can then be analyzed using other readily available computer programs.

The Use of Digraphs

To facilitate the synthesis of fault trees, a variety of strategies has been devised. These have employed mini-fault trees (Fussell, 1973), block diagrams (Caceres and Henley, 1976), and other network analysis techniques (Chu, 1976, and Nehem, 1973). Of these, it would appear that only those algorithms that use digraphs (Lapp and Powers, 1977; Allen and Rao, 1980) have been successful in handling feedback and negative feedforward control loops. It is these control loops that add most to the complexity of fault trees depicting failures in power plants or chemical processes. Accordingly, an ability to handle them is indispensible if we are to be successful in easing the task of fault tree synthesis.

A digraph (or directed graph) is a set of nodes connected by directed arcs and thus is a representation of the system that is particularly convenient for computer processing. In digraphs representing the failure behavior of a system, nodes can depict process variables (e.g., temperature, pressure), the system failure(s) or hazard(s) of interest, or component or subsystem failures (e.g., a failure in a relay etc.). Relationships between the nodes are embodied in the direct arcs between the nodes. These arcs may be conditional upon other events. The gain associated with each arc can be specified: If a positive deviation in a variable (or the occurrence of an event) represented by a node results in a positive deviation in the variable (or occurrence of the event) represented by a second node, then the gain of the arc between them is positive (Figure 1). Similarly, gains can be defined as being negative or zero: If a positive deviation in a variable or occurrence of an event results in a negative deviation in a second event, then the gain of the arc between the nodes representing these arcs or events will be negative (Figure 2). If deviations in a variable or occurrence of an event have no direct effect upon a second variable or event (perhaps when certain conditions apply), then the gain of the arc between the nodes representing those variables or events is zero (Figure 3). In drawing a digraph, arcs with zero gains are omitted unless they are conditional upon another event and lie between nodes also connected by an arc with a nonzero gain. Arcs that are conditional upon additional events are simply represented by placing the conditional node alongside the arc.

In the preparation of a digraph, a rigorous convention is often followed in defining nodes (Lapp and Powers, 1977; Andow, 1980). With this convention, nodes are restricted to representing system failures and hazards of interest, state variables (with no specified deviations), and specific component failures. These restrictions, however, frequently lead to artificial descriptions and even, in the hands of unwary analysts, to gross errors. I believe a great deal more can be accomplished if less restrictive rules for the definition of nodes are adopted and if the digraph is used simply as an information flow diagram. This not only provides an explicit description of physical reality but allows for a more natural development of the digraph, thus allowing the analyst to focus upon the definition of the

0196-4313/84/1023-0175$01.50/0

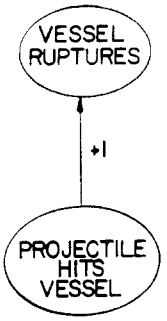

Figure 1. Positive gain.

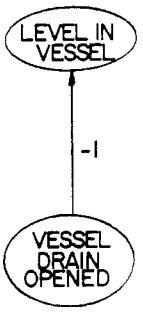

Figure 2. Negative gain.

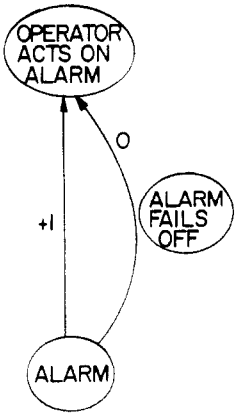

Figure 3. Zero gain.

system under study. Thus, for example, in modeling a simple pressure control loop (Figure 4a) we can extract the digraph from the relevant portion of the piping and instrument diagram, inserting the gains as appropriate (Figure 4b). In contrast, if rigid rules are followed, the control loop is depicted in a digraph by nodes representing air pressures in instrument air lines (Figure 4c), a depiction that seems somewhat artificial.

Errors in the digraph arise if the analyst insists on representing a state variable by a single node when both positive and negative deviations in the node are possible. This insistence can be confusing and can also introduce spurious negative feedforward and feedback loops and create other problems when modeling split range controllers and in situations where multiple streams combine.

As an example, consider the occurrence of a fire involving a flammable liquid stored in a tank (Figure 5a). Two causes of the fire are the ignition of liquid after it spills from an overfilled tank and ignition at a pump that has been allowed to run dry.

Drawing simple partial digraphs to represent these causes, we obtain Figure 5b if the rigid convention is followed and Figure 5c if the digraph is treated as an information flow diagram where nodes can represent deviations in state variables. The former digraph is erroneous in that it introduces a spurious negative feedforward loop

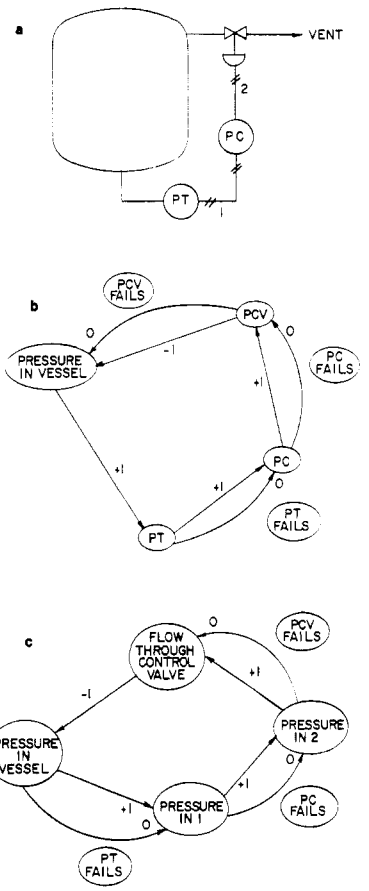

Figure 4. (a) Simple pressure control loop. (b) Digraph of pressure control loop (liberal rules). (c) Digraph of pressure control loop (restrictive rules).

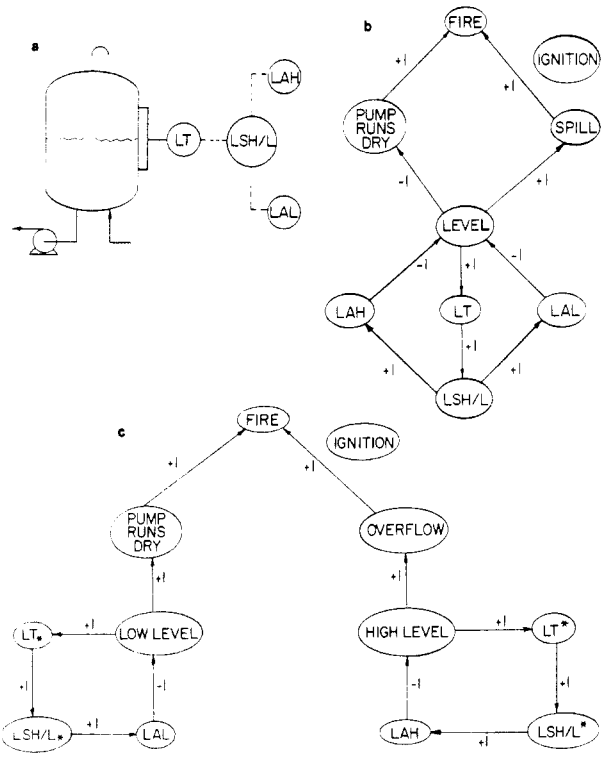

Figure 5. (a) Liquid storage tank. (b) Digraph (restrictive rules). (c) Digraph (liberal rules).

(level – pump runs dry – fire, level – overflow – spill) and implies that both high and low level alarm loops are applicable to deviations in level irrespective of their direction. The latter digraph is not only correct but it is also more

explicit and thus, I believe, easier to develop and follow.

Several approaches have been devised to define the gain associated with each arc in the digraph. While Allen and Rao (1980) simply require that the sign of the gain be provided (i.e., that gains be restricted to the values +1, 0, and −1), Lapp and Powers (1977) allow a discrete range of values to be assigned to each gain (+10, +1, 0, −1, −10), and Kumamoto et al. (1981) suggest that gains be adequate to allow dynamic simulation of the system to occur. Without engaging in a philosophical discussion as to whether fault trees represent a symbolic as opposed to dynamic simulation of a system, I would point out that the more complex approaches of Lapp and Powers (1977) and Kumamoto et al. (1981) have several disadvantages. To allow the magnitude of the gain to be assigned, by stating that should a disturbance with a gain of ±10 enter a loop then that disturbance cannot be handled by that loop is superficially attractive in that it does not require the analyst to inquire as to the causes of loop failure. However, a simple example concerning a reactor (Figure 6a) is sufficient to demonstrate a problem with this approach. In Figure 6b we see an attempt to draw a simplified digraph representing the causes of reactor overpressure where a temperature control loop, unlike the relief valve and the addition of inert material, is unable to compensate for the introduction of large quantities of contaminant into the reactor. The digraph shown in Figure 6b is, however, in error as it falsely concludes that both the temperature control loop and the addition of inert material are not effective if large quantities of contaminant are introduced. Indeed, it is difficult to see how it should be drawn when multiple control loops of differing capabilities are provided to handle disturbances. Using the approach suggested by Allen and Rao (1980), we can simply draw the digraph shown in Figure 6c, defining the presence of a large quantity of contaminant as a cause for the temperature control loop being inadequate.

The problem posed by assigning exact values to the gains as proposed by Kumamoto et al. (1981) is that such an approach would require that detailed design information be available. This would obviously preclude the use of digraphs to facilitate fault tree analysis in the early design stages when its use to assess major safety and reliability concerns is of most importance. It would also make the preparation of the digraph a far more arduous task.

A practical problem that is often addressed in fault tree analysis is the handling of sequential systems or "phased missions." Ziehms (1974) presented a fault tree for a phased mission that comprised subtrees specifically devoted to each phase, and subsequent analysts have tended to follow this approach. Despite Andow's (1980) comments that to use fault trees for the analysis of phased missions is to mismatch problem and technique, in practice few problems are encountered when partial digraphs or subtrees are prepared for each time phase (Schaeiwitz et al., 1977), thereby avoiding the possibility of contradictory events occurring simultaneously. In general, all that is required is a careful definition of the timing of certain events. As Fussell (1981) pointed out in his study of phase missions, two types of failures can occur in each phase: transition failures in which a cause of system failure that had "slept" through previous phases becomes active as a new phase is initiated, and failures that subsequently occur in the course of that phase. By carefully defining the timing of each failure, both these failures can easily be represented in a digraph. The careful definition of the timing of events also facilitates the assignment of failure and repair probabilities to events. In drawing digraphs

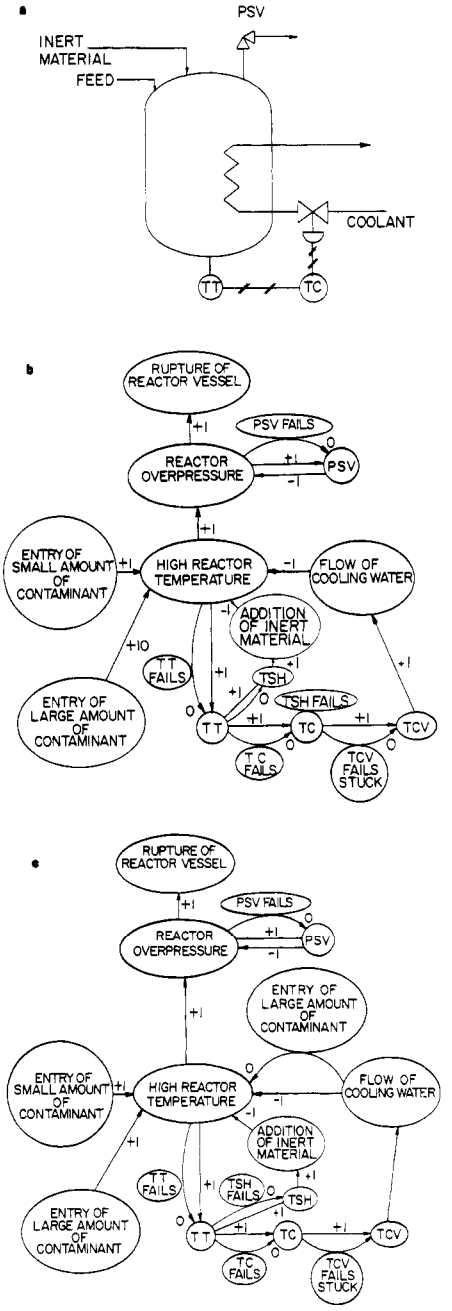

Figure 6. (a) Reactor. (b) Simplified digraph of reactor. (c) Digraph following Allen and Rao.

for phased missions, it is also important to ensure that the possibility of repair of "sleeping" failures is adequately represented. Where appropriate, this can be done through the introduction of nodes explicitly stating no repairs are effected or by ensuring that repair data introduced in a quantitative analysis reflect this requirement.

The Synthesis of Fault Trees from Digraphs

The algorithm we use to synthesize fault trees from digraphs is complex. In essence it comprises four steps: the editing of the digraph, the identification of feedback and negative feedforward loops, the synthesis of a tree, and the editing of this tree. A feature of this algorithm that will be noted in the examples presented in this paper is that it examines disturbances to loops one at a time. Although the algorithm has been described elsewhere (Allen and Rao, 1980), it would be useful here to expand upon several aspects of this algorithm, in particular upon the way in which it handles cascaded negative feedback loops and negative feedforward loops.

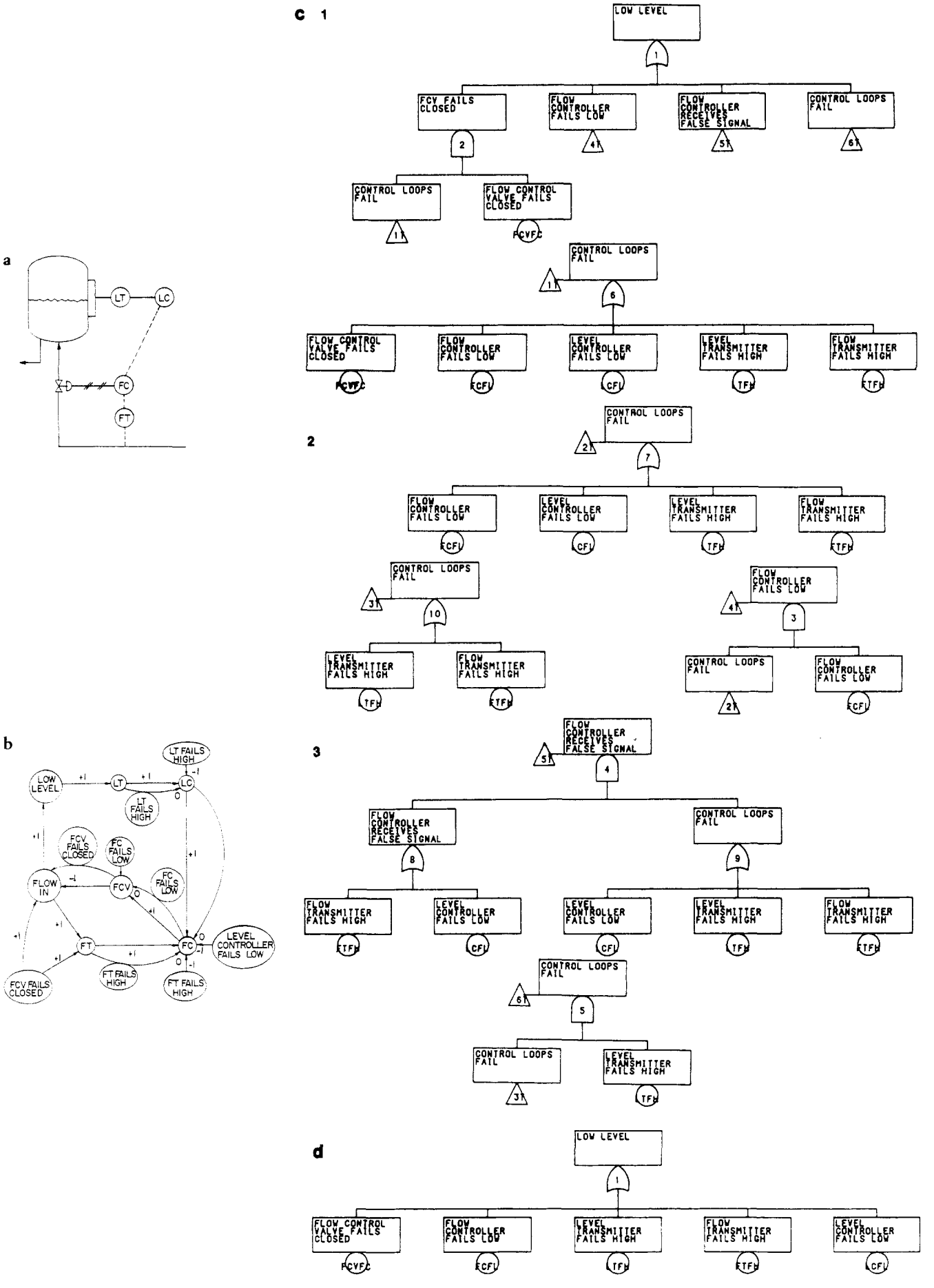

Figure 7. (a) Boiler feedwater system. (b) Digraph of boiler feedwater system. (c) Fault tree. (d) Simplified fault tree.

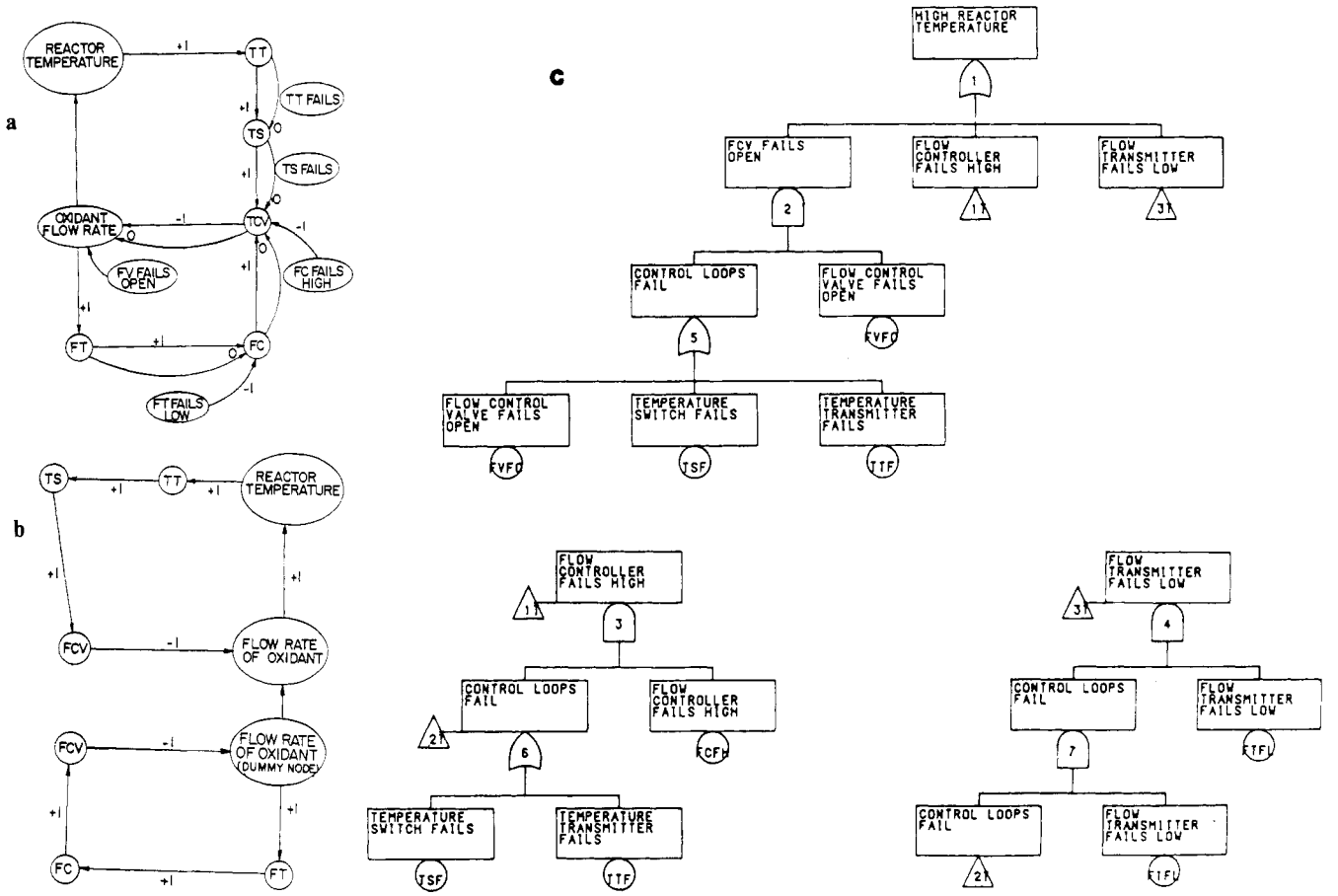

Figure 8. (a) Digraph showing cascaded loops. (b) Equivalent digraph. (c) Fault tree.

Cascaded loops are commonplace. One such loop that is frequently encountered is that used to ensure the smooth delivery of boiler feedwater to a steam generator (Figure 7a). Here a level control loop resets the set point of a flow-control loop thus ensuring that a constant level is maintained within the steam generator without encountering the rapid fluctuations in valve position that might be encountered if the position of the flow control valve were directly determined by the level controller. A simple digraph for this system can be drawn (Figure 7b) and a fault tree created (Figure 7c) by adding the failures of the flow control loop (flow in – FT – FC – FCV – flow in) to those of the level control loop (level – LT – LC – FC – FCV – flow in – level). This fault tree can be greatly simplified (Figure 7d) before analysis.

By this means, therefore, the class of cascaded loops in which failures in inner loops lead to the failure of all loops can be addressed. However, this representation is not always desired, and if our algorithm for the synthesis of fault trees is to be correct, we need to be able to handle other cases to avoid encountering additional problems. For example, we need to be able to handle cascaded loops that in essence function independently. Again this is best described with an example: The digraph shown in Figure 8 depicts a situation in which excessive quantities of an oxidant entering a reactor leads to excessive temperature in the reactor. To guard against this, two negative feedback control loops are incorporated in the system. The first (flow rate – FT – FC – FCV – flow rate) controls the flow of oxidant at a preset level. The second (temperature – TT – TS – FCV – flow rate – temperature), activated upon the occurrence of a high reactor temperature, shuts off the flow of oxidant. These loops are cascaded; however, it would be erroneous to assume that, for example, if the flow transmitter were to fail low, an excessive temperature would result. In reality, the second control loop would also need to fail. However, given a failure in the second control loop, a failure in the flow transmitter could lead to an excessive temperature. Thus we cannot simply disregard the first inner loop as Andow (1980) would suggest. Instead, we need to devise an algorithm that would allow us to handle cascaded loops as though they are independent of each other—in essence to treat the digraph presented in Figure 8a as the digraph depicted in Figure 8b. The resultant fault tree is presented in Figure 8c. This can of course be readily incorporated in a computer program. However, I know of no way in which the algorithm can, a priori, distinguish between those cascaded loops in which the failure of inner loops can lead to the failure of all loops and those cascaded loops in which this is not the case; such a decision must, I believe, rest with the analyst.

A second problem is less tractable. This problem concerns negative feedforward loops. Our present algorithm assumes that where multiple feedforward loops originate on a single node with each loop having in common the arm representing the propagation of the disturbance through the system, then all the negative feedforward loops must fail for the disturbance to propagate. This may not always be the case, however. For example, consider a case in which two gas streams enter a vessel with a single exhaust stream (Figure 9a). Should the exhaust fan fail, then the inlet fans should shut down to prevent excessive pressure. Should the digraph for the failure of this pressure balancing system be drawn as in Figure 9b, we see that there are two negative feedforward loops originating on the node "exhaust fan fails" and that the failure of either loop can lead to excess pressure in the vessel. One solution to this problem is to redraw the digraph, a task that is made easier if the digraph is treated as an information flow diagram into which all information possessed by the analyst is

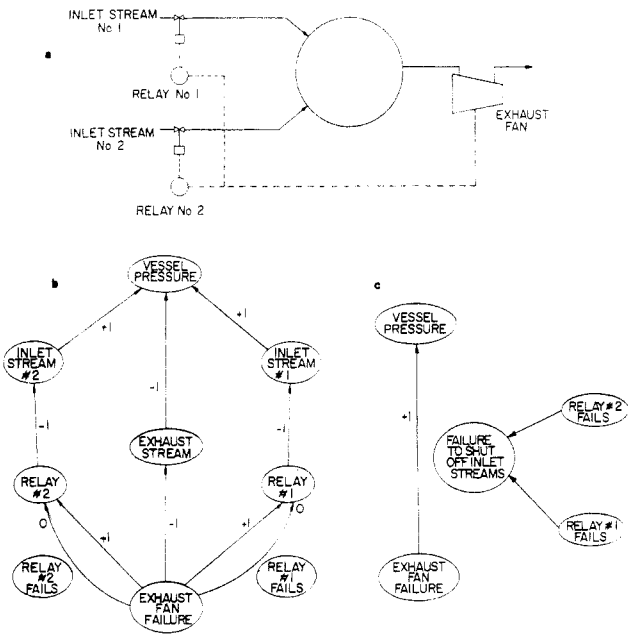

Figure 9. (a) Pressure balancing system. (b) Digraph with negative feedforward loops. (c) Revised digraph.

inserted rather than being handled as a purely mechanical exercise. Adopting this approach, Figure 9c can be drawn. While this digraph reflects reality, it must be admitted that it does so in a manner that circumscribes the use of digraphs.

The Complete Analysis

Traditionally the results of a fault tree analysis are presented in the form of drawings of the fault tree and a list of the principal causes (cut-sets) of the event of interest. It is possible, however, to bypass the time-consuming and expensive task of drawing and reviewing fault trees by using a series of computer programs to jump directly from digraph to cut-sets. The cut-sets are then examined in conjunction with the digraphs to determine the accuracy and consistency of the cut-sets (i.e., to ensure that cut-sets are indeed sets of sufficient conditions for the event of interest to occur). Should a cut-set not be valid, then the digraph needs to be corrected.

I believe that with this approach we greatly facilitate the task of identifying the causes of hazard occurrence, without impairing the adequacy of the analysis, particularly in large and complex problems involving thousands of events and hundreds of control loops. However, it must be admitted that this belief is contentious.

Acknowledgment

The author wishes to acknowledge C. Dunglinson and W. Tilton of DuPont for bringing many problems to his attention and for many stimulating discussions on the use of digraphs. He also wishes to thank the editor and referees for their helpful comments.

Literature Cited

Allen, D. J.; Rao, M. S. M. *Ind. Eng. Chem. Fundam.* **1980**, *19*, 79.
Andow, P. K. *IEEE Trans. Reliab.* **1980**, *29*, 2.
Caceres, S.; Henley, E. J. *Ind. Eng. Chem. Fundam.* **1976**, *15*, 128.
Chu, B. B. "A Computer-Oriented Approach to Fault Tree Construction", prepared for the Electric Power Research Institute, Report No. NP-288, EPRI Research Project 297-18, Palo Alto, CA, 1976.
Fussell, J. B. *Nucl. Sci. Eng.* **1973**, *52*, 421.
Fussell, J. B. "Phased-Mission System Rellability Analysis", prepared for the Electric Power Research Institute, Report No. NP-1945, EPRI Research Project 1233-2, Palo Alto, CA, 1981.
Kumamoto, H.; Henley, E. J.; Inoue, K. *IEEE Trans. Reliab.* **1981**, *30*, 110.
Lapp, S. A.; Powers, G. J. *IEEE Trans. Reliab.* **1977**, *26*, 2.
Nehem, R. F. "Gert-Graphical Evaluation and Review Technique, A Quantitative Hazard Analysis Tool", USAMC-ITC, Repot No. 3-73-12, 1973.
Schaeiwitz, J. A.; Lapp, S. A.; Powers, G. J. *Ind. Eng. Chem. Process Des. Dev.* **1977**, *16*, 529.
Ziehms, H. Ph.D. Dissertation, Naval Postgraduate School, Dec 1974 (as reported in Fussell, 1981).

Received for review November 3, 1982
Revised manuscript received October 5, 1983
Accepted December 6, 1983

Diffusion and Reaction in a Char Particle and in the Surrounding Gas Phase. Two Limiting Models

Stratis V. Sotirchos[†] and Neal R. Amundson*
University of Houston, Houston, Texas 77004

Two models describing reaction and diffusion in the boundary layer and in the interior of a porous char particle are developed. Both models take into account the occurrence of the heterogeneous combustion and gasification reactions in the interior of the particle, but the rate of the homogeneous oxidation of carbon monoxide is assumed to be either infinitely slow or infinitely fast. The models are used chiefly to study the effects of intraparticle thermal gradients on the solution structure in conjunction with the intraparticle diffusional limitations. It appears that consideration of thermal gradients influences the solution drastically, especially whenever multiple solutions arise. In addition, comparison of the two models reveals once again the strong effect of the homogeneous reaction on the solution structure.

1. Introduction

The problem under consideration is the burning of a char particle exposed to an oxygen-containing environment. The combustion process involves diffusion and reaction in the interior of the particle and in the boundary layer. Oxygen diffuses through the boundary layer and the pores of the particle and reacts with the carbon producing CO and CO_2. The carbon monoxide reacts in the gas phase with the oxygen to form more CO_2 which in turn reacts with the carbon to form CO. The ratio of the

[†]Department of Chemical Engineering, University of Rochester, Rochester, NY 14627.

0196-4313/84/1023-0180$01.50/0