PROCESS DESIGN AND CONTROL

Studies on the Digraph-Based Approach for Fault-Tree Synthesis. 2. The Trip Systems

Chuei-Tin Chang,* Ding-Shang Hsu, and Der-Ming Hwang

Department of Chemical Engineering, National Cheng Kung University,
Tainan, Taiwan 70101, Republic of China

The digraph-based fault-tree synthesis procedures for feedforward and feedback trip systems are presented in this paper. The proposed techniques are suitable for assessing risks associated with *all* possible top events in a unified framework. This feature is potentially useful in automating the fault-tree construction process and thus represents an improvement of the conventional method. Two application examples are also provided to demonstrate the feasibility of our approach.

Introduction

It is a common design practice in situations where a hazardous condition may arise in a plant to provide some form of automatic protective system. One of the principal types of such systems is a trip, which closes or opens a valve (or switch) if a fault is detected during operation. Presently, the decision as to whether a trip is necessary in a given operation depends largely on the design philosophy. As a result, there are quite wide variations in practice in the use of trips. This decision, however, can be put on a less subjective basis by carrying out a rigorous fault-tree analysis to assess the risk of all possible accidents quantitatively.

There are basically two types of equipment malfunctions associated with the protective systems. Specifically, a trip must be reliable against the *functional failures*, i.e., failures which prevent the actuation of a trip when a process upset occurs, and also the *operational failures*, i.e., failures which initiate the trip action when no hazardous condition exists (Lees, 1980). It is therefore in our interest to evaluate the risk associated with both loss of protection against the process upset caused by the former failure and plant shutdown due to the latter reason, i.e., spurious trip. Furthermore, it is perhaps equally important to address the unexpected problems caused by *normal* activation of the trip. This is due to the fact that the trip action usually causes the plant to be operated in a mode which is unfamiliar even to an experienced engineer. Thus, to ensure the comprehensiveness of the risk analysis, it is essential to construct several separate fault trees using outcomes of the above three types of scenarios as the top events.

In previous publications (Chang and Hwang, 1992; Chang *et al.*, 1993), the authors proposed a fault-tree synthesis algorithm which is quite effective in many realistic applications. This algorithm is essentially an improved version of the popular digraph-based method (Lapp and Powers, 1977; Andrews and Brennan, 1990). In particular, generalized fault-tree structures (operators) corresponding to various digraph configurations, i.e., tree, feedforward loop (FFL) and feedback loop (FBL), were developed for systems with coupled control and *process* loops. In order to facilitate later discussions, they are repeated in Appendix A of this paper and, also, the associated terminologies (i.e., the definitions of faults and failures) are provided in Appendix B. It should be emphasized that, since these operators are general, the corresponding fault-tree construction procedure can be computerized easily. The resulting generic software is suitable for risk assessment for a large number of practical systems (Hwang, 1992).

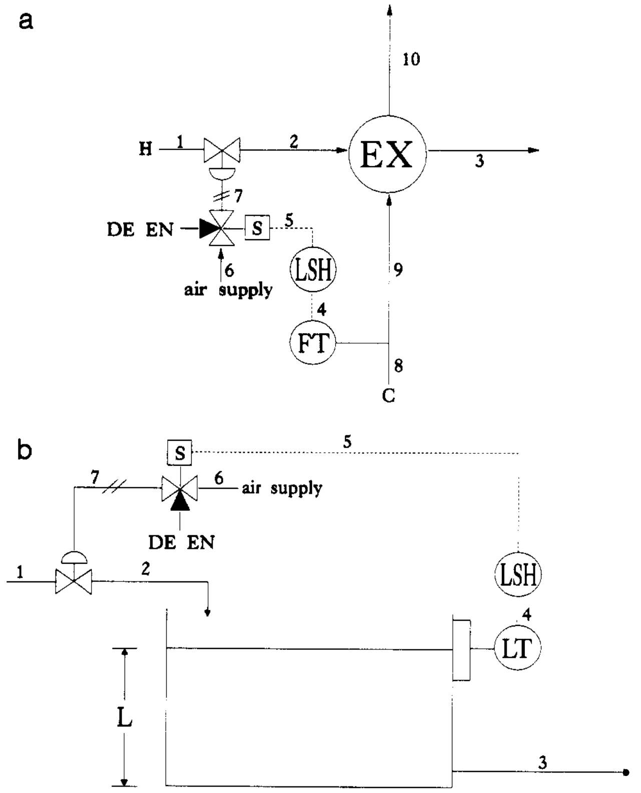

Figure 1. Simplified flowsheet of (a) a heat exchange system with feedforward trip and (b) a liquid storage system with feedback trip.

Since any trip system can be viewed as a special type of control equipment, the corresponding diagraph must also contain a negative feedforward loop (NFFL) or a negative feedback loop (NFBL). A direct application of the procedure suggested by Chang and Hwang (1992), however, fails to produce correct results. This is due to

* Author to whom correspondence should be addressed. E-MAIL: nckut047@twnmoe10.bitnet.

0888-5885/94/2633-1700$04.50/0

the fact that, although the diagraph structures of trip systems are similar to those of the regular control loops, the corresponding control logic is not the same. On the other hand, it should be noted that Lees and his co-workers (Hunt *et al.*, 1992) have already developed useful fault-tree construction techniques for trip systems with a different approach in modeling, i.e., the mini-fault trees. However, in applying their method, separate models have to be adopted to analyze the functional and operational failures associated with the same system. As a result, the corresponding computer code is inefficient. Thus, to improve the automatic fault-tree construction algorithm, it is our intention in this work to develop of a unified diagraph-based fault-tree synthesis procedure which is suitable for *all* possible top events in a given trip system.

Diagraph Structures of Trip Systems

Generally speaking, there are only two basic diagraph structures that can be identified in various trip systems (Hunt *et al.*, 1992), i.e., the NFFL and the NFBL. In this work, the term "feedforward loop" (FFL) is used to represent a digraph configuration in which two or more paths start from a common node and converge at another node, and "feedback loop" (FBL) is a path which starts and ends at the same node. The negative feedforward loop (NFFL) is a special type of FFL in which the signs of the products of edge gains along different paths are not the same. Similarly, a negative feedback loop (NFBL) is also a FBL on which the product of the edge gains around the loop is negative.

Typical examples of feedforward and feedback trip systems are presented in Figure 1 parts a and b, respectively. The trip in the first example is designed to guard against loss of cooling in a heat exchanger EX (Figure 1a). Specifically, the switch LSH is activated when the cold-stream flow rate is lower than a given value. Once the switch is triggered, the solonoid valve will be set at the de-energized position by a signal from the switch. Consequently, the instrument air in line 7 will be vented and, then, the air-to-open control valve on the inlet pipeline will be closed completely. The second example is concerned with a liquid storage tank (Figure 1b). The trip in this case is used to protect against overfill, i.e., the switch LSH is designed to be activated when the liquid level exceeds a given limit. Notice that, when compared with a motor-operated valve, the design of these two trip systems, i.e., the combination of the solonoid valve and control valve, is actually a more expensive alternative. The present design is chosen on the ground that important features of the proposed fault-tree synthesis procedures can be demonstrated more clearly.

The system digraphs of the above two examples can be found in Figure 2 parts a and b, respectively. The physical meanings of the symbols used in these diagraphs are explained in the Nomenclature section at the end of this paper. In Figure 2a, two NFFLs can be identified , i.e.,

$$\left\{\begin{array}{l} m8 \xrightarrow{+1} m9 \xrightarrow{-1} t3 \\ m8 \xrightarrow{+1} s4 \xrightarrow{-1} s5 \xrightarrow{-1} s7 \xrightarrow{+1} m2 \xrightarrow{+1} t3 \end{array}\right\} \tag{1}$$

and

$$\left\{\begin{array}{l} m8 \xrightarrow{+1} m9 \xrightarrow{-1} t10 \\ m8 \xrightarrow{+1} s4 \xrightarrow{-1} s5 \xrightarrow{-1} s7 \xrightarrow{+1} m2 \xrightarrow{+1} t10 \end{array}\right\} \tag{2}$$

On the other hand, one can observe from Figure 2b that the path

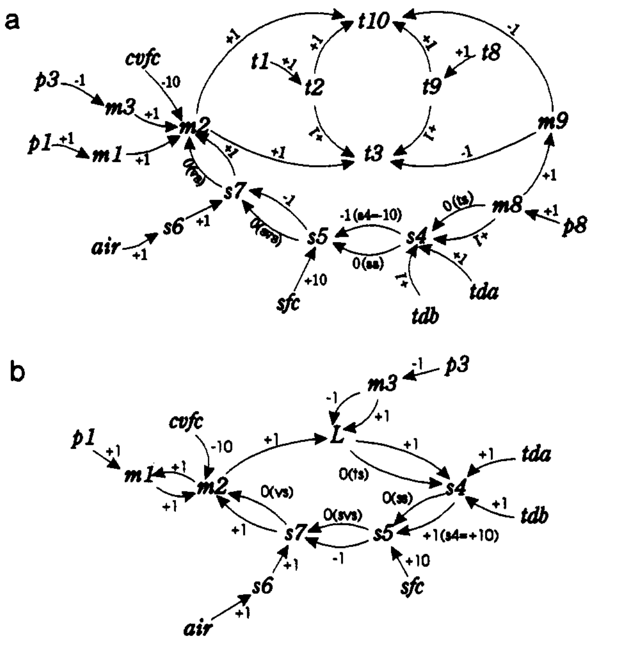

Figure 2. System digraph of (a) the heat exchange system and (b) the liquid storage system.

$$L \xrightarrow{+1} s4 \xrightarrow{+1} s5 \xrightarrow{-1} s7 \xrightarrow{+1} m2 \xrightarrow{+1} L \tag{3}$$

forms a NFBL. Notice that, in both examples, only a portion of the disturbances entering the trip system can be eliminated. For instance, the protective system of the heat exchanger in Figure 1a may be activated by a large reduction in the flow rate of cooling water but remains inactive when the same flow is increased. Similarly, the trip of the storage system in Figure 1b can only be triggered by a drastic increase in the liquid level. This special nature is reflected by the conditional edges between the sensor signal ($s4$) and the signal from the switch ($s5$) respectively in Figure 2 parts a and b. Thus, in both cases, $s5$ is only allowed to have two values, i.e., 0 (the switch is inactive) and 10 (the switch is triggered).

Selection of the Top Events

As mentioned before, to ensure comprehensiveness, it is often necessary to obtain several separate fault trees corresponding to all undesirable events in a given trip system. By definition, if a fault tree is to be used for evaluating the risk associated with operational failures, then the direct outcome of trip action should be selected as the top event. For example, the event "$m2$(−10)" (the outlet flow of the trip valve stops) can be chosen for the two systems presented in Figure 1 parts a and b. Further, it should also be noted that this fault tree must be constructed under the assumption that all normal trip-activation scenarios are unallowed.

For the functional failures, the top event should be chosen as the hazardous condition agianst which the trip is designed to protect. Alternatively, it can also be identified on the basis of the digraph configuration. In a feedforward trip system, the top event is associated with the ending node of the NFFL, e.g., "$t3$(+10)" in the heat exchanger example (see Figures 1a and 2a). On the other hand, the top event for a feedback trip system should be the process upset associated with the measured variable, e.g., "L(+10)" in the liquid storage example (see Figures 1b and 2b). Notice that the value +10 is used in this study to indicate there is a large deviation from the normal level.

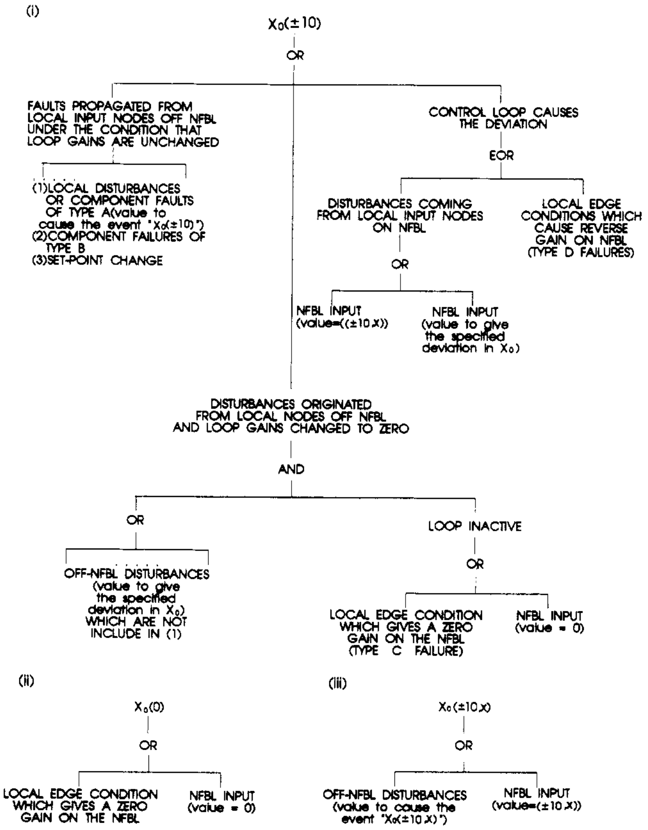
(i)
$X_0(\pm10)$
OR
FAULTS PROPAGATED FROM LOCAL INPUT NODES OFF NFBL UNDER THE CONDITION THAT LOOP GAINS ARE UNCHANGED
(1)LOCAL DISTURBANCES OR COMPONENT FAULTS OF TYPE A(value to cause the event "$X_0(\pm10)$")
(2)COMPONENT FAILURES OF TYPE B
(3)SET-POINT CHANGE
CONTROL LOOP CAUSES THE DEVIATION
EOR
DISTURBANCES COMING FROM LOCAL INPUT NODES ON NFBL
OR
NFBL INPUT (value=((±10,x))
NFBL INPUT (value to give the specified deviation in X_0)
LOCAL EDGE CONDITIONS WHICH CAUSE REVERSE GAIN ON NFBL (TYPE D FAILURES)
DISTURBANCES ORIGINATED FROM LOCAL NODES OFF NFBL AND LOOP GAINS CHANGED TO ZERO
AND
OR
OFF-NFBL DISTURBANCES (value to give the specified deviation in X_0) WHICH ARE NOT INCLUDE IN (1)
LOOP INACTIVE
OR
LOCAL EDGE CONDITION WHICH GIVES A ZERO GAIN ON THE NFBL (TYPE C FAILURE)
NFBL INPUT (value = 0)
(ii)
$X_0(0)$
OR
LOCAL EDGE CONDITION WHICH GIVES A ZERO GAIN ON THE NFBL
NFBL INPUT (value = 0)
(iii)
$X_0(\pm10,x)$
OR
OFF-NFBL DISTURBANCES (value to cause the event "$X_0(\pm10,x)$")
NFBL INPUT (value=(±10,x))

Figure 3. Generalized fault-tree structure of feedback trip systems: (i) substructure A, (ii) substructure B, and (iii) substructure C.

A change of magnitude of 1 is assumed to be not enough to trigger the trip system.

Finally, other undesirable events should also be adopted as the top events. They must be determined, however, on a case-by-case basis according to process knowledge specific to the given problem.

Modified Fault-Tree Structures for Trip Systems

Since the control logic of a trip system is different from that of a regular control loop, the fault propagation behavior should also be different. In this study, qualitative steady-state analysis similar to that described in the previous paper (Chang and Hwang, 1994) has been performed for both feedforward and feedback trip systems. For the sake of the conciseness of this paper, detailed results of this analysis are not elaborated here. Instead, the resulting conclusions are summarized in two generalized fault tree structures:

Feedforward Systems. In this case, available techniques, i.e., structure II in Appendix A, can be adopted with only minor modifications in implementing the fault-tree structure. Specifically, in selecting the inputs to substructures IIA and IIB–2, two different types of possible scenarios, classified on the basis of whether or not they can be attributed to events that trigger the trip action, must both be considered.

Feedback Systems. The generalized fault-tree structures for the feedback trip systems are presented in Figure 3, i.e., substructures A, B, and C. One can observe that they are actually very similar to those for a regular control NFBL, i.e., substructures IIIA, IIIB, and IIIC (see Appendix A). There are basically two unique features that are different in using substructure A. First, in choosing the inputs to the left branch of this substructure, the entries under (1) should include both trip-triggering and non-trip-triggering events. Also, notice that some of the values of "NFBL inputs" in the right branch are represented by a newly created symbol (±10,x). This special representation can be interpreted as the state of a variable which would have a value of +10 (or −10) without feedback but reaches a different state x eventually due to the trip action. In applying substructure A, the existence of an input with value (±10,x) should only be considered when it is associated with the sensor output and X_0 is a variable representing a signal from the switch.

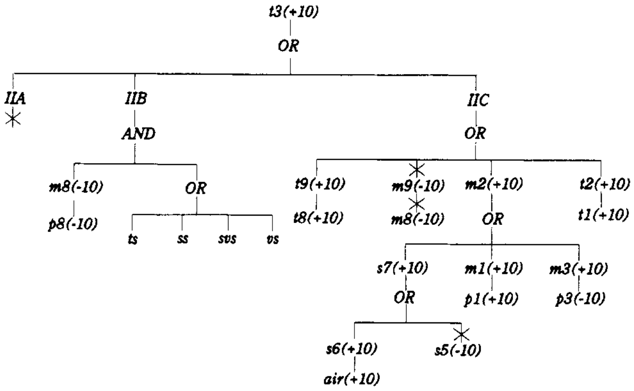

Figure 4. Fault tree associated with the consequence of functional failures in the heat exchange system.

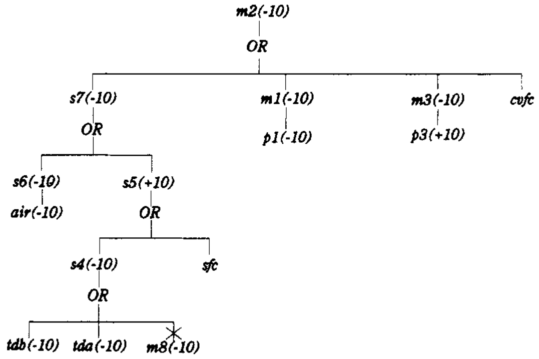

Figure 5. Fault tree associated with the consequence of operational failures in the heat exchange system.

The other two substructures are rather simple. Substructure B is essentially the same as the existing operator for "$X_0(0)$," i.e., substructure IIIB. Substructure C is meant to be used for developing the nonbasic event "X_0(±10,x)" in a fault tree.

The final value of a loop variable (x) in the representation (±10,x) can be determined by an analysis of the results of qualitative simulation (Chang and Hwang, 1992; Chang and Hwang, 1994) on a case-by-case basis. In most cases, this value is nonzero. One of the implications of this fact is that abnormal disturbances can propagate in a trip system even when the trip functions properly. The process variables may be affected not only by constant changes (±10) in the loop variables but also by changes represented by the symbol (±10,x). Thus, in considering the causes of any abnormal condition in a feedback trip system, both possibilities must be included.

Application Examples

To demonstrate that the proposed techniques can be applied in a consistent fashion for constructing fault trees associated with all possible top events in a trip system, the following two examples are presented:

Example 1. The first example is concerned with the feedforward trip system described in Figures 1a and 2a.

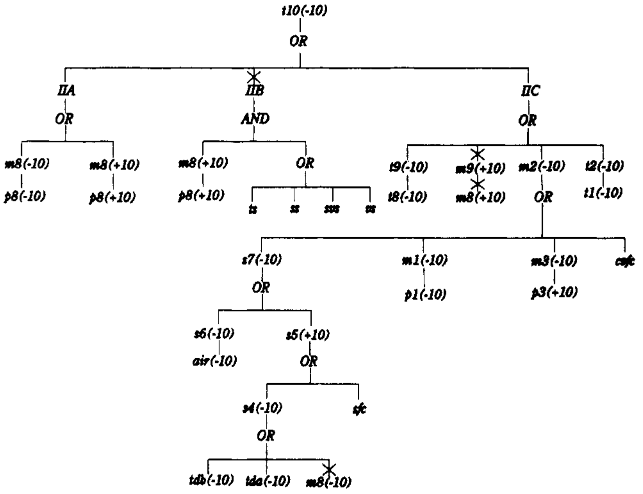

Figure 6. Fault tree associated with the undesirable event "*t*10(−10)" in the heat exchange system.

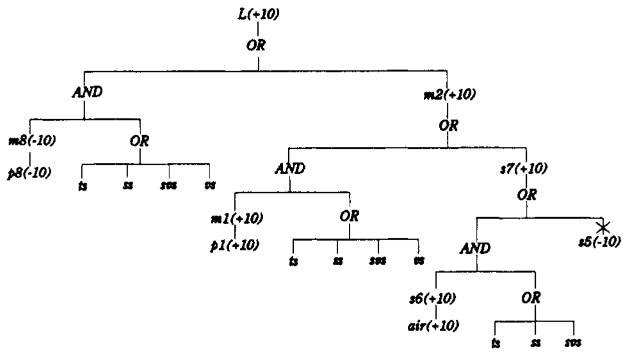

Figure 7. Fault tree associated with the consequence of functional failures in the liquid storage system.

In addition to the fault trees needed for identifying the hazards of functional and operational failures, it is assumed in our study that a large decrase in the outlet temperatures of the cold stream, i.e., "*t*10(−10)," is also undesirable. Three fault trees have thus been constructed accordingly.

1. The fault tree associated with the consequence of functional failures: By following the principles outlined in the previous section for implementing structure II, the fault tree corresponding to the top event "*t*3(+10)" can be constructed (Figure 4). Notice that *s*5 can only assume non-negative values. Thus, the fault tree cannot be developed further when the event "*s*5(−10)" is reached.

It can be observed from Figure 4 that there are no suitable entries under IIA in this trip system. Under IIB, several causes of the top event which involve functional failures can be identified:

{*p8*(−10),*ts*}, {*p*8(−10),*ss*}, {*p*8(−10), *svs*}, {*p*8(−10),*vs*}

On the other hand, the cut sets found under IIC, i.e.,

{*t*1(+10)}, {*t*8(+10)}, {*p*1(+10)}, {*p*3(−10)}, {*air*(+10)}

cannot be regarded as functional failures since the trip is not supposed to be activated in these cases. They are, however, correct causes of the top event and thus should be included in this fault tree. As a matter of fact, it is rather important to identify such causes of a hazardous condition in assessing the integrity of a given feedforward trip system.

2. The fault tree associated with the consequence of operational failures: As indicated previously, this fault tree should be constructed with the top event "*m*2(−10)" and, also, under the assumption that the trip-triggering event (i.e., loss of cooling) does not occur. Since *m*2 is a node on the path of a NFFL and not an ending node (see Figure 2a), the fault-tree structure corresponding to a simple tree-like digraph, i.e., structure I in Appendix A, should be applicable in this situation. The resulting fault tree is presented in Figure 5. Notice that the unallowed trip-triggering event "*m*8(−10)" is severed from the fault tree. The corresponding minimum cut sets can be divided into two types:

{*cvfc*}, {*sfc*}, {*air*(−10)}, {*tda*(+10)}, {*tdb*(+10)}

{*p*1(−10)}, {*p*3(+10)}

The first five events are the causes of spurious system shutdown and thus can be regarded as the operational failures. Here, "*cvfc*" and "*sfc*" represent the failures associated with the control valve and the switch, respectively. The cut set {*air*(−10)} indicates that a drastic decrease in air supply pressure may also cause the trip valve to be closed without affecting the cold-stream flow rate. Two different sensor failures are included in this fault tree, i.e., "*tda*(+10)" and "*tdb*(+10)." "*tdb*(+10)"

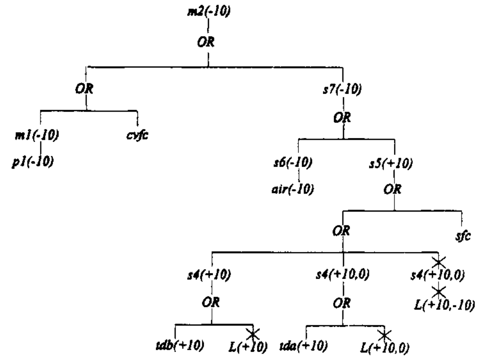

Figure 8. Fault tree associated with the consequence of operational failures in the liquid storage system.

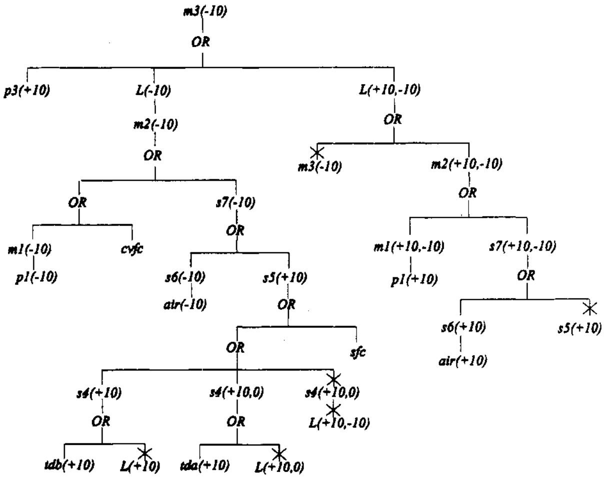

Figure 9. Fault tree associated with the undesirable event "*m*3(−10)" in the liquid storage system.

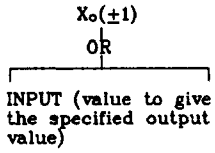

Figure 10. Generalized fault-tree structure for digraphs with the configuration of a tree (structure I).

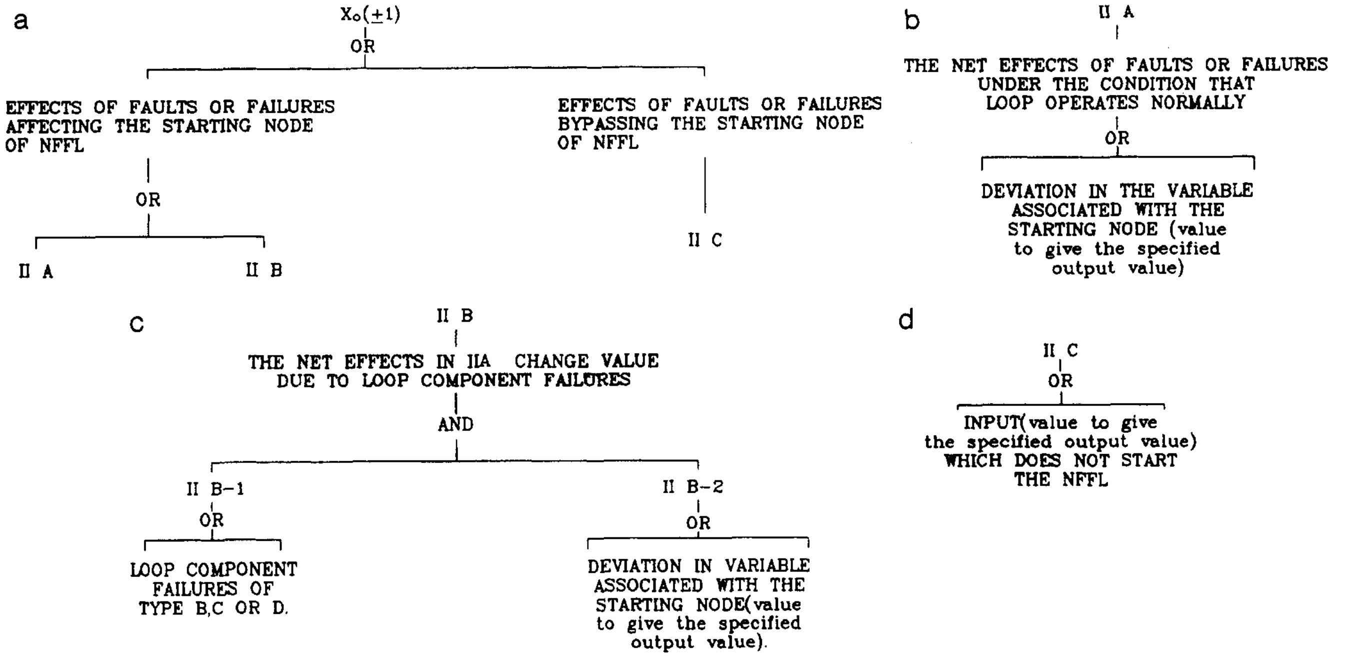
a
X₀(±1)
OR
EFFECTS OF FAULTS OR FAILURES AFFECTING THE STARTING NODE OF NFFL
EFFECTS OF FAULTS OR FAILURES BYPASSING THE STARTING NODE OF NFFL
OR
II A
II B
II C
b
II A
THE NET EFFECTS OF FAULTS OR FAILURES UNDER THE CONDITION THAT LOOP OPERATES NORMALLY
OR
DEVIATION IN THE VARIABLE ASSOCIATED WITH THE STARTING NODE (value to give the specified output value)
c
II B
THE NET EFFECTS IN IIA CHANGE VALUE DUE TO LOOP COMPONENT FAILURES
AND
II B-1
OR
LOOP COMPONENT FAILURES OF TYPE B,C OR D.
II B-2
OR
DEVIATION IN VARIABLE ASSOCIATED WITH THE STARTING NODE(value to give the specified output value).
d
II C
OR
INPUT(value to give the specified output value) WHICH DOES NOT START THE NFFL

Figure 11. Generalized fault-tree structure for the variable associated with the end node of a NFFL: (a) structure II, (b) substructure IIA, (c) substructure IIB, and (d) substructure IIC.

represents the event that the flow sensor fails high, i.e., a type B failure (see Appendix B). On the other hand, "*tda*(+10)" denotes a type A failure, e.g., a drift in the zero of the flow sensor. These two different types of sensor failures are included in this paper mainly to demonstrate the use of the modified versions of structure III for NFBLs, which will be detailed later in the next example.

Strictly speaking, the second group of events do not involve operational failures. The event "*p*1(−10)" may be caused by a pump failure upstream which is not associated with any of the trip components. One of the outcomes of such an event is a large decrease in the flow of the hot stream *H*, i.e., "*m*2(−10)," which happens to be the consequence of the operational failures. Similarly, a large increase in the *p*3, which may be the result of a blockage in pipeline No. 3, can also stop the hot-stream flow. In addition, notice that none of the above two events "*p*1(−10)" and "*p*3(+10)" result in the hazardous condition, i.e., loss of cooling. Thus, they are included in the fault tree associated with the "operational failures" mainly on the ground that such events also satisfy the implied assumptions, i.e., they do not cause the normal trip-activation scenarios but produce the same effect on *m*2.

3. The fault tree associated with other undesirable events: Since *t*10 is an ending node of a NFFL in Figure 2a, structure II is again applicable to the top event "*t*10(−10)." The resulting fault tree is presented in Figure 6. Notice that, in selecting the inputs to IIA, both "*m*8(−10)" and "*m*8(+10)" are appropriate. This is due to the fact that, although the former triggers the trip action while the latter does not, they produce the same effect on *t*10, i.e., a drop in the outlet temperature of cold stream.

In our study, substructure IIB is primarily designed to model the effects of the faults or failures that propagate through the starting node of a NFFL under the condition that a trip component fails simultaneously. However, since the event "*m*8(+10)" *alone* has already been identified as one of the causes of "*t*10(−10)" under IIA, the entire branch under IIB in this case must thus be deleted from the fault tree.

It should also be noted that one of the inputs to IIC, i.e., "*m*2(−10)," was adopted as the top event of the fault tree in Figure 5. Thus, all the minimum cut sets identified previously for assessing the risk associated with the operational failures can also be included as the cut sets of the present fault tree. In addition, other causes of the top event "*t*10(−10)" can be found under IIA and IIC, i.e.,

$$\{p8(+10)\}, \{t8(-10)\}, \{t1(-10)\}$$

$$\{p8(-10)\}$$

These four events are all process disturbances. The first three of them do not produce an effect that triggers the trip. The last one, on the other hand, does cause to trip to be actuated.

Finally, it should be emphasized that, although mutually exclusive events, i.e., "*p*8(−10)" and "*p*8(+10)," are included here as the causes of the same outcome "*t*10(−10)," their correctness can be easily verified by an analysis of the actual system behavior.

Example 2. The second example is concerned with the feedback trip system described in Figures 1b and 2b. Let us assume that, other than the undesirable consequences caused by functional and operational failures, there is a need to assess the risk associated with the event "*m*3(−10)" due to possible downstream problems. The fault trees corresponding to these scenarios are presented in the following.

1. The fault tree associated with the consequence of functional failures: By following the generalized structure presented in Figure 3, the fault tree corresponding to the top event "*L*(+10)" can be constructed (Figure 7). Again, the value of *s*5 must be non-negative. Thus, the fault tree cannot be developed further when the event "*s*5(−10)" is reached.

The minimum cut sets corresponding to this fault tree can be classified into two categories:

$$\{p1(+10),ts\}, \{p1(+10),vs\}, \{p1(+10),svs\}, \{p1(+10),ss\},$$
$$\{p3(+10),ts\}, \{p3(+10),vs\}, \{p3(+10),svs\}, \{p3(+10),ss\}$$

$$\{air(+10),ts\}, \{air(+10),svs\}, \{air(+10),ss\}$$

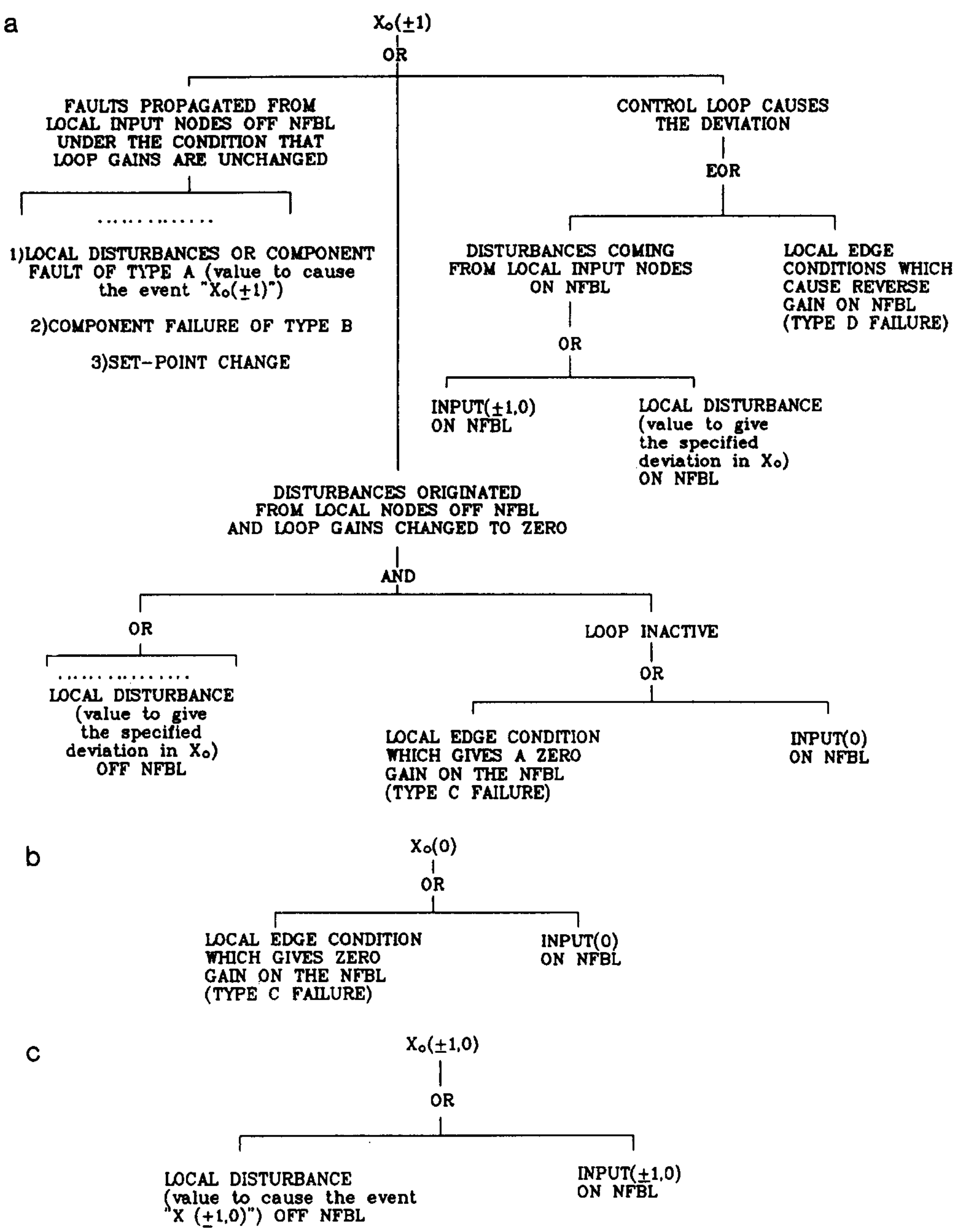
a
Xo(±1)
OR
FAULTS PROPAGATED FROM LOCAL INPUT NODES OFF NFBL UNDER THE CONDITION THAT LOOP GAINS ARE UNCHANGED
....
1)LOCAL DISTURBANCES OR COMPONENT FAULT OF TYPE A (value to cause the event "Xo(±1)")
2)COMPONENT FAILURE OF TYPE B
3)SET-POINT CHANGE
CONTROL LOOP CAUSES THE DEVIATION
EOR
DISTURBANCES COMING FROM LOCAL INPUT NODES ON NFBL
OR
INPUT(±1,0) ON NFBL
LOCAL DISTURBANCE (value to give the specified deviation in Xo) ON NFBL
LOCAL EDGE CONDITIONS WHICH CAUSE REVERSE GAIN ON NFBL (TYPE D FAILURE)
DISTURBANCES ORIGINATED FROM LOCAL NODES OFF NFBL AND LOOP GAINS CHANGED TO ZERO
AND
OR
....
LOCAL DISTURBANCE (value to give the specified deviation in Xo) OFF NFBL
LOOP INACTIVE
OR
LOCAL EDGE CONDITION WHICH GIVES A ZERO GAIN ON THE NFBL (TYPE C FAILURE)
INPUT(0) ON NFBL
b
Xo(0)
OR
LOCAL EDGE CONDITION WHICH GIVES ZERO GAIN ON THE NFBL (TYPE C FAILURE)
INPUT(0) ON NFBL
c
Xo(±1,0)
OR
LOCAL DISTURBANCE (value to cause the event "X (±1,0)") OFF NFBL
INPUT(±1,0) ON NFBL

Figure 12. (a) Generalized fault-tree structure used for a moderate deviation (±1) in the variable associated with a node on the NFBL (structure IIIA). (b) Generalized fault-tree structure for no deviation (0) in the variable associated with a node on the NFBL (structure IIIB). (c) Generalized fault-tree structure used for the deviation (±1,0) in the variable associated with a node on the NFBL (structure IIIC).

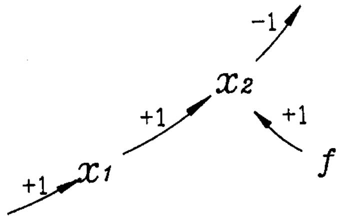

Figure 13. Digraph representation of a type A fault or failure.

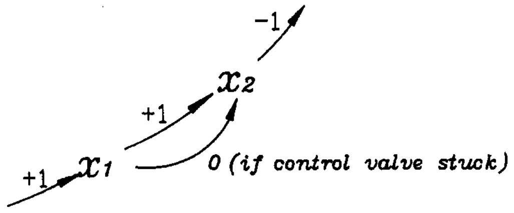

Figure 14. Digraph representation of a type C failure.

The first is basically the combination of a process disturbance from upstream ("$p1(+10)$") or downstream ("$p3(+10)$") and a functional failure in the trip system. The second type of cause is due to the simultaneous occurrence of a component failure and a surge in the instrument air pressure. Notice that, when compared with the feedforward trip systems, the feedback systems are superior in the sense that none of the causes of the top event can be attributed to events other than the functional failures.

2. The fault tree associated with the consequence of operational failures: This fault tree should be constructed with the top event "$m2(-10)$" and, also, under the assumption that the normal trip-triggering event (i.e., high liquid level) does not occur. Since $m2$ is a node on an NFBL (see Figure 2b), the generalized structure in Figure 3 is still applicable. The resulting fault tree is presented in Figure 8. Notice that the development of this tree is terminated wherever an unallowed event is reached, i.e., "$L(+10)$" or "$L(+10,x)$". The corresponding minimum cut

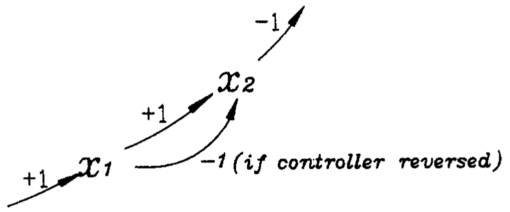

Figure 15. Digraph representation of a type D failure.

sets are basically the same as those identified from the fault tree in Figure 5 for Example 1, i.e.,

$$\{cffc\}, \{sfc\}, \{air(-10)\}, \{tda(+10)\}, \{tdb(+10)\}$$

$$\{p1(-10)\}$$

The first five events can be considered as the operational failures. Their respective significance has already been discussed and thus will not be repeated here. Notice that, since "*tda*(+10)" is a type A sensor failure, the sensor output *s*4 can be assumed to be still affected by the liquid level if it occurs. Thus, *s*4 should reach the normal value eventually after the trip is activated, i.e., "*s*4(+10,0)" should be the result of "*tda*(+10)". The occurrence probability of such a series of events is certainly very low. They are included in this paper to demonstrate the use of (±10,*x*) in the modified fault-tree structure.

Similar to results obtained from the fault tree presented in Figure 5, the second type of cut sets in this case is not associated with the operational failures either. They are included here on the basis of the same rationale, i.e., such events do not result in the normal trip-activation scenarios but produce the same outcome "*m*2(-10)".

3. The fault tree associated with other undesirable events: From the system digraph presented in Figure 2b, one can see that the variable *m*3 is corresponding to a node with two inputs, i.e., *L* and *p*3. Thus, the event "*m*3-(-10)" may be due to a large increase in the downstream pressure *p*3 (which causes the liquid level to rise and triggers the trip eventually) or a change in the liquid level *L* (which may be the result of some other events). Notice that *L* is a node on the NFBL associated with the trip system and, thus, two possible states of *L* should be considered, i.e., "*L*(-10)" and "*L*(+10,-10)". Here, "*L*(-10)" denotes a large decrease in the liquid level without activating switch LSH. "*L*(+10,-10)," on the other hand, represents a series of events, i.e., the liquid level first rises continuously until *L* = +10 (which triggers LSH) and, then, drops to a level significantly lower than the normal value (*L* = -10). On the basis of the above considerations, a fault tree can be produced according to the proposed general structure. The result is presented in Figure 9.

The minimum cut sets of this fault tree are very similar to those corresponding to the operational failures. Essentially, all the causes identified in the previous fault tree (Figure 8) can also be found in the present case. In addition, the following cut sets can be obtained:

$$\{air(+10)\}, \{p1(+10)\}, \{p3(10)\}$$

The three cut sets listed above are not operational failures. The events "*air*(+10)", "*p*1(+10)", and "*p*3(+10)" are actually process disturbances which are large enough to cause the switch to be triggered. In other words, the trip functions normally as expected in these cases. This result is correct since the outlet flow rate *m*3 must drop to zero after system shutdown.

Finally, it should also be noted that mutually exclusive events are included as the causes of the same outcome "*m*3(-10)", i.e., "*air*(-10)" vs "*air*(+10)" and "*p*1(-10)" vs "*p*1(+10)." Although this may appear unacceptable logically, the correctness of these results can be readily confirmed by simulating the actual system behavior.

From the results presented in the above two examples, one can observe clearly that it is indeed possible to construct a fault tree for *any* given top event using the unified framework suggested in this paper. On the other hand, with the conventional approaches, e.g., the mini-fault tree, the operational and functional failures must be treated differently using several templates. Also, the procedure for building fault trees associated with other undesirable events in the trip systems has not been developed explicitly in any of the previous works. Finally, notice that in order to assess the risk associated with functional failures, three separate fault trees have to be constructed first and a final tree can then be assembled from portions of the previous there (Hunt *et al.*, 1992). Thus, it is easy to see that the proposed method represents an improvement of the conventional approaches and the resulting software should be more efficient.

Conclusions

Modified fault-tree synthesis procedures for feedforward and feedback trip systems are presented in this paper. The results generated with the proposed techniques are suitable for assessing the risks associated with *all* possible undesirable events, not just the outcomes of the functional and operational failures. This feature is useful in automating the fault-tree synthesis process. *None* of the published approaches treated this problem in a unified framework such as the one suggested here.

Nomenclature

L = height of the liquid level in the storage tanks

*m*1–*m*3, *m*8–*m*10 = mass flow rate in pipeline nos. 1–3 and 8–10, respectively

*p*1, *p*3, *p*8 = pressure in pipeline nos. 1, 3, and 8, respectively

*s*4–*s*7 = signal in lines 4–7, respectively

sfc, *cvfc* = operational failures associated with the switch and the control valve, respectively, i.e., spurious signals are generated (type B failure)

*t*1–*t*3, *t*8–*t*10 = temperature in pipeline nos. 1–3 and 8–10, respectively

tda = drift in the zero of the flow or level sensor (type A failure)

tdb = flow or level sensor fails high (type B failure)

ts, *ss*, *vs*, *svs* = functional failures associated with the sensor, the switch, the control valve, and the solonoid valve, respectively, i.e., their outputs are unaffected by the inputs (type C failure)

Appendix A: Generalized Fault-Tree Structures for Three Diagraph Configurations

In a previous study, Chang and Hwant (1993) used three generalized structures to develop fault trees for three types of digraph configurations. They are presented in Figures 10–12 for the trees, the negative feed forward loops (NFFLs), and the negative feed back loops (NFBLs), respectively.

Appendix B: Classification of Faults and Failures

In this study, the word *fault* is used to designate the departure from an acceptable range of a measurable process variable or calculated parameter associated with an equipment. *Failure*, on the other hand, is taken to mean complete inoperability of an equipment for its intended purpose. Further, they are classified into four types based on their digraph representations and, also, the patterns of their propagation in the system.

Type A. For faults such as disturbances in the process variables or partial component failures (i.e., degradation in the equipment's performance) such as a small leak or a partial plug in a control valve, the corresponding digraph representation should be a node without inputs. The outward edges of such nodes are directed to process variables. A typical digraph model can be found in Figure 13, where x_1 and x_2 are process variables and f is the fault or failure of type A. The effects of this type of faults/failures can be determined by assigning a nonzero value (±1 or ±10) to f, and the values of the other variables in the digraph can then be evaluated accordingly. Notice that, in analyzing these effects for the purpose of classification, the implied assumption is that no other failures exist simultaneously. Further, it should also be noted that, if *both* $x1$ and $x2$ are on the same FBL, the value of x_2 can be affected not only by f but also by x_1.

Type B. The digraph configuration of component failures such as sensor failing high or control valve failing to close is actually the same as that of type A. However, their effects should be analyzed differently. If a failure of type B (f) occurs and both x_1 and x_2 are variables on the same NFBL, then x_2 is always affected by f alone and should be independent of the input x_1.

Type C. Component failures such as a sensor being stuck or a control valve being stuck should be modeled by conditional edges with zero gain. An example can be found in Figure 14. The occurrence of a failure of this type only changes the configuration of the system digraph, i.e., the edge between x_1 and x_2 can be considered as nonexistent. The state variables of the system remain at the normal levels without additional disturbances.

Type D. Component failures such as a controller being reversed (from direct action to reverse action or vice versa) or a control valve being reversed (from air-to-open to air-to-close or vice versa) can also be represented by conditional edges. An example of such failures is presented in Figure 15, which is also represented by a change in the configuration only. Obviously, the occurrence of a failure of type D changes the direction of the effects of an additional fault (if it occurs) propagating from x_1 to x_2.

Literature Cited

Andrews, J. D.; Brennan, G. Application of the Digraph Method of Fault Tree Construction to a Complex Control Configuration. *Reliab. Eng. Syst. Saf.* **1990**, *28*, 357.

Chang, C. T.; Hwang, H. C. New Developments of the Digraph-Based Techniques for Fault-Tree Synthesis. *Ind. Eng. Chem. Res.* **1992**, *31*, 1490.

Chang, C. T.; Hwang, K. S. Studies on the Digraph-Based Approach for Fault-Tree Synthesis. 1. The Ratio-Control Systems. *Ind. Eng. Chem. Res.* **1994**, *33*, 1520.

Chang, C. T.; Hwang, H. C.; Hwang, D. M. Fault-Tree Synthesis Techniques for Process Systems with Coupled Feedforward and Feedback Loops. Paper presented at International Conference on Safety, Health and Loss Prevention in the Oil, Chemical and Process Industries, Singapore, February 1993.

Hunt, A.; Kelly, B. E.; Mulhi, J. S.; Lees, F. P.; Rushton, A. G. The Propagation of Faults in Process Plants: 9, Trip Systems in Fault Tree Synthesis. *Reliab. Eng. Syst. Saf.* **1992**, *39*, 229.

Hwang, D. M. Application of Qualitative Steady-State Analysis in Fault Tree Synthesis. MS Thesis, National Cheng Kung University, Tainan, Taiwan, ROC, 1992.

Lapp, S. A.; Powers, G. J. Computer-Aided Synthesis of Fault Trees. *IEEE Trans. Reliab.* **1977**, *R-26*, 2.

Lees, F. P. *Loss Prevention in the Process Industries*; Butterworths: London, 1980; Vol. 1.

Received for review November 2, 1993
Revised manuscript received April 14, 1994
Accepted April 28, 1994[⊗]

[⊗] Abstract published in *Advance ACS Abstracts*, June 1, 1994.