Available online at www.sciencedirect.com
SCIENCE DIRECT®
Reliability Engineering and System Safety 81 (2003) 163–181
RELIABILITY ENGINEERING & SYSTEM SAFETY
www.elsevier.com/locate/ress

# Fault-tree structures of override control systems☆

Shi-Ning Ju[a], Cheng-Liang Chen[a], Chuei-Tin Chang[b,*]

[a]Department of Chemical Engineering, National Taiwan University, Taipei 10617, Taiwan, ROC
[b]Department of Chemical Engineering, National Cheng Kung University, Tainan 70101, Taiwan, ROC

Received 19 August 2002; accepted 1 April 2003

**Abstract**

The development of a systematic fault-tree synthesis procedure for the override control systems is presented in this paper. The unique configuration of the digraph model under study is first described and then analyzed in detail. On the basis of qualitative simulation of the fault propagation patterns, the corresponding generalized fault-tree structures are then established. It can be observed clearly from the simulation results that none of the existing techniques are capable of producing the correct fault-trees. To demonstrate the correctness of our analysis, successful application of the proposed structure to furnace operation is also presented.


*Keywords:* Fault-tree analysis; Override control system; Digraph; Qualitative simulation; Fault propagation pattern

## 1. Introduction

Override controllers are used to handle an operation problem often confronting the process engineers in chemical plants, i.e. there is just one manipulated variable but two or more outputs to be controlled in a given system. Typically, only one of these outputs is regulated under normal condition. The control objective may be switched during operation to that associated with another output if the process is considered to be unsafe. In an override control system, the switching action is usually accomplished by choosing the lowest (or highest) value among all controller outputs with a low (or high) selector. Thus, other than its regulatory function, override control should also be regarded as a protective strategy. As an often-used protective mechanism, its reliability and safety related issues should be of primary importance. Consequently, there is a real need to quantitatively evaluate the risk of system failure with fault-tree analysis.

In a previous publication [1], 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 [2–11]. Specifically, a set of generalized fault-tree structures (operators) corresponding to various digraph configurations, i.e. tree, feed forward loop and feedback loop, were developed for systems with coupled control and *process* loops. However, it is our belief that a direct application of the existing procedures may fail to produce correct results in the present case. This is due to the fact that the digraph configuration of a selector is really dependent upon the relative values of its inputs. In other words, the complex logics of selectors in the override control systems cannot be adequately modelled with standard digraph representations. This special feature creates unique fault propagation patterns, which are not accounted for in the conventional digraph-based fault-tree synthesis processes. As a result, it becomes necessary to perform a detailed study of the system behaviors and then revise the fault-tree construction techniques, respectively, for the override control systems.

The rest of this article is organized as follows. First, the structural characteristics embedded in the digraph model of a typical override control systems are described in detail. A brief review of the qualitative simulation procedure is also presented. On the basis of the digraph model, the results of a series of exhaustive qualitative simulation studies are then analyzed thoroughly. It can be clearly observed that the existing procedures are indeed incapable of producing fault-trees that incorporate all possible fault propagation patterns considered in this work. Next,

*Abbreviations:* LS, low selector; NFBL, negative feedback loop; NFFL, negative feed forward loop.

☆ RESS MS# 020819.

* Corresponding author. Tel.: 886-6-275-7575x62663; fax: 886-6-234-4496.
*E-mail address:* ctchang@mail.ncku.edu.tw (C.-T. Chang).

0951-8320/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0951-8320(03)00086-3

**Nomenclature**

| | |
|---|---|
| $atd1, atd2$ | the sensor failures of type A (i.e. a drift in the zero) corresponding to exit-stream temperature sensor and tube-surface temperature sensor, respectively. |
| *bcvfc* | the control valve failing close (a type B failure). |
| $bias1, bias2$ | the biases in outputs from controllers TRC-1 and TRC-2, respectively, (type A faults). |
| $btd1, btd2$ | the type B failures corresponding to exit-stream temperature sensors and tube-surface temperature, respectively. |
| $btp1, btp2$ | a set-point change in the temperature controllers TRC-1 and TRC-2, respectively. |
| $cs1(0), cs2(0)$ | the controllers TRC-1 and TRC-2 stick, respectively, (type C failures). |
| $cvs(0)$ | the control valve sticks (a type C failure). |
| *hs* | hot spot on the tube wall (a type A fault). |
| *LSfto* | low selector fails to override. |
| *LSstk* | low selector sticks. |
| $ts1(0), ts2(0)$ | the exit-stream temperature sensor and tube-surface temperature sensor stick, respectively, (type C failures). |

the generalized fault-tree structures for the top event of either a moderate or a large deviation in the controlled variable of override loop are derived from the simulated scenarios. Finally, to demonstrate the correctness of our techniques, successful application of the proposed structures to a realistic example, i.e. a furnace control system, is shown at the end of this article.

## 2. A typical override control system

For illustration convenience, the operation of a furnace is considered as an example throughout this paper. The basic feedback control strategy for this system is shown in Fig. 1, i.e. a temperature controller (TRC-1) is used to adjust the fuel flow rate for maintaining the exit temperature of process stream at the desired set point. However, since the outside surface temperature of the furnace tubes must always remain below the metallurgical limit, an override control system is often preferred for the dual purposes of stability and safety in operation. This override control scheme is presented in Fig. 2. It can be observed that an extra temperature controller (TRC-2) is adopted to control the tube-surface temperature with the same manipulated variable, i.e. the fuel flow rate.

Since, at any instance, only one of the two outputs from controllers TRC-1 and TRC-2 can be used to manipulate fuel flow rate, there is always a need to make an intelligent choice between the two during operation. The selection criteria are of course associated with the dual operation purposes mentioned earlier. Under normal operating conditions, the tube temperature should be lower than the allowable upper limit. Thus, without more serious operational problems, the control objective in this situation is simply to maintain a stable temperature in the exit process stream with controller TRC-1. If, under the influence of certain fault and/or failure, the tube temperature exceeds the acceptable limit, the concerns about potential hazardous consequences should then become the focus of control. In other words, controller TRC-2 takes over and *overrides*

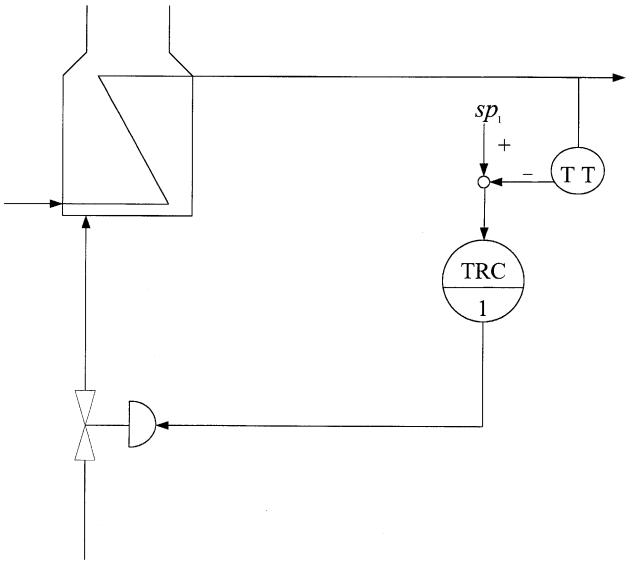



Fig. 1. The flow diagram of a furnace with simple feedback control.

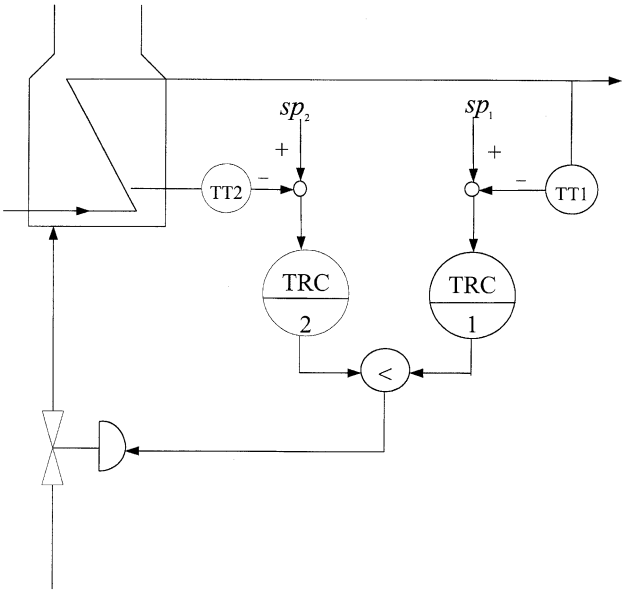



Fig. 2. The flow diagram of a furnace with override control.

the commands issued by TRC-1 at this time. Thus, to facilitate decision making with respect to the aforementioned principles, a low selector (LS) is installed in this override control system.

For the sake of brevity, let us assume that both TRC-1 and TRC-2 are PI controllers. Thus, an important consideration in designing such an override control system is that of reset windup protection. In practice this protection is usually provided by a mechanism of *external reset feedback* in the controller. The block diagrams representing the information flows in the two controllers can be found in Fig. 3. On the basis of the well-established procedure [1,2], the digraph model of this override control system (see Fig. 4) can be easily constructed. The symbols $T_n$ and $m_n$ in Fig. 4 denote, respectively, the temperature and flow rate of process stream $n$, and $S_l$ represents the measurement or control signal on line l. The definitions of other nodes in this model can be found in Nomenclature. Two negative feedback loops (NFBLs) can be identified in the digraph, i.e. loop I (the exit-stream temperature control loop): $T_4 \rightarrow S_7 \rightarrow S_8 \rightarrow S_9 \rightarrow m_3 \rightarrow T_4$ and loop II (the tube-surface temperature control loop): $T_2 \rightarrow S_5 \rightarrow S_6 \rightarrow S_9 \rightarrow m_3 \rightarrow T_2$. In addition, there are two negative feed forward loops (NFFLs)

$$\left\{ \begin{array}{l} T_1 \rightarrow T_4 \rightarrow S_7 \rightarrow S_8 \rightarrow S_9 \rightarrow m_3 \rightarrow T_2 \\ T_1 \rightarrow T_2 \end{array} \right\} \quad (1)$$

$$\left\{ \begin{array}{l} m_1 \rightarrow T_4 \rightarrow S_7 \rightarrow S_8 \rightarrow S_9 \rightarrow m_3 \rightarrow T_2 \\ m_1 \rightarrow T_2 \end{array} \right\} \quad (2)$$

Notice that the value of node 'Normal Set Pt II' is fixed at $+1$, i.e. the set point is higher than the normal value of

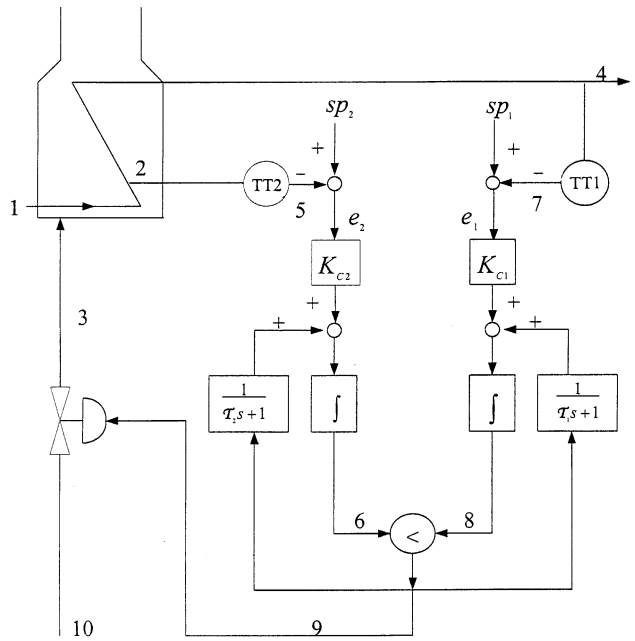



Fig. 3. The block diagram of PI controllers (with external reset feedback) in the override control system of a furnace.

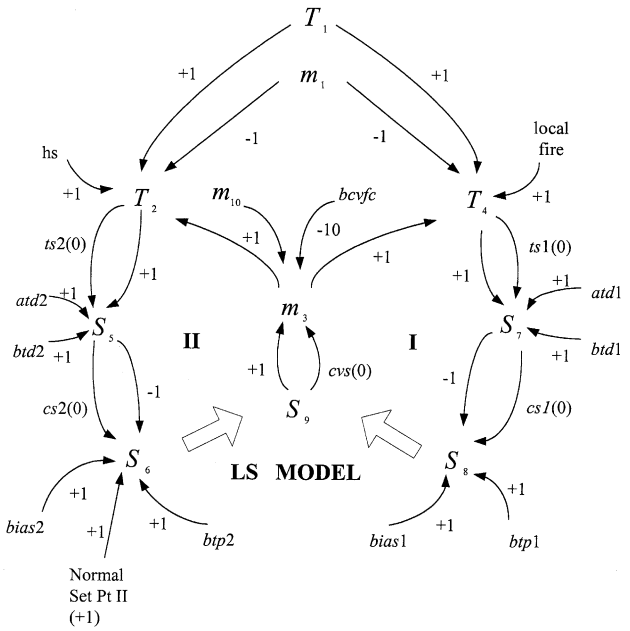



Fig. 4. The digraph model of a furnace with override control.

tube temperature. This is due to the need to utilize loop I during normal operation. In other words, the value of $S_6$ (the output of TRC-2) should be larger than that of $S_8$ (the output of TRC-1) in this situation.

Finally, it should be noted that the special symbols $S_6 \Rightarrow S_9$ and $S_8 \Rightarrow S_9$ in Fig. 4 are used to represent the unique relations between the input and output of a LS. These relations cannot be concisely expressed with standard digraph representations. In particular, each of them can be viewed as two simple arcs in opposite directions (Fig. 5). The arcs $S_6 \rightarrow S_9$ and $S_8 \rightarrow S_9$ show that the output of LS may be affected by either of its inputs. On the other hand, the external reset feedback mechanisms in the two controllers are described with the arcs $S_9 \rightarrow S_6$ and $S_9 \rightarrow S_8$. The arc gains of a functional LS can be found in Table 1. The gains in row 1 are associated with the condition that loop I is in charge and those in row 2 are the ones used in case of loop II taking over. The values of $(S_6 - S_8)$ in the first column represent the overriding conditions. Specifically, '$+1$' (or '$+10$') in the first row denotes that $S_6$ is moderately (or significantly) larger than $S_8$. On the other hand, '$-1$' and '$-10$' in the second row represent the

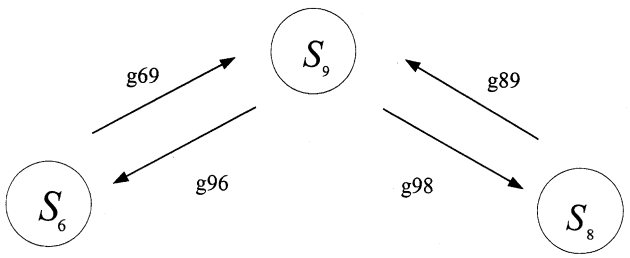



Fig. 5. The digraph model of a selector.

Table 1
The digraph model of a functional low selector

| $S_6$–$S_8$ | $g_{69}$ | $g_{96}$ | $g_{89}$ | $g_{98}$ |
|---|---|---|---|---|
| 0, +1, +10 | 0 | +1 | +1 | +1 |
| −1, −10, 0 | +1 | +1 | 0 | +1 |

Table 2
The modified LS model

| LS state | $S_6$–$S_8$ | $g_{69}$ | $g_{96}$ | $g_{89}$ | $g_{98}$ |
|---|---|---|---|---|---|
| Functional | 0, +1, +10 | 0 | +1 | +1 | 0 |
| | −1, −10, 0 | +1 | 0 | 0 | +1 |
| Fails to override | All | 0 | +1 | +1 | 0 |
| Sticks | All | 0 | 0 | 0 | 0 |

corresponding conditions in the opposite direction. Since '0' is a qualitative statement of $S_6 \simeq S_8$ and either loop may be activated in this situation, this condition is included in both rows. The normal control mode is reflected in the gains $g_{89}$ ( = 1) and $g_{69}$ ( = 0) given in row 1. The values of these two gains in row 2 indicate that $S_9$ is only affected by $S_6$ in the overriding control mode. To incorporate the external reset feedback mechanisms in the LS model, the gains $g_{98}$ and $g_{96}$ are both set to +1 in rows 1 and 2.

For the system corresponding to the first row in Table 1, it is clear that $S_8$ and $S_9$ can be lumped to form a fictitious node. By the same token, $S_6$ and $S_9$ can also be merged in the model described by row 2. Thus, in order to simplify the digraph configuration and thus avoid overly complex analysis, Table 1 is replaced by the first two rows of Table 2 in this study. In addition, two LS failure modes are considered in this work: (1) LS fails to override and (2) LS sticks. Their digraph models can be represented with the third and fourth rows in Table 2. In the former case, loop I is in control under any circumstances. On the other hand, the output of LS is independent of its inputs should the latter failure occurs.

## 3. The standard digraph model

In this study, it is assumed that there exists an underlying basic digraph configuration for all override control systems. Thus, it is possible to develop a generic fault-tree synthesis procedure accordingly. To facilitate the derivation of generalized fault-tree structures, let us use a set of standard identifiers to replace the node labels in Fig. 4. The resulting digraph model is presented in Fig. 6. Notice first that the nodes shared by loops I and II are denoted with single-subscript identifiers while the rest with double-subscript ones. The first subscript represents the output of a fundamental component in the control system and the second (if it exists) is used for loop

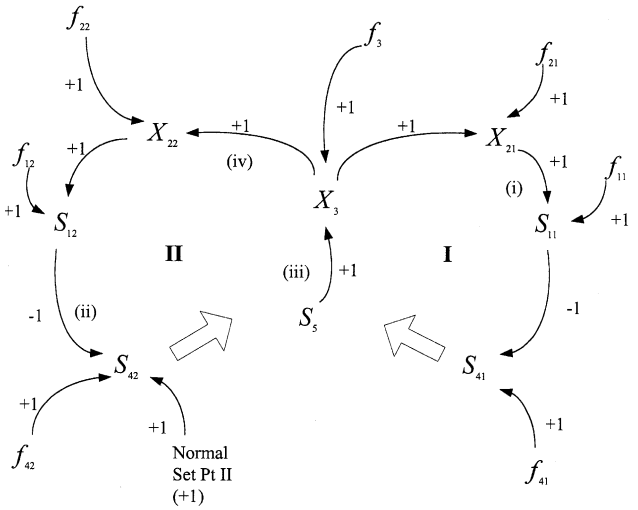



Fig. 6. The standard digraph model of an override control system.

identification. Specifically, $S_{11}$, $S_{41}$ and $X_{21}$ denote, respectively, the sensor output, controller output and controlled variable in loop I; $S_{12}$, $S_{42}$ and $X_{22}$ denote, respectively, the sensor output, controller output and controlled variable in loop II; $X_3$ is the only manipulated variable in the override control system; $f_{ij}$ represents the fault or failure affecting the loop variable at position $ij$. Finally, it is assumed that a LS is adopted in the standard system and its output is $S_5$. Notice that the LS model in the standard digraph is also described with Table 2. In this case, the variables $S_6$, $S_8$ and $S_9$ should be replaced with $S_{42}$, $S_{41}$ and $S_5$, respectively.

The two NFBLs in Fig. 6 can be identified easily, i.e. loop I: $X_{21} \rightarrow S_{11} \rightarrow S_{41} \rightarrow S_5 \rightarrow X_3 \rightarrow X_{21}$ and loop II: $X_{22} \rightarrow S_{12} \rightarrow S_{42} \rightarrow S_5 \rightarrow X_3 \rightarrow X_{22}$. Here, loop I is the *normal control loop* and loop II is the *override control loop*. To facilitate later discussion, these two loops are further divided into four paths in this study: path (i): $X_3 \rightarrow X_{21} \rightarrow S_{11} \rightarrow S_{41}$; path (ii): $X_{22} \rightarrow S_{12} \rightarrow S_{42}$; path (iii): $S_5 \rightarrow X_3$; path (iv): $X_3 \rightarrow X_{22}$.

## 4. Qualitative simulation

To develop fault-tree structures corresponding to the standard digraph, it is necessary to gain a thorough understanding of the fault propagation behaviors in the override control system first. All possible initiating faults and equipment failures can be classified into four different types (A, B, C and D) according to the criteria suggested by Himmelblau [12] and Chang and Hwang [1]. For the sake of completeness, their definitions are repeated in Appendix A.

To reduce the number of scenarios that must be included in the fault-trees, the following assumptions are adopted

- The two controllers in the override control system are well designed and tuned.
- Component malfunctions that reverse the signs of arc gains in the digraph, i.e. the type D failures, do not exist in the system.
- The probability of simultaneous occurrence of two or more type B and/or C failures is negligible.

The first assumption implies that all failure mechanisms due to improper use of control parameters, i.e. the proportional gain and reset time, are excluded from consideration. On the other hand, type D failures are excluded because they can be almost always eliminated by preventive inspection before startup. Finally, the third assumption is justified by the fact that the probability of a single type B or C failure is usually very low and that of multiple such failures should be even lower. It is thus only necessary to consider the effects of a single type A fault or type B failure and the combined effects of a type A fault and a type C failure.

Generally speaking, a digraph model explicitly describes the cause–effect relationships between deviations in process variables (represented by 0, $\pm 1$ and $\pm 10$) and component failures (represented by 0, 1 and 10). The effects of a type A fault or a type B failure can thus be determined by first assigning a non-zero value ($\pm 1$ or $\pm 10$) to the corresponding node variable $f_{ij}$ and then evaluating the values of all other affected variables. In a simple loop-free digraph, any of these variables can be determined by multiplying its input value with the corresponding edge gain. In other words, the output value of an arc can be computed according to the following equation

$$v_{\text{out}} = \begin{cases} gv_{\text{in}} & \text{if } -10 \le gv_{\text{in}} \le +10 \\ +10 & \text{if } gv_{\text{in}} > +10 \\ -10 & \text{if } gv_{\text{in}} < -10 \end{cases} \quad (3)$$

where $g$, $v_{\text{in}}$ and $v_{\text{out}}$ denote, respectively, the gain, input and output values. This evaluation process is generally referred to as *qualitative simulation* in the present study.

However, this approach becomes infeasible if the system digraph contains NFBLs and/or NFFLs. In particular, two opposite effects on the loop variables are caused by an external disturbance. To describe the behaviors of the loop variables more accurately, Chang and Hwang [1] proposed an improved procedure to simulate qualitatively the corresponding fault propagation sequences in a *single* NFBL. For the sake of completeness, a brief description of the additional computation rules used for qualitative simulation is also included in Appendix B. On the other hand, the net effect of multiple inputs on the terminal node of a NFFL can be determined according to the rules presented in Table 3.

If operation safety of the override control system is the main concern of fault-tree analysis, the appropriate top events should be those expressed in terms of positive deviations in the controlled variable of loop II, i.e. $X_{22}(+1)$ and $X_{22}(+10)$. In the former case, $X_{22}$ is raised to a level near the allowable upper limit, i.e. $+1$, which should be approximately the same as the set point of loop II. Notice that such an event itself represents a potential hazard leading to accidents. However, if $X_{22}$ exceeds the upper limit by a significant amount, i.e. reaches the level $+10$, undesirable consequences are almost certain to occur. Since the focuses of our analysis are inevitably concerned with these two events, the subsequent discussions are thus presented accordingly.

Table 3
The rules of simultaneous effects

| Inputs | Output |
|---|---|
| $-10, -10$ | $-10$ |
| $-10, -1$ | $-10$ |
| $-10, 0$ | $-10$ |
| $-1, -1$ | $-10$ |
| $-10, +1$ | $-1$ |
| $-1, 0$ | $-1$ |
| $-10, +10$ | 0 |
| $-1, +1$ | 0 |
| 0, 0 | 0 |
| $-1, +10$ | $+1$ |
| $0, +1$ | $+1$ |
| $0, +10$ | $+10$ |
| $+1, +1$ | $+10$ |
| $+1, +10$ | $+10$ |
| $+10, +10$ | $+10$ |

## 5. Scenarios causing a moderate deviation in $X_{22}$

The qualitative simulation approach described earlier can be applied to the multi-loop override control systems. Following is a detailed account of the simulation results.

### 5.1. Effects of a type A fault

As mentioned previously, the LS model cannot be properly built with standard digraph elements and thus cannot be handled directly with any of the existing simulation techniques. In particular, the digraph configuration of LS may be varying in the course of fault propagation depending upon the values of its two inputs, $S_{41}$ and $S_{42}$. Thus, special care must be taken to ensure the validity of LS model after $S_{41}$ and $S_{42}$ are computed during simulation.

The results of qualitative simulation corresponding to 'controllable' faults of type A are summarized in Table 4. A qualitative value '1' is assigned to be the magnitude of the fault if it occurs. Notice that the responses of the system variables are expressed with symbols of the form $(v_0, v_\infty)$. This symbol is interpreted as the state of a loop variable which would have a value $v_0$ without feedback but approaches $v_\infty$ at the new steady state due to the regulatory action. A more detailed discussion can be found in Appendix B. In this study, a fault is referred to as controllable if its effects can be compensated with loop I, i.e. the final values of its loop variables can be brought back to 0 (the normal set point of

Table 4
Simulation results: a controllable type A fault

| Fault origin | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ (+1) | (+1, 0) | (+1, 0) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | (+1, +1) | N |
| $f_{11}$ (+1) | (−1, −1) | (+1, 0) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | (+1, +1) | N |
| $f_{41}$ (+1) | (+1, 0) | (+1, 0) | (+1, 0) | (+1, 0) | (+1, 0) | (+1, 0) | (+1, 0) | (+1, +1) | N |
| $f_3$ (+1) | (+1, 0) | (+1, 0) | (−1, −1) | (−1, −1) | (+1, 0) | (+1, 0) | (+1, 0) | (−1, 0) | N |
| $f_{22}$ (+1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (+1, +1) | (+1, +1) | (0, 0) | I |
| $f_{12}$ (+1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (+1, +1) | (0, 0) | I |
| $f_{42}$ (+1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (+10, +10) | N |
| $f_{21}$ (−1) | (−1, 0) | (−1, 0) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | I |
| $f_{11}$ (−1) | (+1, +1) | (−1, 0) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | I |
| $f_{41}$ (−1) | (−1, 0) | (−1, 0) | (−1, 0) | (−1, 0) | (−1, 0) | (−1, 0) | (−1, 0) | (+1, +1) | N |
| $f_3$ (−1) | (−1, 0) | (−1, 0) | (+1, +1) | (+1, +1) | (−1, 0) | (−1, 0) | (−1, 0) | (+10, +10) | N |
| $f_{22}$ (−1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (−1, −1) | (−1, −1) | (+10, +10) | N |
| $f_{12}$ (−1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (−1, −1) | (+10, +10) | N |
| $f_{42}$ (−1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | I |

loop I). Notice that the eventual system status in each scenario is indicated in the last column of Table 4. Specifically

- 'N' denotes that loop I remains in control and the relation between the inputs of LS can be written as

$$S_{42} - S_{41} = +1 \text{ or } +10 \tag{4}$$

- 'Y' denotes that loop II is activated and the relations between the inputs of LS can be described with

$$S_{42} - S_{41} = -1 \text{ or } -10 \tag{5}$$

- 'I' denotes that either loops may be in charge, since the two inputs are approximately equal, i.e.

$$S_{42} - S_{41} = 0 \tag{6}$$

Notice also that none of the type A faults in Table 4 activate loop II. The simulation results presented in the rows corresponding to 'N' were obtained on the basis of the first row of Table 2. It should be noted that the controller output on loop II, i.e. $S_{42}$, is affected simultaneously by three inputs, i.e. $S_5$, $S_{12}$, and also the normal set point of loop II. Thus, it should be evaluated according to the rules of simultaneous effects given in Table 3. Although the status of override control system corresponding to the row labelled by 'I' is indeterminable, the simulation of fault propagation pattern is done by assuming that loop II has not been triggered. If this condition is violated, the final values of loop II variables should always reach the set point. As a result, simulation of the corresponding fault propagation behavior is quite straightforward. Finally, it should be noted that, since loop I is in charge, only five variables, i.e. $X_{21}$, $S_{11}$, $S_{41}$, $S_5$ and $X_3$, can be treated as loop variables. However, all other variables, i.e. $X_{22}$, $S_{12}$ and $S_{42}$, are also expressed in the form $(v_0, v_\infty)$ in Table 4. This is because they are affected directly or indirectly by the loop variables in the cases considered here.

The results of qualitative simulation corresponding to type A faults with magnitude 10 are summarized in Table 5.

Table 5
Simulation results: an uncontrollable type A fault

| Fault origin | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ (+10) | (+10, +1) | (+10, +1) | (−10, −10) | (−10, −10) | (−10, −10) | (−10, −10) | (−10, −10) | (+1, +1) | N |
| $f_{11}$ (+10) | (−10, −10) | (+10, +1) | (−10, −10) | (−10, −10) | (−10, −10) | (−10, −10) | (−10, −10) | (+1, +1) | N |
| $f_{41}$ (+10) | (+1, +1) | (+1, +1) | (+10, +10) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | Y |
| $f_3$ (+10) | (+10, +1) | (+10, +1) | (−10, −10) | (−10, −10) | (+10, +1) | (+10, +1) | (+10, +1) | (−10, −10) | I |
| $f_{22}$ (+10) | (−1, −1) | (−1, −1) | (0, 0) | (−1, −1) | (−1, −1) | (+10, +1) | (+10, +1) | (−1, −1) | Y |
| $f_{12}$ (+10) | (−1, −1) | (−1, −1) | (0, 0) | (−1, −1) | (−1, −1) | (−1, −1) | (+10, +1) | (−1, −1) | Y |
| $f_{42}$ (+10) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (+10, +10) | N |
| $f_{21}$ (−10) | (−10, −1) | (−10, −1) | (+10, +10) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | Y |
| $f_{11}$ (−10) | (+1, +1) | (−10, −1) | (+10, +10) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | Y |
| $f_{41}$ (−10) | (−10, −1) | (−10, −1) | (−10, −1) | (−10, −1) | (−10, −1) | (−10, −1) | (−10, −1) | (+1, +1) | N |
| $f_3$ (−10) | (−10, −1) | (−10, −1) | (+10, +10) | (+10, +10) | (−10, −1) | (−10, −1) | (−10, −1) | (+10, +10) | I |
| $f_{22}$ (−10) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (−10, −10) | (−10, −10) | (+10, +10) | N |
| $f_{12}$ (−10) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (−10, −10) | (+10, +10) | N |
| $f_{42}$ (−10) | (−1, −1) | (−1, −1) | (0, 0) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | Y |

In this study, these faults are treated as *uncontrollable* with respect to loop I, but otherwise when loop II is in charge. This is reflected in the final value of the loop variable directly affected by the type A fault. Notice that the set point of loop II, which is given a qualitative value of +1, is higher than that of loop I, i.e. 0. The results presented in the rows marked with 'N' or 'I' were obtained with the same method for generating Table 4. In cases associated with rows 1, 2, 4, 10 and 11, the results clearly show that loop I is saturated. In the scenarios corresponding to rows 7, 12 and 13, loop I is not affected by faults. On the other hand, since loop II is activated in the cases corresponding to the rows labeled with 'Y,' the final values of loop II variables should be obtained on the basis of the second row in Table 2. The results in rows 3, 5, 6, 8 and 9 show that the sensor output $S_{12}$ in override loop can be brought to the new set point +1. The results in row 14 show that the $S_{12}$ reaches a value −1, which is the 'apparent' set point resulting from the combined effects of $f_{42}$ and the original set point. Finally, if loop II is triggered, the controller output on loop I should be affected by two simultaneous inputs, i.e. $S_5$ and $S_{11}$, and thus the evaluation of $S_{41}$ in these cases should also be done according to the rules of simultaneous effects (Table 3).

### 5.2. Effects of a type B failure

Notice that a fault of type A does not change the structure of NFBL, i.e. the feedback mechanism of the control system is still intact. However, if a component failure of type B occurs on an *activated* NFBL, the regulatory function of the corresponding control loop will be lost completely. In these situations, the use of representation $(v_0, v_\infty)$ is no longer suitable for describing the behaviors of its loop variables.

The related results are presented in Table 6. In cases associated with rows 1–4, 10 and 11, since loop I is broken by a type B failure and also loop II is not activated by LS, the simulation of fault propagation patterns can be accomplished easily with the conventional techniques. In cases corresponding to rows 5–7 and 12–14, since loop II is disconnected and also LS selects only the controller output from loop I, loop I is virtually unaffected by the failures. Finally, the results in rows 8 and 9 show that loop II is activated and the type B failures are located on loop I in these two cases. The value of sensor output in loop II should be brought to the new set point +1 due to the servomechanism of the override controller.

### 5.3. Combined effects of a type A fault and a type C failure

It should be noted that type C failures are described with conditional arcs in the digraph. Since its gain is zero, the corresponding arc in digraph is essentially severed by such a failure. As mentioned earlier, the loops in an override control system can be classified into four paths on the basis of their respective structural characteristics. The discussion here is thus divided, respectively

- *The type C failure is on path (i).* Loop I is broken in the event of a type C failure occurring on path (i). Thus, a NFBL is formed only when the override loop takes over. A sample of the simulation results is provided in Table 7. Notice that, only in cases corresponding to rows 8 and 9 in Table 7, the regulatory function of loop II is active and thus the values of its loop variables are represented with the format $(v_0, v_\infty)$.
- *The type C failure is on path (ii) or (iv).* Loop II is disconnected in this situation. Loop I is disabled only when $S_{41}$ is higher than $S_{42}$. A sample of the corresponding simulation results is given in Table 8. It can be observed that loop I is in charge in every scenario. Thus, all the loop-I variables are expressed in the form $(v_0, v_\infty)$.
- *The type C failure is on path (iii).* Both loops are broken if such a failure occurs. Since the resulting digraph does not contain NFBL, all variables in the system can be easily computed with the conventional techniques. The results can be found in Table 9.

Table 6
Simulation results: a type B failure

| Fault origin | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ (+1) | +1 | +1 | −10 | −10 | −10 | −10 | −10 | +1 | N |
| $f_{11}$ (+1) | −10 | +1 | −10 | −10 | −10 | −10 | −10 | +1 | N |
| $f_{41}$ (+1) | +1 | +1 | +1 | +1 | +1 | +1 | +1 | +1 | I |
| $f_3$ (+1) | +1 | +1 | −10 | −10 | +1 | +1 | +1 | −10 | I |
| $f_{22}$ (+1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | +1 | +1 | (0, 0) | I |
| $f_{12}$ (+1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | +1 | (0, 0) | I |
| $f_{42}$ (+1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (+10, +10) | N |
| $f_{21}$ (−1) | −1 | −1 | (+1, +10) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | Y |
| $f_{11}$ (−1) | (+1, +1) | −1 | (+1, +10) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | Y |
| $f_{41}$ (−1) | −1 | −1 | −1 | −1 | −1 | −1 | −1 | +1 | N |
| $f_3$ (−1) | −1 | −1 | +10 | +10 | −1 | −1 | −1 | +10 | I |
| $f_{22}$ (−1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | −1 | −1 | (+10, +10) | N |
| $f_{12}$ (−1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | −1 | (+10, +10) | N |
| $f_{42}$ (−1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | I |

Table 7
Simulation results: a controllable type A fault and the type C failure corresponding to $X_3 \xrightarrow{0} X_{21}$

| Fault origin | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ (+1) | +1 | +1 | −10 | −10 | −10 | −10 | −10 | +1 | N |
| $f_{11}$ (+1) | 0 | +1 | −10 | −10 | −10 | −10 | −10 | +1 | N |
| $f_{41}$ (+1) | 0 | 0 | +1 | +1 | +1 | +1 | +1 | +1 | I |
| $f_3$ (+1) | 0 | 0 | 0 | 0 | +1 | +1 | +1 | 0 | I |
| $f_{22}$ (+1) | 0 | 0 | 0 | 0 | 0 | +1 | +1 | 0 | I |
| $f_{12}$ (+1) | 0 | 0 | 0 | 0 | 0 | 0 | +1 | 0 | I |
| $f_{42}$ (+1) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | +10 | N |
| $f_{21}$ (−1) | −1 | −1 | (+1, +10) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | Y |
| $f_{11}$ (−1) | 0 | −1 | (+1, +10) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | (+1, +1) | Y |
| $f_{41}$ (−1) | 0 | 0 | −1 | −1 | −1 | −1 | −1 | +1 | N |
| $f_3$ (−1) | 0 | 0 | 0 | 0 | −1 | −1 | −1 | +10 | N |
| $f_{22}$ (−1) | 0 | 0 | 0 | 0 | 0 | −1 | −1 | +10 | N |
| $f_{12}$ (−1) | 0 | 0 | 0 | 0 | 0 | 0 | −1 | +10 | N |
| $f_{42}$ (−1) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I |

Table 8
Simulation results: a controllable type A fault and the type C failure corresponding to $X_3 \xrightarrow{0} X_{22}$

| Fault origin | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ ($+_1$) | (+1, 0) | (+1, 0) | (−1, −1) | (−1, −1) | (−1, −1) | 0 | 0 | (0, 0) | N |
| $f_{11}$ ($+_1$) | (−1, −1) | (+1, 0) | (−1, −1) | (−1, −1) | (−1, −1) | 0 | 0 | (0, 0) | N |
| $f_{41}$ ($+_1$) | (+1, 0) | (+1, 0) | (+1, 0) | (+1, 0) | (+1, 0) | 0 | 0 | (+10, +1) | N |
| $f_3$ ($+_1$) | (+1, 0) | (+1, 0) | (−1, −1) | (−1, −1) | (+1, 0) | 0 | 0 | (0, 0) | N |
| $f_{22}$ ($+_1$) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | +1 | +1 | (0, 0) | I |
| $f_{12}$ ($+_1$) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | 0 | +1 | (0, 0) | I |
| $f_{42}$ ($+_1$) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | 0 | 0 | (+10, +10) | N |
| $f_{21}$ ($-_1$) | (−1, 0) | (−1, 0) | (+1, +1) | (+1, +1) | (+1, +1) | 0 | 0 | (+10, +10) | N |
| $f_{11}$ ($-_1$) | (+1, +1) | (−1, 0) | (+1, +1) | (+1, +1) | (+1, +1) | 0 | 0 | (+10, +10) | N |
| $f_{41}$ ($-_1$) | (−1, 0) | (−1, 0) | (−1, 0) | (−1, 0) | (−1, 0) | 0 | 0 | (0, +1) | N |
| $f_3$ ($-_1$) | (−1, 0) | (−1, 0) | (+1, +1) | (+1, +1) | (−1, 0) | 0 | 0 | (+10, +10) | N |
| $f_{22}$ ($-_1$) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | −1 | −1 | (+10, +10) | N |
| $f_{12}$ ($-_1$) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | 0 | −1 | (+10, +10) | N |
| $f_{42}$ ($-_1$) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | 0 | 0 | (0,0) | I |

Table 9
Simulation results: a controllable type A fault and the type C failure corresponding to $S_5 \xrightarrow{0} X_3$

| Fault origin | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|
| $f_{21}$ (+1) | +1 | +1 | −10 | −10 | 0 | 0 | 0 | −1 | N |
| $f_{11}$ (+1) | 0 | +1 | −10 | −10 | 0 | 0 | 0 | −1 | N |
| $f_{41}$ (+1) | 0 | 0 | +1 | +1 | 0 | 0 | 0 | +10 | N |
| $f_3$ (+1) | +1 | +1 | −10 | −10 | +1 | +1 | +1 | −10 | I |
| $f_{22}$ (+1) | 0 | 0 | 0 | 0 | 0 | +1 | +1 | 0 | I |
| $f_{12}$ (+1) | 0 | 0 | 0 | 0 | 0 | 0 | +1 | 0 | I |
| $f_{42}$ (+1) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | +10 | N |
| $f_{21}$ (−1) | −1 | −1 | +10 | +10 | 0 | 0 | 0 | +10 | I |
| $f_{11}$ (−1) | 0 | −1 | +10 | +10 | 0 | 0 | 0 | +10 | I |
| $f_{41}$ (−1) | 0 | 0 | −1 | −1 | 0 | 0 | 0 | 0 | N |
| $f_3$ (−1) | −1 | −1 | +10 | +10 | −1 | −1 | −1 | +10 | I |
| $f_{22}$ (−1) | 0 | 0 | 0 | 0 | 0 | −1 | −1 | +10 | N |
| $f_{12}$ (−1) | 0 | 0 | 0 | 0 | 0 | 0 | −1 | +10 | N |
| $f_{42}$ (−1) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I |

## 6. The generalized fault-tree structures for top event $X_{22}(+1)$

The conventional digraph-based approach [1,2] is followed in this work to synthesize the fault-trees. In principle, every event in a fault-tree can be associated with a distinct node in the corresponding digraph. The fault-tree can be synthesized by identifying the appropriate logic gate (and also its input events) connected to an undeveloped event on the basis of the subgraph containing the corresponding node and its inputs and outputs. Thus, the above simulation results should be analyzed and re-organized on the basis of *node locations* to facilitate implementation of such an approach.

It can be observed from Tables 4–9 that every possible cause of $X_{22}(+1)$ involves at least a type A fault or a type B failure. Each of these faults/failures is always represented in the system digraph by a *primal node*, i.e. a node without inputs. Under the condition that LS is functional, these basic events can be conveniently classified into two groups according to the *locations* of the corresponding primal nodes. If the output of a primal node affects a variable on the path between the LS output $S_5$ and $X_{22}$, the basic event is referred to as a *downstream disturbance* in this paper. Otherwise, it is *upstream disturbance*. The causes of top event $X_{22}(+1)$ are thus divided into two parts, respectively.

- The causes with downstream disturbances include
  - the single type A faults: $f_{22}(+1), f_3(+10)$;
  - the single type B failures: $f_{22}(+1), f_3(+1)$;
  - the combinations of a type A fault and a type C failure: $\{f_3(+1), S_5 \xrightarrow{0} X_3\}$, $\{f_3(+1), X_3 \xrightarrow{0} X_{21}\}$, $\{f_3(+1), X_{21} \xrightarrow{0} S_{11}\}$, $\{f_3(+1), S_{11} \xrightarrow{0} S_{41}\}$.
- The causes with upstream disturbances include
  - the single type A faults: $f_{41}(+10)$, $f_{11}(-1)$, $f_{21}(-1)$;
  - the single type B failures: $f_{41}(+1)$, $f_{11}(-1)$, $f_{21}(-1)$;
  - the combinations of a type A fault and a type C failure: $\{f_{41}(+1), X_3 \xrightarrow{0} X_{21}\}$, $\{f_{41}(+1), X_{21} \xrightarrow{0} \times S_{11}\}$, $\{f_{41}(+1), S_{11} \xrightarrow{0} S_{41}\}$.

Thus, it can be observed that the propagation mechanisms of these two groups of causes are significantly different. In the former case, a downstream disturbance directly affects the controlled variable of loop II through the path between the corresponding primal node and $X_{22}$. The effects of disturbance cannot be compensated since loop II is not triggered and, at the same time, loop I is either saturated by an uncontrollable type A fault or disabled by a type B or type C failure.

In the latter case, an upstream disturbance gives rise to the outcome $X_{22}(+1)$ *indirectly*. It should be noted that the corresponding faults and/or failures are located exclusively on path (i). In particular, the upstream disturbance must first produce a change in $S_{41}$ before affecting $X_{22}$. There are two important scenarios that must be considered here. First of all, if the positive deviation in $S_{41}$ is large, i.e. $S_{41}(+10)$, then loop II should be in charge and all its loop variables, including $X_{22}$, should be regulated according to the new set point $+1$. Second, if the positive deviation in $S_{41}$ is only moderate, i.e. $S_{41}(+1)$, the resulting override status becomes indeterminable since the qualitative value of the controller output from loop II is also $+1$. Although either loop may be in charge, let us assume that loop II takes over in this situation to ensure a more pessimistic analysis. As a result, $X_{22}$ should also eventually reach the level of $+1$, i.e. the set point of loop II. From the aforementioned analysis, it can be concluded that all the upstream events mentioned earlier result in loop II overriding. The top event $X_{22}(+1)$ is just the inevitable outcome of the intermediate event $S_{41}(+10)$ or $S_{41}(+1)$.

From the simulation results, another interesting feature of the failure mechanism of override control system can be observed, i.e. none of the faults/failures affecting the variables on path (ii) are included as the causes of top event $X_{22}(+1)$. Thus, these basic events can actually be ignored in analysis. This insight is very critical in the development of a generalized fault-tree synthesis procedure. Specifically, a correct fault-tree can be constructed with the traditional techniques [1,2] according to the fictitious digraph given in Fig. 7. Notice that loop I is assumed to be always in charge despite the fact that loop II may be activated. This approach is feasible because (1) the causes with downstream disturbances do not trigger loop II and (2) the causes with upstream disturbances always result in $S_{41}(+1)$ or $S_{41}(+10)$. Notice that the fault-tree produced on the basis of Fig. 7 is guaranteed to include all the former causes as its basic events and $S_{41}(+1)$ as an intermediate event. On the other hand, the latter causes can be identified in the sub tree under $S_{41}(+1)$ generated with the proposed approach.

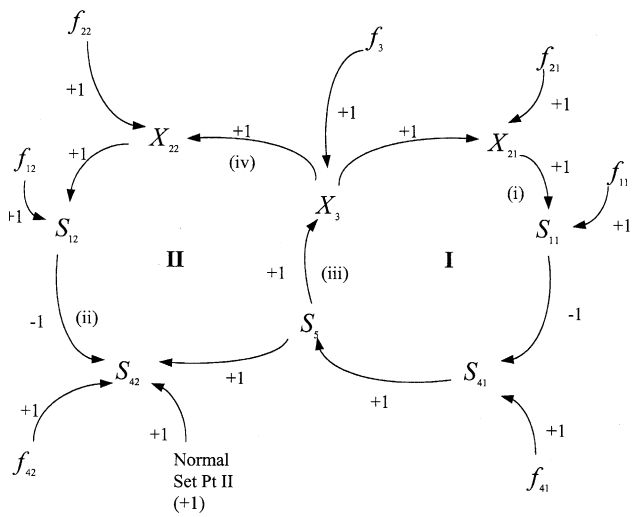
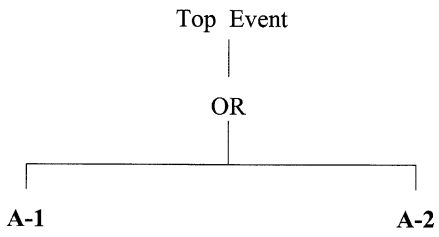



Fig. 7. The fictitious digraph used for synthesizing fault-trees with top event $X_{22}(+1)$.




**Remarks** : The top event considered in this fault tree is limited to the event $X_0(\pm 1)$ which is associated with a node on path (iv). The sub-tree A-1 can be developed by applying the traditional algorithms according to a fictitious system digraph with fixed configuration. This fictitious digraph is constructed by assuming LS cannot be triggered under any circumstance. On the other hand, the sub-tree A-2 can be developed on the basis of the observation that the only valid failure mode concerning the LS is "LS stuck ." In this case, both loop I and loop II are broken and the traditional algorithms are still applicable.

Fig. 8. Structure A.

Finally, let us consider the impacts of LS failures

- *LS fails to override*. The digraph model of LS is described in the third row of Table 2. Since the fault-tree associated with top event $X_{22}(+1)$ in this case should be the same as that obtained according to the fictitious digraph in Fig. 7, this failure mode can be ignored in the fault-tree.
- *LS sticks*. The digraph model of LS is described in the fourth row of Table 2. Specifically, both loops are broken by such a failure. Since the resulting system digraph contains only tree-like structures, the corresponding fault-tree can be easily generated according to the input–output relations defined by the arcs.

On the basis of earlier discussions, a generalized fault-tree synthesis procedure for top event $X_{22}(+1)$ can be summarized with Fig. 8. Since the figure is self-explanatory, its implementation steps are not elaborated here.

## 7. Scenarios causing a large deviation in $X_{22}$

To study the failure mechanisms leading to $X_{22}(+10)$, it is best to consider them separately on the basis of downstream and upstream disturbances. As mentioned before, the former may directly affect the controlled variable of loop II via the path between $S_5$ and $X_{22}$. On the other hand, the latter must first create a large positive deviation in $S_5$ before causing the top event. This implies that the controller outputs on loops I and II must both reach the maximum level since only the signal with lower value can pass LS. In other words, $S_{41}(+10)$ and $S_{42}(+10)$ should coexist and they are the results of the faults/failures on paths (i) and (ii). Following is a detailed analysis of these two types of fault propagation behaviors.

### *7.1. The causes with downstream disturbances*

The combinations of faults and/or failures considered here are essentially the same as those for $X_{22}(+1)$. The discussions are thus presented with the same format in the sequel.

#### *7.1.1. The effects of a type A fault*

From Tables 4 and 5, it can be observed that none of the type A faults are capable of producing a large positive deviation in $X_{22}$.

#### *7.1.2. The effects of a type B failure*

The results in Table 6 show that the top event cannot be the result of any type B failure of magnitude 1. Additional simulation studies have been carried out to determine the effects of serious type B failures off path (iii) and (iv). From the results presented in Table 10, we can conclude that these failures can indeed be the root causes of $X_{22}(+10)$.

#### *7.1.3. The combined effects of a type A fault and one or more type C failure*

The magnitude of type A faults in this case must be 10. This is due to the observation that none of the related scenarios listed in Tables 7–9, i.e. the rows corresponding to $f_3$ and $f_{22}$, end up with the given top event. Notice also that the first loop variable affected by a type A fault considered here is located either on paths (iii) or (iv). The following discussions are thus presented, respectively

- *The affected loop variable is on path (iii)*. Since path (iii) is the shared path of loops I and II, the effects of entering disturbances can be regulated by one of the two controllers depending on the relative values of their output. In either case, $X_{22}$ can be controlled to reach a level which is lower than or equal to $+1$ if the system is normal. Thus, the top event can occur only under the additional condition that both loops are inactive at the same time. This condition can be the result of a single type C failure on path (iii) or two simultaneous type C failures on path (i) and (ii). The corresponding simulation results are presented in Table 11. Notice that the latter scenario above actually violates one of the basic

Table 10
Simulation results: a large type B failure off path (iii) or (iv)

| Fault origin | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|
| $f_{22}$ (+10) | −10 | −10 | 0 | −10 | −10 | +10 | +10 | −10 | Y |
| $f_3$ (+10) | +10 | +10 | −10 | −10 | +10 | +10 | +10 | −10 | I |

Table 11
Simulation results: an uncontrollable type A fault off path (iii) and both loops inactive

| Type C failures | Type A fault | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|---|
| $S_5 \xrightarrow{0} X_3$ | $f_3$ (+10) | +10 | +10 | −10 | −10 | +10 | +10 | +10 | −10 | I |
| $X_{22} \xrightarrow{0} S_{12}$, $X_3 \xrightarrow{0} X_{21}$ | $f_3$ (+10) | 0 | 0 | 0 | 0 | +10 | +10 | 0 | +1 | N |
| $X_{22} \xrightarrow{0} S_{12}$, $S_{11} \xrightarrow{0} S_{41}$ | $f_3$ (+10) | +10 | +10 | 0 | 0 | +10 | +10 | 0 | +1 | N |
| $X_{22} \xrightarrow{0} S_{12}$, $X_{21} \xrightarrow{0} S_{11}$ | $f_3$ (+10) | +10 | 0 | 0 | 0 | +10 | +10 | 0 | +1 | N |
| $S_{12} \xrightarrow{0} S_{42}$, $X_3 \xrightarrow{0} X_{21}$ | $f_3$ (+10) | 0 | 0 | 0 | 0 | +10 | +10 | +10 | +1 | N |
| $S_{12} \xrightarrow{0} S_{42}$, $X_{21} \xrightarrow{0} S_{11}$ | $f_3$ (+10) | +10 | 0 | 0 | 0 | +10 | +10 | +10 | +1 | N |
| $S_{12} \xrightarrow{0} S_{42}$, $S_{11} \xrightarrow{0} S_{41}$ | $f_3$ (+10) | +10 | +10 | 0 | 0 | +10 | +10 | +10 | +1 | N |

Table 12
Simulation results: an uncontrollable type A fault off path (iv) and a type C failure on loop II

| Type C failures | Type A fault | $X_{21}$ | $S_{11}$ | $S_{41}$ | $S_5$ | $X_3$ | $X_{22}$ | $S_{12}$ | $S_{42}$ | Override status |
|---|---|---|---|---|---|---|---|---|---|---|
| $X_{22} \xrightarrow{0} S_{12}$ | $f_{22}$ (+10) | (0,0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | +10 | 0 | (+1, +1) | N |
| $S_{12} \xrightarrow{0} S_{42}$ | $f_{22}$ (+10) | (0,0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | +10 | +10 | (+1, +1) | N |
| $S_5 \xrightarrow{0} X_3$ | $f_{22}$ (+10) | 0 | 0 | −10 | −10 | 0 | +10 | +10 | −10 | I |
| $X_3 \xrightarrow{0} X_{22}$ | $F_{22}$ (+10) | −10 | −10 | 0 | −10 | −10 | +10 | +10 | −10 | Y |

assumptions of this study, i.e. the simultaneous occurrence of two or more type C failures should be ignored. Such possibilities are still studied here for the sake of completeness. The user can exclude them in practical applications on a case-by-case basis.

- *The affected loop variable is on path (iv).* The effects of these disturbances are capable of activating loop II. As a result, the controlled variable $X_{22}$ must be brought to the new set point, i.e. +1, if the overriding mechanism is functional. Thus, in order to ensure the occurrence of top event $X_{22}(+10)$, a type C failure must exist simultaneously on loop II. This conclusion can be clearly observed from the simulation results presented in Table 12.

### 7.2. The causes with upstream disturbances

As mentioned before, the causes of top event in this case can actually be viewed as the causes of two simultaneous events, $S_{41}(+10)$ and $S_{42}(+10)$. Thus, the corresponding faults/failures can be identified by tracing along paths (i) and (ii) separately. Let us consider them in turn.

#### 7.2.1. The faults/failures along path (i)

Under the condition that $S_{42} = +10$, the causes of $S_{41}(+10)$ can be obtained by building a fault-tree under it with the conventional digraph-based approach [1,2]. This tree can be constructed simply by treating loop I as an isolated NFBL.

#### 7.2.2. The faults/failures along path (ii)

Since the set point of loop II is higher than the normal value of its controlled variable, the causes of $S_{42}(+10)$ cannot be identified with the same approach as in the previous case. It is thus necessary to evaluate the effects of various faults and failures on path (ii) under the condition that $S_{41} = +10$. The results of qualitative simulation are presented in Tables 13–16. The effects of type A faults can be found in Tables 13 and 14. Notice that only the type A faults affecting the downstream variables of $X_{22}$, i.e. $f_{12}$ and $f_{42}$, can cause both $S_{42}(+10)$ and $X_{22}(+10)$. The same conclusion can be made concerning the effects of type B failures (see Table 15). The effects of a single type C failure on path (ii) are described in Table 16. Notice that, since the set point of loop II is always higher than the sensor output $S_{12}$, a constant error input to the controller is bound to

Table 13
Simulation results: a controllable type A fault occurs off path (ii) under the condition that $S_{41} = +10$

| Fault origin | $X_{22}$ | $S_{12}$ | $S_{42}$ | $S_5$ | $X_3$ | Override status |
|---|---|---|---|---|---|---|
| $f_{22}$ (+1) | (+1, +1) | (+1, +1) | (0, 0) | (0, 0) | (0, 0) | Y |
| $f_{12}$ (+1) | (0, 0) | (+1, +1) | (0, 0) | (0, 0) | (0, 0) | Y |
| $f_{42}$ (+1) | (+10, +1) | (+10, +1) | (+10, +1) | (+10, +1) | (+10, +1) | Y |
| $f_{22}$ (−1) | (−1, +1) | (−1, +1) | (+10, +10) | (+10, +10) | (+10, +10) | I |
| $f_{12}$ (−1) | (+10, +10) | (−1, +1) | (+10, +10) | (+10, +10) | (+10, +10) | I |
| $f_{42}$ (−1) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | (0, 0) | Y |

Table 14
Simulation results: an uncontrollable type A fault occurs off path (ii) under the condition that $S_{41} = +10$

| Fault origin | $X_{22}$ | $S_{12}$ | $S_{42}$ | $S_5$ | $X_3$ | Override status |
|---|---|---|---|---|---|---|
| $f_{22}$ (+10) | (+10, +1) | (+10, +1) | (−1, −1) | (−1, −1) | (−1, −1) | Y |
| $f_{12}$ (+10) | (−1, −1) | (+10, +1) | (−1, −1) | (−1, −1) | (−1, −1) | Y |
| $f_{42}$ (+10) | (+10, +10) | (+10, +10) | (+10, +10) | (+10, +10) | (+10, +10) | I |
| $f_{22}$ (−10) | (−10, −1) | (−10, −1) | (+10, +10) | (+10, +10) | (+10, +10) | I |
| $f_{12}$ (−10) | (+10, +10) | (−10, −1) | (+10, +10) | (+10, +10) | (+10, +10) | I |
| $f_{42}$ (−10) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | (−1, −1) | Y |

be created in this case. Due to the integration action of the controller, its output $S_{42}$ should eventually be driven to the maximum value +10.

Finally, it can be observed from Table 13 (rows 4 and 5) that if $S_{41} = +10$, the relation between the controller output $S_{42}$ and the sensor output $S_{12}$ can be represented with Table 17. In addition, the relation between the other variables on path (ii) can be described by Table 18.

## 8. The generalized fault-tree structures for top event $X_{22}(+10)$

The analysis of simulation results in Section 7 can be summarized with eight generalized fault-tree structures. They are described in detail in the sequel.

Let us first consider the fault-tree structure presented in Fig. 9. Notice that the format of structure B is essentially the same as that of structure A (see Fig. 8). In developing the fault-tree for a given override control system, this structure is applicable to a large deviation in the current output variable corresponding to a node on path (iv) (except the starting node). Substructure B-1 is used to incorporate the causes of top event under the condition that LS is functional. The sub tree under B-1 can be further expanded with structure C (Fig. 10). On the other hand, if the LS fails, the fault-tree can be developed under substructure B-2. It is assumed in this work that there are only two possible LS failures, i.e. (1) LS fails to override and (2) LS sticks. In both cases, the resulting digraph configurations are *not* dependent upon the values of its inputs (see Table 2). Consequently, the existing fault-tree structures [1,2] are applicable in this situation.

As mentioned earlier, structure C can be used to develop fault-tree under an output, which is associated with a non-starting node on path (iv). Its direct inputs are organized in three substructures. They are described in a left-to-right order as follows. The uncontrollable effects of large type B failures are included in the first substructure. The corresponding simulation results can be found in the first row of Table 10. The second substructure reflects the scenarios presented in Table 12, i.e. the combined effects of a large type A fault off path (iv) and a type C failure on loop II. To facilitate concise description of this substructure, two

Table 15
Simulation results: a moderate type B failure occurs off path (ii) under the condition that $S_{41} = +10$

| Fault origin | $X_{22}$ | $S_{12}$ | $S_{42}$ | $S_5$ | $X_3$ | Override status |
|---|---|---|---|---|---|---|
| $f_{22}$ (+1) | +1 | +1 | 0 | 0 | 0 | N |
| $f_{12}$ (+1) | 0 | +1 | 0 | 0 | 0 | N |
| $f_{42}$ (+1) | +10 | +10 | +10 | +10 | +10 | I |
| $f_{22}$ (−1) | −1 | −1 | +10 | +10 | +10 | I |
| $f_{12}$ (−1) | +10 | −1 | +10 | +10 | +10 | I |
| $f_{42}$ (−1) | 0 | 0 | 0 | 0 | 0 | N |

Table 16
Simulation results: a type C failure occurs on path (ii) under the condition that $S_{41} = +10$

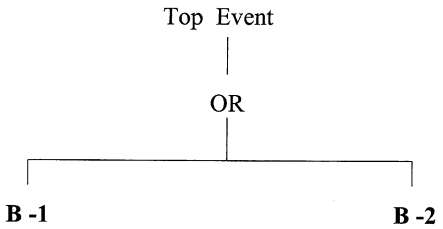
| Failure location | $X_{22}$ | $S_{12}$ | $S_{42}$ | $S_5$ | $X_3$ | Override status |
|---|---|---|---|---|---|---|
| $X_{22} \xrightarrow{0} S_{12}$ | +10 | 0 | +10 | +10 | +10 | I |
| $S_{12} \xrightarrow{0} S_{42}$ | +10 | +10 | +10 | +10 | +10 | I |

Table 17
Additional propagation patterns between the sensor output and controller output on loop II

| Set point | Gain | Sensor output | Controller output |
|---|---|---|---|
| +1 | +1 | (+1, −1) | (+10, +10) |
| | −1 | (−1, +1) | (+10, +10) |
| −1 | −1 | (+1, −1) | (−10, −10) |
| | +1 | (−1, +1) | (−10, −10) |

Table 18
Additional propagation patterns between the variables on loop II (except the controller output)

| Gain | Input | Output |
|---|---|---|
| +1 | (+1, −1) | (+1, −1) |
| | (−1, +1) | (−1, +1) |
| −1 | (+1, −1) | (−1, +1) |
| | (−1, +1) | (+1, −1) |




**Remarks**: The top event considered in this fault tree is $X_0(\pm 10)$ corresponding to a node on path (iv). The sub-tree B-1 can be further expanded with structure C. The valid failure modes included under the sub-tree B-2 should be (1) "LS stuck"and (2)"LS fails to override." In both cases, the traditional algorithms are applicable since the corresponding system digraphs are not affected by the inputs to the selector.

Fig. 9. Structure B.

special terms, incidence node and feedback path, are introduced in this work. Their respective definitions are

- *Incidence node*: The first NFBL node encountered in the digraph-based fault-tree synthesis process.
- *Feedback path*: A path on NFBL which starts at the incidence node and ends at the node representing the current output.

Finally, the third substructure in structure C is designed to trace the causes of a deviation in the current output along path (iv).

Notice that there is a need to further develop the branch under the third substructure of structure C if its local input on path (iv) is the starting node $X_3$. Since $X_3$ is also the terminal node of path (iii), structure D in Fig. 11 can be

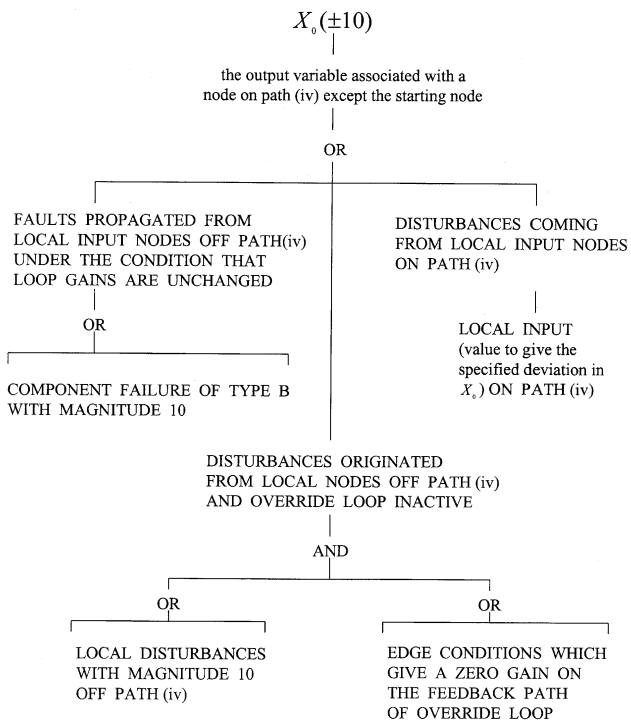



Fig. 10. Structure C.

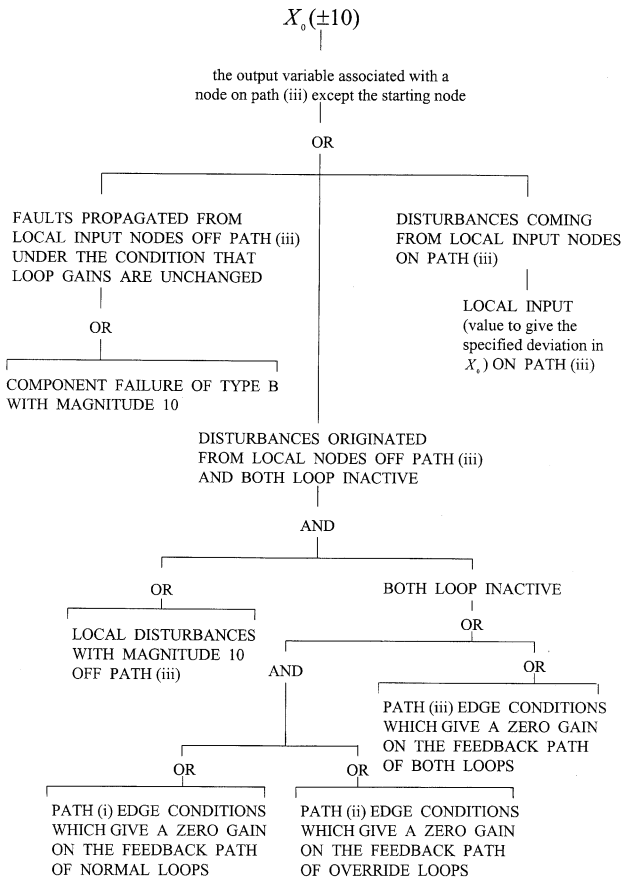



Fig. 11. Structure D.

utilized for this task. This structure is also arranged in three parts. The first substructure on the left can be used to describe the effects of local type B failures. The corresponding simulation results can be found in the second row of Table 10. The failure mechanisms listed in Table 11, i.e. the results of a large local type A fault and both the loops inactive, are summarized in the second substructure. The substructure on the right is introduced to perform the same task as its counterpart in structure C. In other words, it is used to identify the fault propagation patterns along path (iii).

Structure E is developed solely for the purpose of building fault-tree under selector output (see Fig. 12). This structure is needed in developing the third substructure in structure D along path (iii) when the local input is the starting node $S_5$. There are two levels of inputs below the current output in structure E. The inputs in the first is connected to the current output with an AND gate. As explained earlier, the two inputs of LS must both reach $+10$ to maximize its output. Notice that $-10$ is included as a possible value of the local inputs and current output. This is due to the need to apply the generalized structures to override control systems with high selectors. As suggested in Section 7.2 that, if the value of LS input on path (ii) is $+10$, loop I can be treated as an isolated simple NFBL. Thus, the right branch in the first level can be further

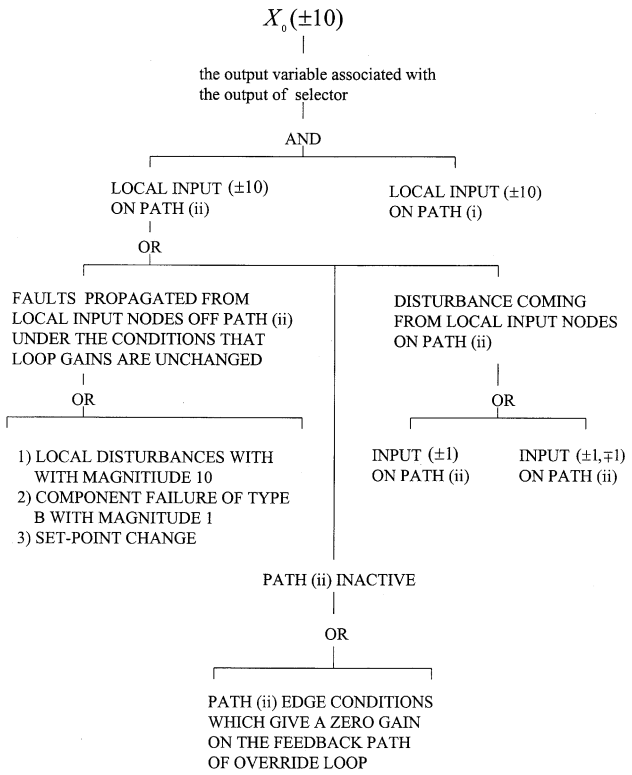



Fig. 12. Structure E.

developed according to the existing fault-tree structures [1,2]. On the other hand, the left branch in the first level of structure E should be connected with the third-level inputs using an OR gate. Notice that these inputs are organized in three substructures. The substructure on the left is simply an alternative description of the scenarios listed in Tables 14 and 15. The one in the middle represent the failure mechanisms included in Table 16, i.e. the controller on loop II is saturated by a type C failure on path (ii). The substructure on the right is used to trace the causes of a deviation in the current output along path (ii). There are two possibilities

1. *The input value on path (ii) is −1 or +1.* If LS is used in the override control system, the set point of loop II is higher than the normal value of controlled variable. From the fifth row of Table 15, it can be seen that a type B failure on path (ii) is capable of driving the controller output to maximum. A simple structure F (Fig. 13) has been developed to incorporate such scenarios in the fault-tree.
2. *The input value on path (ii) is (−1, +1) or (+1, −1).* Again, as a result of the higher-than-normal set point on loop II, it can be observed from rows 4 and 5 in Table 13 that the effects of a moderate type A fault may propagate along path (ii). These fault propagation behaviors can be described with structure G in Fig. 14. The values of local inputs on path (ii) can be determined according to Tables 17 and 18.

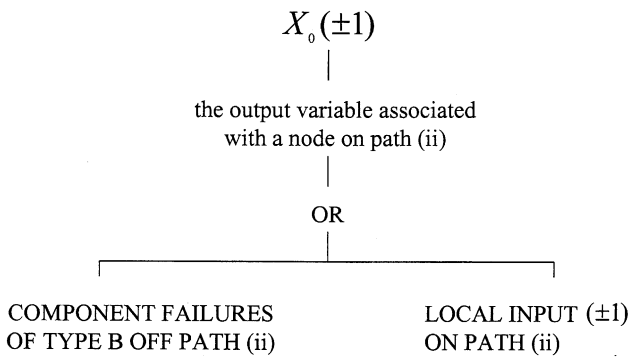



Fig. 13. Structure F.

## 9. Applications

To demonstrate the correctness of the earlier generalized fault-tree structures, they have been applied to the furnace control system described in Fig. 3. The two top events chosen for this example are $T_2(+1)$, i.e. the tube-surface temperature reaches allowable upper limit, and $T_2(+10)$, i.e. the tube-surface temperature is significantly higher than the upper limit. Notice that there are two negative feed forward loops in this system, i.e. Eqs. (1) and (2). Let us assume that the net effects of disturbances propagating through their respective starting nodes have already been evaluated in advance. More specifically, it has been determined that a moderate change in $T_1$ or $m_1$ does not create noticeable change in $T_2$ and, also, $T_1(-10)$ or $m_1(+10)$ causes only a moderate positive deviation in $T_2$, i.e. $T_2(+1)$.

The fault-tree with top event $T_2(+1)$ can be constructed according to structure A in Fig. 8. The fully developed branches under A-1 and A-2 can be found in Figs. 15 and 16, respectively. Under the condition that LS is functional, the sub-tree below A-1 can be developed on the basis of a fictitious digraph corresponding to the one presented in Fig. 7. If the LS sticks, it is clear from Table 2 that the system digraph does not contain loops. The sub-tree under A-2 should be constructed according to the definitions of arc

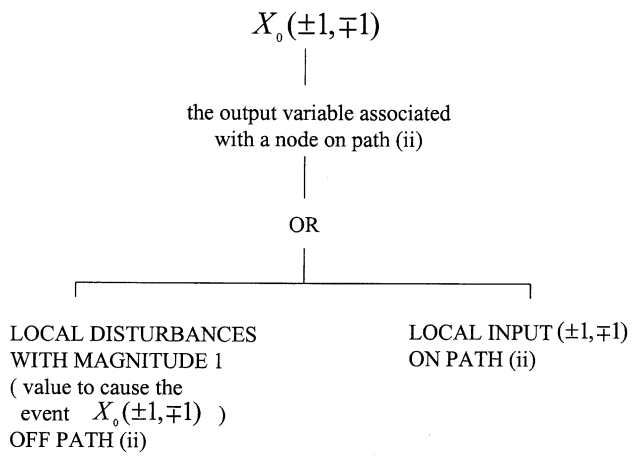



Fig. 14. Structure G.

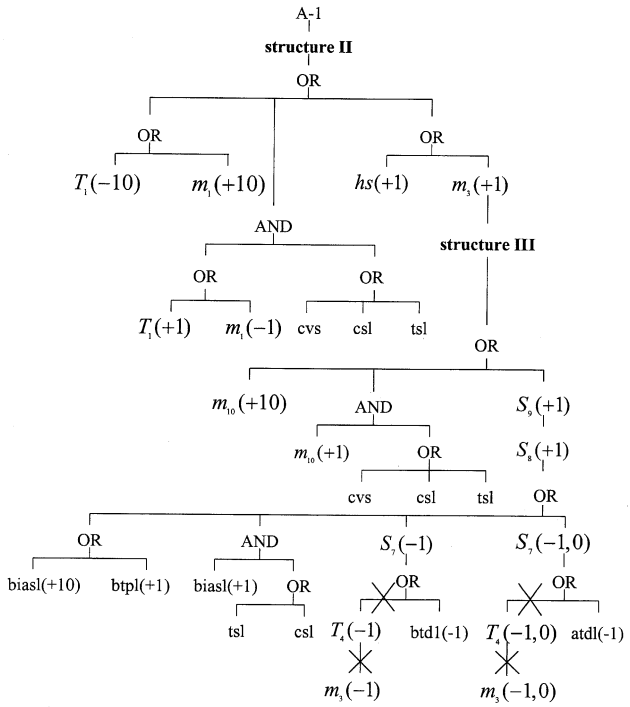



Fig. 15. The fault-tree corresponding to the top event $T_2(+1)$ in the furnace control system: sub-tree A-1.

gains. In both cases, the existing techniques are applicable. In particular, three conventional fault-tree structures, i.e. I, II and III, have been used to build these two sub trees. Since a detailed explanation of these structures can be found [1,2] elsewhere, they are not elaborated in this paper for the sake of brevity. Finally, the minimal cut sets of the above fault-tree are listed in Table 19. Notice that none of the faults/failures on path (ii) can be the root causes of $T_2(+1)$ and, also, the type C failures included in the cut sets are all located on loop I. This is the result of using the fictitious digraph for generating sub-tree A-1.

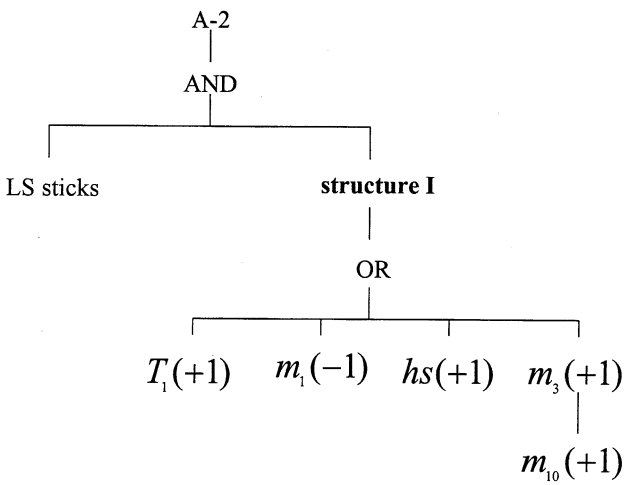



Fig. 16. The fault-tree corresponding to the top event $T_2(+1)$ in the furnace control system: sub-tree A-2.

Table 19
The minimal cut sets of the fault-tree with top event $T_2(+1)$

| Set no. | Basic events | | Set no. | Basic events | |
|---|---|---|---|---|---|
| 1 | $T_1$ (−10) | | 13 | $m_1$ (−1) | *Cvs* |
| 2 | $m_1$ (+10) | | 14 | $m_1$ (−1) | *cs*1 |
| 3 | *hs* (+1) | | 15 | $m_1$ (−1) | *Ts*1 |
| 4 | $m_{10}$ (+10) | | 16 | $m_1$ (−1) | *LS stk* |
| 5 | *bias*1 (+10) | | 17 | $m_{10}$ (+1) | *cvs* |
| 6 | *btp*1 (+1) | | 18 | $m_{10}$ (+1) | *cs*1 |
| 7 | | | 19 | $m_{10}$ (+1) | *ts*1 |
| 8 | *atd*1 (−1) | | 20 | $m_{10}$ (+1) | *LS stk* |
| 9 | $T_1$ (+1) | *cvs* | 21 | *bias*1(+1) | *cs*1 |
| 10 | $T_1$ (+1) | *cs*1 | 22 | *bias*1(+1) | *ts*1 |
| 11 | $T_1$ (+1) | *ts*1 | | | |
| 12 | $T_1$ (+1) | *LSstk* | | | |

On the other hand, the fault-tree with top event $T_2(+10)$ can be synthesized on the basis of structure B in Fig. 9. The sub-tree under B-1 can be found in Fig. 17 and those under B-2 are presented in Figs. 18 and 19. The former is obtained under the condition that LS is functional. The corresponding digraph can be defined by Fig. 4 and Table 2. Thus,

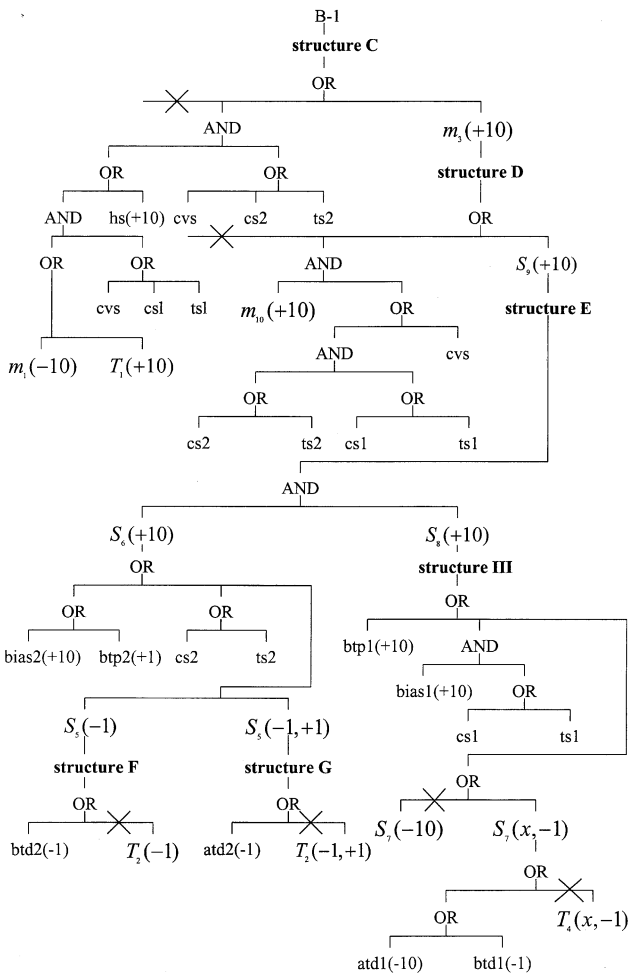



Fig. 17. The fault-tree corresponding to the top event $T_2(+10)$ in the furnace control system: sub-tree B-1.

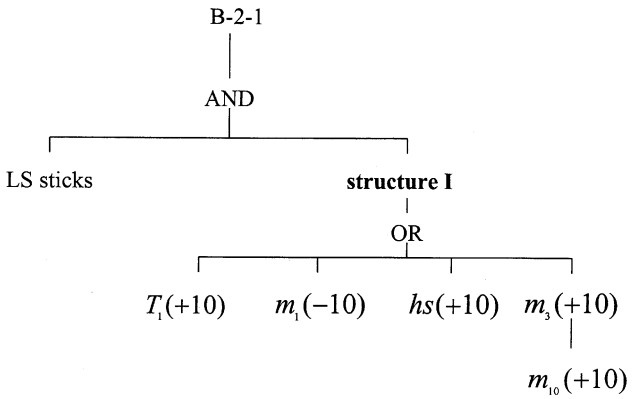



Fig. 18. The fault-tree corresponding to the top event $T_2(+10)$ in the furnace control system: sub-tree B-2-1.

the generalized fault-tree structures proposed in this paper, i.e. structures C–G, should be utilized for building sub-tree B-1. The fault propagation patterns in a system with failed selector are supposed to be described in sub-trees B-2-1 (LS sticks) and B-2-2 (LS fails to override). As explained earlier, the traditional algorithms are applicable in these situations. The resulting minimal cut sets are presented in Table 20. It can be observed that, in every cut set, there is at least one fault or failure originated from loop II. This result shows that the probability of hazardous top event $T_2(+10)$ is almost nil under the condition that override loop functions

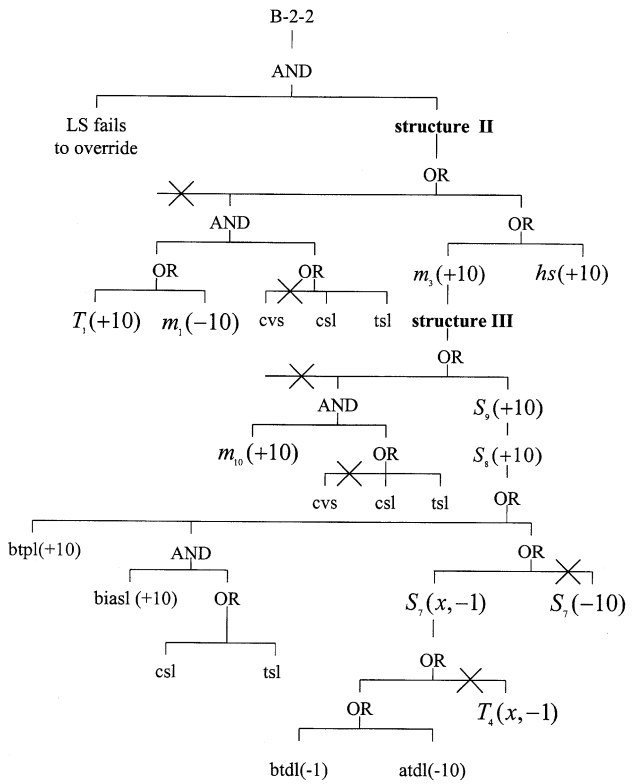



Fig. 19. The fault-tree corresponding to the top event $T_2(+10)$ in the furnace control system: sub-tree B-2-2.

Table 20
The minimal cut sets of the fault-tree with top event $T_2$ (+10)

| Set no. | Basic events | | Set no. | Basic events | | |
|---|---|---|---|---|---|---|
| 1 | *hs*(+10) | *cvs* | 33 | $m_{10}$ (+10) | *ts*1 | *ts*2 |
| 2 | *hs*(+10) | *cs*2 | 34 | $m_{10}$ (+10) | ts1 | cs2 |
| 3 | *hs*(+10) | *ts*2 | 35 | $m_{10}$ (+10) | *cs*1 | ts2 |
| 4 | *hs*(+10) | *LS stk* | 36 | $m_{10}$ (+10) | *cs*1 | cs2 |
| 5 | *hs*(+10) | *LS fto* | 37 | $m_{10}$ (+10) | *ts*1 | *LS fto* |
| 6 | $T_1$ (+10) | *cvs* | 38 | $m_{10}$ (+10) | *cs*1 | *LS fto* |
| 7 | $T_1$ (+10) | *LS stk* | 39 | $T_1$ (+10) | *ts*1 | *ts*2 |
| 8 | $m_1$ (−10) | *cvs* | 40 | $T_1$ (+10) | *ts*1 | *cs*2 |
| 9 | $m_1$ (−10) | *LS stk* | 41 | $T_1$ (+10) | *cs*1 | *ts*2 |
| 10 | $m_{10}$ (+10) | *cvs* | 42 | $T_1$ (+10) | *cs*1 | *cs*2 |
| 11 | $m_{10}$ (+10) | *LS stk* | 43 | $T_1$ (+10) | *ts*1 | *LS fto* |
| 12 | *Btp*1(+10) | *bias*2(+10) | 44 | $T_1$ (+10) | *cs*1 | *LS fto* |
| 13 | *btp*1(+10) | *btp*2(+1) | 45 | $m_1$ (−10) | *ts*1 | *ts*2 |
| 14 | *btp*1(+10) | *btd*2(−1) | 46 | $m_1$ (−10) | *ts*1 | *cs*2 |
| 15 | *btp*1(+10) | *atd*2(−1) | 47 | $m_1$ (−10) | *cs*1 *ts*2 | |
| 16 | *btp*1(+10) | *cs*2 | 48 | $m_1$ (−10) | *cs*1 | *cs*2 |
| 17 | *btp*1(+10) | *ts*2 | 49 | $m_1$ (−10) | *ts*1 | *LS fto* |
| 18 | *btp*1(+1) | *LS fto* | 50 | $m_1$ (−10) | *cs*1 | *LS fto* |
| 19 | *atd*1(−10) | *bias*2(+10) | 51 | *bias*1(+10) | *ts*1 | *ts*2 |
| 20 | *atd*1(−10) | *btp*2(+1) | 52 | *bias*1(+10) | *ts*1 | *cs*2 |
| 21 | *atd*1(−10) | *btd*2(−1) | 53 | *bias*1(+10) | *cs*1 | *ts*2 |
| 22 | *atd*1(−10) | *atd*2(−1) | 54 | *bias*1(+10) | *cs*1 | *cs*2 |
| 23 | *atd*1(−10) | *cs*2 | 55 | *bias*1(+10) | *ts*1 | *bias*2(+10) |
| 24 | *atd*1(−10) | *ts*2 | 56 | *bias*1(+10) | *cs*1 | *bisa*2(+10) |
| 25 | *atd*1(−10) | *LS fto* | 57 | *bias*1(+10) | *ts*1 | *btp*2(+1) |
| 26 | *btd*1(−1) | *bias*2(+10) | 58 | *bias*1(+10) | *cs*1 | *btp*2(+1) |
| 27 | *btd*1(−1) | *btp*2(+1) | 59 | *bias*1(+10) | *ts*1 | *btd*2(−1) |
| 28 | *btd*1(−1) | *btd*2(−1) | 60 | *bias*1(+10) | *cs*1 | *btd*2(−1) |
| 29 | *btd*1(−1) | *atd*2(−1) | 61 | *bias*1(+10) | *ts*1 | *atd*2(−1) |
| 30 | *btd*1(−1) | *cs*2 | 62 | *bias*1(+10) | *cs*1 | *atd*2(−1) |
| 31 | *btd*1(−1) | *ts*2 | 63 | *bias*1(+10) | *ts*1 | *LS fto* |
| 32 | *btd*1(−1) | *LS fto* | 64 | *bias*1(+10) | *cs*1 | *LS fto* |

properly. In other words, the tube-surface temperature can only be significantly higher than the allowable upper limit if the regulatory mechanism of loop II is destroyed by faults/failures.

## 10. Conclusions

The digraph configuration of override control systems has been rigorously analyzed in this work. On the basis of qualitative simulation of the fault propagation patterns in digraph, the corresponding generalized fault-tree structures have also been established. It is clear that some of the unique failure mechanisms included in these fault-tree structures are not identifiable with any of the existing techniques.

The potential for computerization of the proposed fault-tree synthesis algorithm is obvious. In addition, if implemented manually, this procedure forces a structured approach whereby different users are more likely to produce fault-trees of consistent logic.

## Appendix A. Classification of faults and failures

The definitions of faults and failures suggested by Himmelblau [12] are followed in this work. 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, i.e.

### *A.1. 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 arcs of such nodes are directed to process variables. A typical digraph model can be found in Fig. A1, where $x_1$ and $x_2$ are process variables and $f$ is the fault of type A. The effects of this type of faults/failures can be determined by assigning a non-zero value ($\pm 1$ or $\pm 10$) to $f$ and the values of the other variables in the digraph can then be evaluated, respectively. 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 $x_1$ and $x_2$ are on the same FBL, the value of $x_2$ can be affected not only by $f$ but also by $x_1$.

### *A.2. Type B*

The digraph configuration of component failures such as sensor failing high or control valve failing close is actually

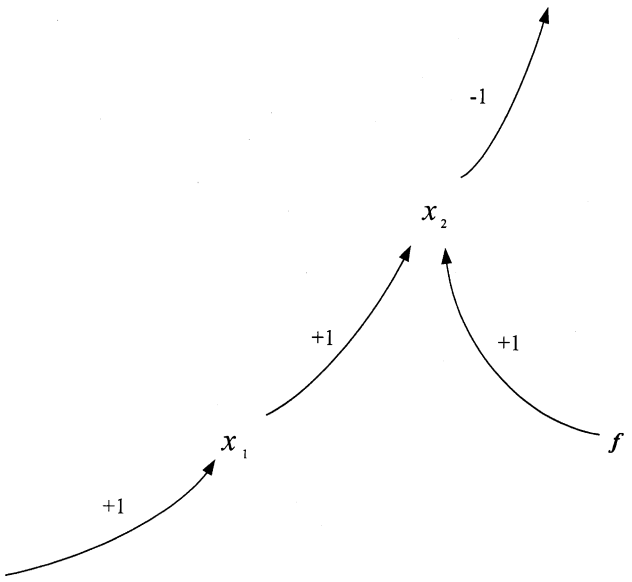



Fig. A1. The digraph model of type A faults.

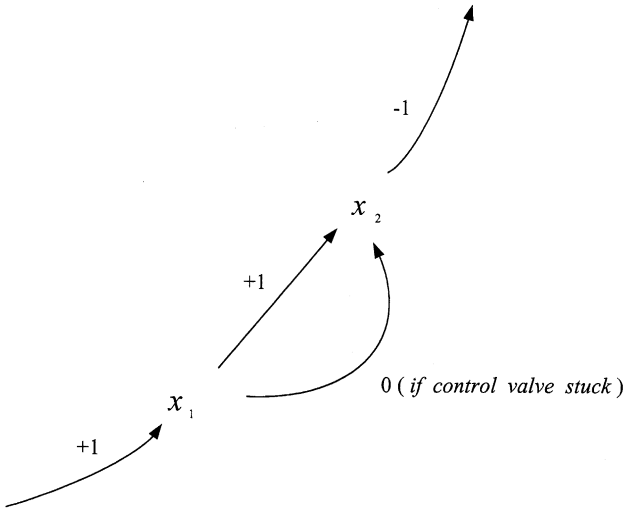



Fig. A2. The digraph model of type C failures.

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 $x_1$.

### *A.3. Type C*

Component failures such as sensor stuck or control valve stuck should be modeled by conditional arcs with zero gain. An example can be found in Fig. A2. The occurrence of a failure of this type only changes the configuration of the system digraph, i.e. the arc between $x_1$ and $x_2$ can be considered as non-existence. The state variables of the system remain at the normal levels without additional disturbances.

### *A.4. Type D*

Component failures such as controller reversed (from direct action to reverse action or vice versa) or control valve reversed (from air-to-open to air-to-close or vice versa) can also be represented by conditional arcs. An example of such failures is presented in Fig. A3, which is also represented by a change in configuration. 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 $x_2$.

## Appendix B. The fault propagation patterns in a single NFBL

The fault propagation patterns in a single NFBL can be determined with qualitative simulation techniques. Let us consider the standard NFBL presented in Fig. B1. In this figure, $S_1$ is the sensor signal, $X_2$ represents the controlled variable, $X_3$ is the manipulated variable, $S_4$ denotes

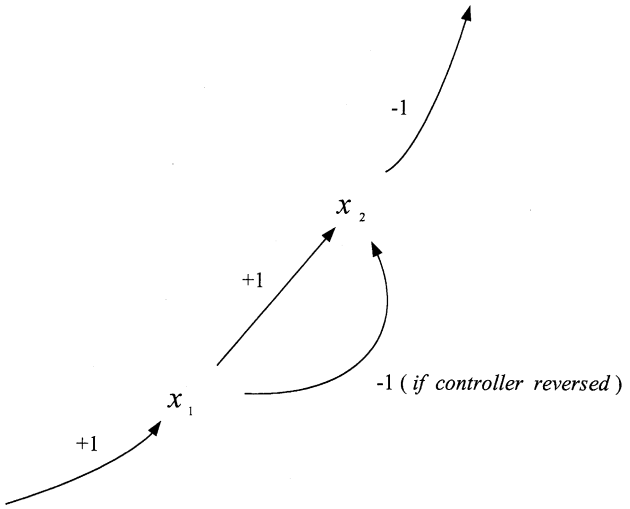



Fig. A3. The digraph model of type D failures.

the output signal from the controller and the nodes $f_1, f_2, f_3$ and $f_4$ are used to represent type A faults and/or type B failures. In addition, to facilitate illustration of the phenomena caused by an external disturbance and/or an equipment failure, let us define an *event symbol* $N(v)$ to represent the abnormal condition associated with a node on the fault propagation path. Here, $N$ denotes the node label and $v$ is its qualitative value. In other words, this symbol denotes the event $N = v$.

Three types of scenarios are described in the sequel

### *B.1. The effects of a type A fault*

As mentioned before, any disturbance to a NFBL generates two opposite effects on the incidence loop variable. For example, although the gain of the edge between $f_2$ and $X_2$ is positive, the product of the gains on the path $f_2 \to X_2 \to S_1 \to S_4 \to X_3 \to X_2$ is negative. The net effect is zero if the control loops functions properly. In other words, the event $f_2(+1)$ causes $X_2(0)$, $S_1(0)$, $S_4(+1)$ and $X_3(+1)$ at new steady state. This special behavior of NFBLs creates a problem in simulating fault propagation, i.e. the cause–effect relations are not consistent with individual edge gains specified in the digraph.

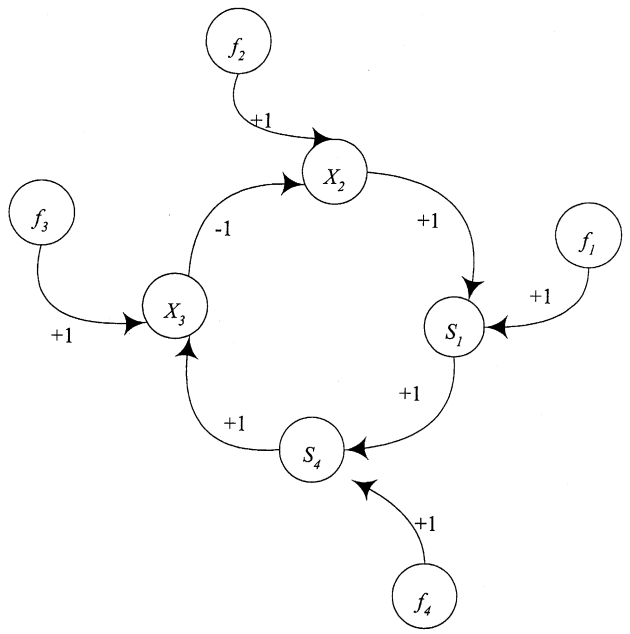



Fig. B1. The digraph configuration of a typical NFBL.

To overcome this problem, Hwang and Chang [1] suggested that the states of loop variables can be represented with symbols of the form $(v_0, v_\infty)$. This symbol can be regarded as the state of a loop variable which would have a value $v_0$ without feedback but approaches $v_\infty$ at the new steady state due to regulatory action. Thus, due to its integral action, the digraph model of PID controller in NFBLs can really be interpreted according to Table B1. Consequently, the effects of a type A fault corresponding to $f_2$ can be described with a set of modified event symbols, i.e.

$$\{X_2(+1,0), S_1(+1,0), S_4(+1,+1), X_3(+1,+1)\} \tag{B1}$$

Furthermore, the implied fault propagation sequence can be expressed explicitly in terms of a precedence order, i.e.

$$f_2(+1) \prec X_2(+1) \prec S_1(+1) \prec S_4(+1) \prec X_3(+1) \prec X_2(0) \prec S_1(0) \prec S_4(+1) \prec X_3(+1) \tag{B2}$$

Here, the symbol $\prec$ is used to represent the direct causal relation between two abnormal events, i.e. $E_1 \prec E_2$ means event $E_1$ precedes event $E_2$.

The patterns of deviations in the loop variables caused by disturbances at various locations are summarized in Table B2. Several interesting features can be observed from this table, i.e. (1) a sub path is formed by the loop variables with values $(v_0, 0)$, (2) the starting node of this sub path is the incidence node, and (3) the terminal node is always the one corresponding to a sensor output.

Table B1
New interpretations of gains between sensor outputs and controller outputs

| Gain | Sensor output | Controller output |
|---|---|---|
| +1 | (+1, 0) | (+1, +1) |
| | (−1, 0) | (−1, −1) |
| −1 | (+1, 0) | (−1, −1) |
| | (−1, 0) | (+1, +1) |

Table B2
Fault propagation patterns in a single NFBL—a type A fault of value +1

| Fault origin | $S_1$ | $X_2$ | $X_3$ | $S_4$ |
|---|---|---|---|---|
| $f_1$ (+1) | (+1, 0) | (−1, −1) | (+1, +1) | (+1, +1) |
| $f_2$ (+1) | (+1, 0) | (+1, 0) | (+1, +1) | (+1, +1) |
| $f_3$ (+1) | (−1, 0) | (−1, 0) | (+1, 0) | (−1, −1) |
| $f_4$ (+1) | (−1, 0) | (−1, 0) | (+1, 0) | (+1, 0) |

Table B3
Fault propagation patterns in a single NFBL—a type A fault of value +10

| Fault origin | $S_1$ | $X_2$ | $X_3$ | $S_4$ |
|---|---|---|---|---|
| $f_1$ (+10) | (+10, +1) | (−10, −10) | (+10, +10) | (+10, +10) |
| $f_2$ (+10) | (+10, +1) | (+10, +1) | (+10, +10) | (+10, +10) |
| $f_3$ (+10) | (−10, −1) | (−10, −1) | (+10, +1) | (−10, −10) |
| $f_4$ (+10) | (−10, −1) | (−10, −1) | (+10, +1) | (+10, +1) |

A similar analysis can be carried out for disturbances of magnitude 10. The value 10 in this study is regarded as a 'very large' quantity which would saturate the control loop [2]. A summary of the corresponding fault propagation patterns is presented in Table B3. Since the loop is saturated, the effects generated by a disturbance with magnitude 10 cannot be cancelled with regulatory action and a non-zero deviation always occurs in the sensor output.

### *B.2. The effects of a type B failure*

From the definitions presented in Appendix A, it is clear that the digraph representation of a component failure of type B is essentially equivalent to that of simultaneous occurrence of a type C failure and a local disturbance. Since in this case the NFBL is broken and also the value of incidence loop variable is fixed at +1 (or −1), the fault propagation pattern can be determined on the basis of the resulting simple digraph without feedback. For example, the simulation result corresponding to a type B failure at $f_3$ can be expressed as

$$f_3(+1) \prec X_3(+1) \prec X_2(-1) \prec S_1(-1) \prec S_4(-10) \qquad \text{(B3)}$$

Notice that, due to the integral action in controller, the value of $S_4$ should reach −10 eventually. A summary of the deviation patterns for type B failures is presented in Table B4.

### *B.3. The combined effects of a type A fault and a type C failure*

As mentioned previously, a type C failure is represented with a conditional edge with zero gain. If it occurs in the control system, the regulatory action in NFBL is essentially lost. In other words, the corresponding feedback loop should be broken due to such a failure. The combined effects of a type A fault and a type C failure can be evaluated by determining the fault propagation behavior in the resulting digraph. For example, the simulation result of a type A fault at $f_4$ and a type C failure between $X_2$ and $S_1$ is

$$f_4(+1) \prec S_4(+1) \prec X_3(+1) \prec X_2(-1) \prec S_1(0) \qquad \text{(B4)}$$

Since the effects of other combinations of type A faults and type C failures can be determined easily in a straightforward fashion, the corresponding simulation results are omitted for the sake of brevity.

Table B4
Fault propagation patterns in a single NFBL—a type B failure of value +1

| Fault origin | $S_1$ | $X_2$ | $X_3$ | $S_4$ |
|---|---|---|---|---|
| $f_1$ (+1) | +1 | −10 | +10 | +10 |
| $f_2$ (+1) | +1 | +1 | +10 | +10 |
| $f_3$ (+1) | −1 | −1 | +1 | −10 |
| $f_4$ (+1) | −1 | −1 | +1 | +1 |

## References


[1] Chang CT, Hwang HC. New developments of the digraph-based techniques for fault-tree synthesis. Ind Engng Chem Res 1992;31:1490.
[2] Lapp SA, Powers GJ. Computer-aided synthesis of fault-trees. IEEE Trans Reliab 1977;R-26:2.
[3] Shaeiwitz JA, Lapp SA, Powers GJ. Fault-tree analysis of sequential systems. Ind Engng Chem Proc Des Dev 1977;16:529.
[4] Chamow MF. Directed graph techniques for the analysis of fault-trees. IEEE Trans Reliab 1978;R-27:7.
[5] Lambert HE. Comments on the lapp-powers 'Computer-aided synthesis of fault-trees'. IEEE Trans Reliab 1979;R-28:6.
[6] Lapp SA, Powers GJ. Update of lapp-powers fault-tree synthesis algorithm. IEEE Trans Reliab 1979;R-28:12.
[7] Allen DJ, Rao MSM. New algorithms for the synthesis and analysis of fault-trees. Ind Engng Chem Fundam 1980;19:79.
[8] Cummings DL, Lapp SA, Powers GJ. Fault-tree synthesis from a directed graph model for a power distribution network. IEEE Trans Reliab 1983;R-32:140.
[9] Allen DJ. Digraphs and fault-trees. Ind Engng Chem Fundam 1984;23:175.
[10] Andrews JD, Morgan JM. Application of digraph method of fault-tree construction to process plant. Reliab Engng 1986;14:85.
[11] Andrews JD, Brennan G. Application of the digraph method of fault-tree construction to a complex control configuration. Reliab Engng Syst Safety 1990;28:357.
[12] Himmelblau DM. Fault detection and diagnosis in chemical and petrochemical processes. New York: Elsevier; 1978. pp. 2–10.