

# Systematic Enumeration of Fuzzy Diagnosis Rules for Identifying Multiple Faults in Chemical Processes

Jung Yang Chen and Chuei-Tin Chang\*

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

A signed directed graph (SDG)-based computation procedure is proposed in this paper to predict the effects of one or more fault propagating in a multiloop process system. The conventional version of qualitative simulation techniques<sup>1</sup> is modified to identify not only the locations of fault origins but also their magnitude levels. In addition, a computer algorithm is presented to generate the IF–THEN inference rules automatically according to the anticipated fault propagation behaviors. The effectiveness and feasibility of this approach has been tested with three case studies. Two of them are concerned with level control systems and the other an exothermic continuously stirred tank reactor (CSTR) with temperature and level control loops.

## 1. Introduction

Due to the need to minimize operating costs while maintaining economic production scale, the chemical plants built in recent years are in general much larger and more complex than they used to be. Furthermore, their processing units are often designed to be operated under more extreme conditions. Thus, the development of hazard identification and risk reduction measures for such processes becomes an issue of major concern. To this end, the on-line fault diagnosis system should be considered as an indispensable tool. Notice that many different methods have already been proposed in the literature, e.g., the state estimator,<sup>2</sup> the expert system,<sup>3,4</sup> the neural network,<sup>5</sup> the signed directed graph (SDG),<sup>6–12</sup> the principal component analysis (PCA),<sup>13</sup> the frequency-domain analysis,<sup>14,15</sup> etc. Generally speaking, these methods could be classified into three distinct groups,<sup>16–18</sup> i.e., the model-based approaches, the knowledge-based approaches, and the data-analysis-based approaches.

The SDG-based fault diagnosis strategy is the focus of present study. In essence, the digraph models have been used in the previous applications to qualitatively characterize the causal relations among faults, failures, and their effects. The advantage of this modeling approach is mainly due to the fact that the SDG can almost always be constructed according to general engineering principles. On the other hand, the more accurate mathematical models and the more case-specific knowledge bases are required to be built from the measurement data and operational experiences obtained in the course of every possible accident. This need is often not satisfiable.

Although the SDG models are easy to develop, it should be noted that they are static in nature. Consequently, the available fault identification techniques are implemented mostly on the basis of steady-state symptoms, e.g., see the work of Rengasamy et al.<sup>19</sup> However, the effects of fault(s) and/or failure(s) usually propagate throughout the entire system sequentially. A series of intermediate events may occur before the inception of catastrophic consequences. Thus, the performance of a diagnosis scheme should be evaluated not only in terms of its correctness but also its timeliness. To enhance the diagnostic efficiency, it becomes necessary to consider the precedence order (in time)

between the fault propagation effects implied in every input–output connection in the digraph.

A series of studies have thus been carried out to develop fault identification techniques by incorporating both the eventual symptoms and also their *occurrence order* into a fuzzy inference system (FIS).<sup>1,20,21</sup> This approach can be implemented in two stages, i.e., (1) the off-line preparation stage and (2) the on-line implementation stage. In the former case, a SDG system model is first constructed to describe the effects of a given set of possible fault origins. The symptom occurrence order (SOO) of any scenario can then be determined accordingly. The corresponding candidate symptom patterns are then translated into a set of IF–THEN fuzzy inference rules for assessing the occurrence possibilities of fault origins. In the next stage, the on-line measurement data were normalized and then used as inputs to the FIS for computing all corresponding occurrence indices. This fault diagnosis strategy has been applied successfully to a number of loop-free processes<sup>20</sup> and also to systems with feedback and/or feed forward control loops.<sup>1,21</sup>

Although significant advancement has been realized by incorporating the concept of symptom occurrence order (SOO) in the SDG-based fault diagnosis procedure, it is still necessary to introduce additional enhancements for practical applications. First of all, the goal of fault diagnosis in previous works is concerned only with the *location* of fault origin. However, varying fault magnitude may cause a change in the propagation pattern in certain cases and, thus, the corresponding diagnostic mechanism should be added in the FIS for their differentiation. Second, only the candidate patterns in single-fault scenarios were encoded with the IF–THEN inference rules in the previous studies. A more sophisticated diagnosis strategy is desirable if the probability of simultaneous faults is not negligible. Finally, other than the control loops considered previously, there must also be *process* loops in realistic systems. A modified version of the pattern generation procedure must also be developed to handle such loops.

The rest of this paper is organized as follows. In section 2, the modified representations of SOOs are introduced for characterizing the evolution process of the on-line symptoms caused by multiple faults propagating in coupled process and control loops. A novel algorithm is then presented in section 3 (and also in Appendices A and B) for automatically generating the fuzzy inference rules associated with any given SOO. Finally, three examples are provided in section 4 to

\* To whom correspondence should be addressed. Tel.: 886-6-275-7575 ext. 62663. Fax: 886-6-234-4496. E-mail address: ctchang@mail.ncku.edu.tw.

demonstrate the feasibility of the proposed approach for isolating more than one fault origin and differentiating two or more fault sizes.

## 2. Predicting Fault Propagation Behaviors

**2.1. Qualitative Simulation Procedure.** By definition, an accident is an unplanned rarely occurring event or a sequence of such events. Some of the catastrophic accidents may not be experienced even in a long-existing plant. Thus, in any realistic system, it is obviously not feasible to collect and analyze the historical data of all possible scenarios. As a result, there is a need to *predict* the fault propagation behaviors with qualitative simulation techniques.

Qualitative reasoning about physical mechanisms has already been widely researched in the past. A simulation algorithm QSIM was developed by Kuipers<sup>22</sup> to anticipate the qualitative transient and steady-state behaviors in practical processes. More specifically, three different problems can be solved with QSIM on the basis of qualitative system models, i.e.

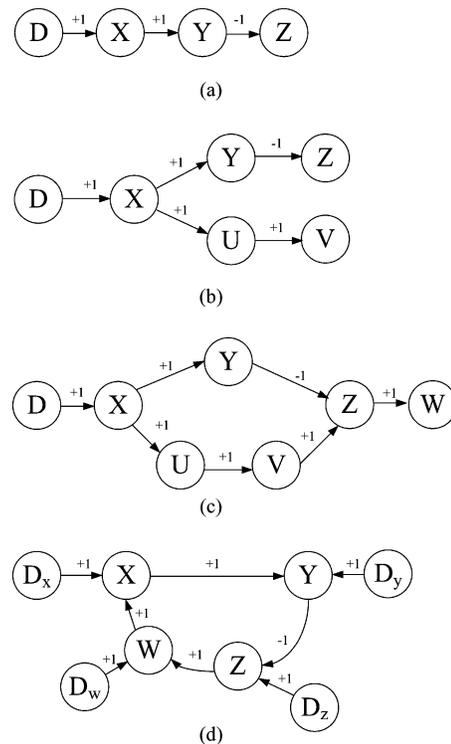
- (1) identification of the possible equilibrium system states according to a set of given conditions,
- (2) determination of the possible dynamic system behaviors evolved from the given initial state, and
- (3) prediction of the possible eventual steady states caused by introducing a perturbation into an originally stable system.

These system models were built with the so-called qualitative differential equations (QDEs) and other qualitative constraint equations. Due to the unprecise nature of the qualitative model, the solutions to each of the above three problems are in general not unique. Furthermore, the computation load needed for simulating all fault propagation behaviors in an industrial process is often prohibitive. A simpler alternative is thus adopted in the present study, i.e., the signed directed graph (SDG). The monotonous input–output relations embedded in an SDG can be either obtained from engineering/operation knowledge or extracted from available QDEs. In the latter case, the QDEs can often be formulated in terms of the traditional differential and algebraic equations with unknown parameters.<sup>16–18,23</sup>

The effects of a fault/failure can be easily simulated with a SDG model in which a set of nodes are connected by directed arcs. It should be first noted that the fault origins are usually associated with the primal nodes, i.e., nodes with no inputs. A set of five values, i.e.,  $\{-10, -1, 0, +1, +10\}$ , may be assigned to each node to *qualitatively* represent deviation from the normal value of the corresponding variable. “0” means that it is at the normal steady state. The negative values are used to denote the lower-than-normal states, and the positive values signify the opposite. The absolute values of nonzero deviations, i.e., 1 or 10, can be interpreted qualitatively as “small” and “large”, respectively. Notice also that the causal relation between two variables under normal conditions can be characterized with a directed arc and the corresponding gain. Again, each gain may assume one of the five qualitative values, i.e.,  $0, \pm 1$ , and  $\pm 10$ . The output value of any arc can be computed with the gain and its input value according to the following equation:

$$v_{\text{out}} = \begin{cases} g \times v_{\text{in}} & \text{if } -10 \leq g \times v_{\text{in}} \leq +10 \\ +10 & \text{if } g \times v_{\text{in}} > +10 \\ -10 & \text{if } g \times v_{\text{in}} < -10 \end{cases} \quad (1)$$

where  $g$ ,  $v_{\text{in}}$ , and  $v_{\text{out}}$  denote respectively the gain, input, and output values. It is obvious that the deviation values of all



**Figure 1.** (a) Fictitious single-path SDG model. (b) Fictitious tree-shaped SDG model. (c) Fictitious feed forward loop. (d) Fictitious feedback loop.

variables affected by one or more fault origin can always be computed with this approach, but the time at which each deviation occurs is indeterminable. Without the reference of time in the SDG-based simulation results, it can nonetheless be safely assumed that the change in an input variable should always occur *earlier* than those in its outputs

**2.2. Fault Propagation Path.** Due to the unique information structure generated with the above approach, a special representation is designed in this study to characterize the predicted fault propagation behaviors. This representation is referred to as the *fault propagation path* (FPP). The FPPs associated with various typical digraph configurations (see Figure 1) have been derived in this work.

Let us first consider the simple single-path SDG given in Figure 1a. The fault propagation path associated with a small disturbance  $D(+1)$  can be written as

$$D(+1) \prec X(+1) \prec Y(+1) \prec Z(-1) \quad (2)$$

Notice that the structure of this path is identical to the corresponding SDG in Figure 1a. Each node here represents a previously nonexistent fault effect. Every effect is specified with a qualitative value  $+1$  or  $-1$ , which can be computed according to eq 1. The precedence order of two consecutive effects is specified with the connecting symbol between them, i.e., the effect on its left should occur earlier than that on the right. The sequence of conditions on this propagation path should be interpreted as the order of occurrence (in time) of different effects resulting from the given fault origin.

The above procedure can also be followed to generate the FPP caused by introducing small disturbances into a system that can be modeled with the tree-shaped digraph given in Figure 1b. In this case, the structure of this FPP is again the same as that of the corresponding SDG. Notice that, although the precedence order of any two effects on the same branch of this FPP can be clearly identified, the order of two distinct events

$$\begin{array}{cccc}
 X(+10) & \prec & Y(+10) & \prec & Z(-10) \\
 & \vee & & \vee & \\
 D(+10) & \prec & X(+1) & \prec & Y(+1) & \prec & Z(-1)
 \end{array}$$

Figure 2. Composite FPP of a single-path SDG.

$$\begin{array}{cccc}
 & & Y(+1) & \prec & Z^{(1)}(-1) & \prec & W^{(1)}(-1) \\
 & \swarrow & & & & & \\
 D(+1) & \prec & X(+1) & & & & \\
 & \searrow & & & & & \\
 & & U(+1) & \prec & V(+1) & \prec & Z^{(2)}(+1) & \prec & W^{(2)}(+1)
 \end{array}$$

Figure 3. Corresponding FPP resulting from  $D(+1)$ .

located on two separate branches should be considered as indeterminable.

The FPP resulting from a “large” disturbance can be obtained by following the identical procedure if fault propagation is immediate. In the case of the single-path SDG in Figure 1a, the corresponding FPP can be written as

$$D(+10) \prec X(+10) \prec Y(+10) \prec Z(-10) \quad (3)$$

However, if a finite time constant is needed to characterize the transient response of an output variable to the disturbance in its input and, also, its direction remains unchanged most of the time during the time window of interest, then an additional constraint must be introduced to facilitate a more accurate description, i.e., *the smaller deviation of a process variable must occur before reaching a larger one of the same variable*. Thus, eq 3 should be revised to incorporate this requirement (see Figure 2). Notice that, in a sense, the fault propagation behaviors described here can be characterized with two distinct types of propagation orders, i.e., spatial and temporal. The former can be regarded as the precedence order of the changes in any two adjacent variables in SDG, while the latter refers to the variation of the same variable with time. To simplify the notation, the composite FPP in Figure 2 will later be written in this paper as

$$D(+10) \prec\prec X(+10) \prec\prec Y(+10) \prec\prec Z(-10) \quad (4)$$

Notice that, if a large disturbance is introduced to the tree-shaped system described in Figure 1b, the resulting FPP can be expressed with the same approach.

A feed forward loop (FFL) is a collection of distinct paths in SDG with common starting and ending nodes. The FFLs can be found in numerous chemical processes, e.g., the feed forward control systems, the ratio control systems, and various processing systems with parallel units, etc. To fix ideas, let us consider the fictitious SDG in Figure 1c as an example. The feed forward loop in this case contains two paths, i.e., (1)  $X \rightarrow Y \rightarrow Z$ , and (2)  $X \rightarrow U \rightarrow V \rightarrow Z$ . Notice that the products of the edge gains along these two paths can be found to be  $-1$  and  $+1$ , respectively. Consequently, this FFL is also referred to as a *negative* feed forward loop (NFFL). It is assumed that the effects of fault origin  $D(+1)$  propagate along separate paths independently. Since the SDG is essentially a static model, it is not possible to tell which effect reaches the ending node  $Z$  first. Thus, the FPP corresponding to a small disturbance in  $D$  should also take the form of a tree, i.e., Figure 3. The symbols  $Z^{(1)}(-1)$  and  $Z^{(2)}(+1)$  are used here to denote the changes in variable  $Z$  caused by disturbances propagating along paths 1 and 2, respectively. Since they represent two separate effects on the

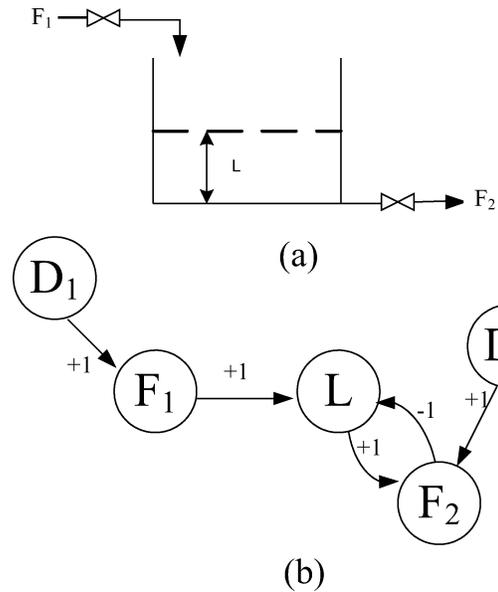


Figure 4. (a) Storage tank. (b) Corresponding digraph model.

Table 1. Steady-State Values of Loop Variables in the Standard Control NFBL Shown in Figure 1d

fault origin	X (controlled variable)	Y (sensor output)	Z (controller output)	W (manipulated variable)
$D_x(+1)$	0	0	-1	-1
$D_y(+1)$	-1	0	-1	-1
$D_z(+1)$	0	0	0	0
$D_w(+1)$	0	0	-1	0
$D_x(+10)$	+1	+1	-10	-10
$D_y(+10)$	-10	+1	-10	-10
$D_z(+10)$	+1	+1	+1	+1
$D_w(+10)$	+1	+1	-10	+1

same variable, a computation procedure is needed to evaluate their net effects at various instances. This reconciliation procedure will be presented later in section 3. Finally, notice that the FPPs in systems with FFLs can always be expressed according to the proposed tree-shaped structure. This approach is applicable even in the case of large upstream disturbance.

A feedback loop (FBL) is a path in the digraph on which the starting and ending nodes coincide. If the product of all edge gains on the loop is negative, it is referred to as a *negative* feedback loop (NFBL). It is in general very difficult to fully simulate the dynamic behavior of a NFBL on the basis of an SDG model alone. To illustrate this point, let us consider the effects of disturbance  $D_x(+1)$  on the example system in Figure 1d. It is obvious that the incipient responses can be determined according to eq 1, i.e.,

$$D_x(+1) \prec X(+1) \prec Y(+1) \prec Z(-1) \prec W(-1) \prec \dots$$

However, since the net effect of two simultaneous inputs, i.e.,  $D(+1)$  and  $W(-1)$ , on  $X$  is uncertain afterward, the event sequence following this initial FPP is really indeterminable without further quantitative and/or qualitative knowledge of the physical system in question.

It should also be noted that the final steady-state values of the loop variables can sometimes be determined *a priori* on the basis of additional process knowledge. If the NFBL in Figure 1d is a *control* loop, these final values can be assigned by following the approach proposed by Ju et al.<sup>24</sup> Table 1 is a complete listing of the final states of control NFBLs in various

scenarios. On the other hand, the final steady states of *process* NFBLs can only be identified on a case-by-case basis. As an example, let us consider the simple storage tank presented in Figure 4a and the corresponding digraph model in Figure 4b. Given the additional information that the outlet flow is gravity driven and, at steady state, the inlet and outlet flow rates must be the same, one can easily conclude that disturbances  $D_1(+1)$  and  $D_2(-1)$  should result in the final states  $[L(+1), F_2(+1)]$  and  $[L(+1), F_2(0)]$ , respectively. In addition, if the mathematical model of the given system is available, Oyeleye and Kramer<sup>25</sup> and later Venkatasubramanian and his co-workers, e.g., see the work of Maurya et al.<sup>10</sup> and Rengaseamy et al.,<sup>19</sup> developed systematic procedures for determining the qualitative steady states under the condition that the model parameters are unknown or uncertain. Although these strategies have been shown to be effective in many applications, a definite identification of the final FBL states still cannot always be guaranteed.

On the basis of the above discussions, one can express the fault propagation behavior caused by the disturbance  $D_X(+1)$  to the standard NFBL in Figure 1d as

$$D_X(+1) \prec X(+1) \prec Y(+1) \prec Z(-1) \prec W(-1) \prec \dots \prec [X(x), Y(y), Z(z), W(w)] \quad (5)$$

where,  $x, y, z,$  and  $w$  represent, respectively, the final steady-state values of  $X, Y, Z,$  and  $W$ , which may or may not be identifiable. Notice also that the final states of all loop variables are lumped into a single node in a square bracket in this FPP and their precedence order is left unspecified. This is due to the difficulties in verifying the occurrence order of these symptoms in real time.

**2.3. Symptom Occurrence Order.** As explained previously, the predicted fault propagation mechanisms are expressed exclusively in this work with FPPs. It should be noted that not all the events included in such representations can be monitored in the practical applications with existing on-line instruments. Thus, to facilitate realistic fault diagnosis, the FPPs should be reduced by merging every pair of a measured variable and its measurement signal in the propagation paths and then eliminating the unmeasured ones. This symptom occurrence order (SOO) is then used as the basis for developing a fault diagnosis system which takes both the eventual symptoms and also their occurrence order into consideration. Since specific examples of this procedure have already been presented elsewhere,<sup>1,20,21</sup> they are not repeated here for the sake of brevity.

### 3. Constructing Fuzzy Inference System

**3.1. Candidate Patterns.** If all symptoms in a SOO can be observed simultaneously, then it is certainly logical to confirm the existence of corresponding fault origin(s). However, since the fault propagation behaviors are dynamical in nature, the resulting on-line measurements should vary with time during the incipient stage. In this work, the collection of on-line symptoms observed at any instance in the fault propagation process is referred to as a *candidate pattern*. It is obvious that every candidate pattern can be considered as an evidence for fault identification. Thus, it is important to enumerate all possibilities and assign each one of them an appropriate confidence level.

In a previous study, Chang et al.<sup>20</sup> developed a systematic procedure to compute the total number of can-

**Table 2. Candidate Patterns Derived from the FPP in Figure 2**

no.	X	Y	Z
1	0	0	0
2	1	0	0
3	1	1	0
4	1	1	-1
5	10	0	0
6	10	1	0
7	10	1	-1
8	10	10	0
9	10	10	-1
10	10	10	-10

didate patterns associated with a tree-shaped simple SOO using only one qualitative level (i.e., small) to characterize the disturbances. Chen and Chang<sup>1</sup> later derived a formula to determine the pattern number for a single-path composite SOO using two qualitative levels (i.e., small and large). This formula has been generalized in the present work to handle  $m$  disturbance levels and  $n$  measurement nodes, i.e.,

$$N_{CP} = \binom{m+n}{m} = \frac{(m+n)!}{m!n!} \quad (6)$$

where  $N_{CP}$  represents the total pattern number. This formula is derived in the proofs of theorems 1 and 2 in Appendix A. To further illustrate its implementation procedure, let us consider the FPP in Figure 2 under the condition that  $X, Y,$  and  $Z$  can be observed on-line. In other words, there are three measurement nodes and two disturbance levels in the corresponding single-path composite SOO. The pattern number in this case should be

$$\frac{(3+2)!}{3!2!} = 10$$

The corresponding patterns can be found in Table 2. On the other hand, the following formulas have also been derived in this work to compute the pattern number associated with a tree-shaped composite SOO:

$$N_{CP} = \mathcal{N}\{\mathbf{P}^{(0)}(m, n_0)\} = \binom{m+n_0-1}{m} + \sum_{j_1=1}^m \binom{m-j_1+n_0-1}{m-j_1} \prod_{i_1=1}^{B_0} \mathcal{N}\{\mathbf{P}^{(0,i_1)}(j_1, n_{0,i_1})\} \quad (7)$$

where  $\mathbf{P}^{(0)}(m, n_0)$  denotes the initial path of a tree-shaped composite SOO having  $m$  disturbance levels and  $n_0$  measurement nodes;  $\mathbf{P}^{(0,i_1)}(j_1, n_{0,i_1})$  denotes the  $i_1$ th branch path connecting to the end of  $\mathbf{P}^{(0)}(m, n_0)$  with  $j_1$  disturbance levels and  $n_{0,i_1}$  measurement nodes;  $B_0$  denotes the total number of these branch paths. Finally, notice that  $\mathcal{N}\{\bullet\}$  is a counting operator and the counting operation is carried out recursively, i.e.,

$$\mathcal{N}\{\mathbf{P}^{(0,i_1,i_2,\dots,i_k)}(j_k, n_{0,i_1,i_2,\dots,i_k})\} = \binom{j_k+n_{0,i_1,i_2,\dots,i_k}-1}{j_k} + \sum_{j_{k+1}=1}^{j_k} \binom{j_k-j_{k+1}+n_{0,i_1,i_2,\dots,i_k}-1}{j_k-j_{k+1}} \times \prod_{i_{k+1}=1}^{B_{0,i_1,i_2,\dots,i_k}} \mathcal{N}\{\mathbf{P}^{(0,i_1,i_2,\dots,i_{k+1})}(j_{k+1}, n_{0,i_1,i_2,\dots,i_{k+1}})\} \quad (8)$$

where  $k = 1, 2, \dots$  and  $\mathbf{P}^{(0,i_1,i_2,\dots,i_{k+1})}(j_{k+1}, n_{0,i_1,i_2,\dots,i_{k+1}})$  is the

**Table 3. Candidate Patterns Derived from the Composite FPP Related to the Treed-Shaped SDG in Figure 1b**

no.	X	Y	Z	U	V	no.	X	Y	Z	U	V
1	0	0	0	0	0	24	10	1	0	1	1
2	1	0	0	0	0	25	10	1	-1	1	1
3	1	1	0	0	0	26	10	10	0	1	1
4	1	1	-1	0	0	27	10	10	-1	1	1
5	10	0	0	0	0	28	10	10	-10	1	1
6	10	1	0	0	0	29	10	0	0	10	0
7	10	1	-1	0	0	30	10	1	0	10	0
8	10	10	0	0	0	31	10	1	-1	10	0
9	10	10	-1	0	0	32	10	10	0	10	0
10	10	10	-10	0	0	33	10	10	-1	10	0
11	1	0	0	1	0	34	10	10	-10	10	0
12	1	1	0	1	0	35	10	0	0	10	1
13	1	1	-1	1	0	36	10	1	0	10	1
14	10	0	0	1	0	37	10	1	-1	10	1
15	10	1	0	1	0	38	10	10	0	10	1
16	10	1	-1	1	0	39	10	10	-1	10	1
17	10	10	0	1	0	40	10	10	-10	10	1
18	10	10	-1	1	0	41	10	0	0	10	10
19	10	10	-10	1	0	42	10	1	0	10	10
20	1	0	0	1	1	43	10	1	-1	10	10
21	1	1	0	1	1	44	10	10	0	10	10
22	1	1	-1	1	1	45	10	10	-1	10	10
23	10	0	0	1	1	46	10	10	-10	10	10

$i_{k+1}$ th branch path (with  $j_{k+1}$  disturbance levels and  $n_{0,i_1,i_2,\dots,i_k,i_{k+1}}$  measurement nodes) connecting to the end of path  $\mathbf{P}^{(0,i_1,i_2,\dots,i_k)}$  ( $j_k$  disturbance levels and  $n_{0,i_1,i_2,\dots,i_k}$  measurement nodes). If there are no further branches connected to the end of the branch path  $\mathbf{P}^{(0,i_1,i_2,\dots,i_k)}$  ( $j_k$  disturbance levels and  $n_{0,i_1,i_2,\dots,i_k}$  measurement nodes), i.e.,  $B_{0,i_1,i_2,\dots,i_k} = 0$ , then

$$\prod_{i_{k+1}=1}^0 [\bullet] = 1 \quad (9)$$

The derivation of the above formulas is presented in the proof of theorem 3 in Appendix A. As an example, let us consider the tree-shaped SDG presented in Figure 1b. The composite FPP resulting from a large disturbance  $D(+10)$  can be obtained with the aforementioned approach. Let us assume that all nodes except the disturbance itself are observable in this FPP, and thus,  $m = 2$ ,  $n_0 = 1$ ,  $n_{0,1} = n_{0,2} = 2$ ,  $B_0 = 2$ , and  $B_{0,1} = B_{0,2} = 0$ . The total number of candidate patterns can be computed according to eqs 7–9, i.e.

$$N_{CP} = \binom{2}{2} + \sum_{j_1=1}^2 \binom{2-j_1}{2-j_1} \prod_{i_1=1}^2 \mathcal{N}\{\mathbf{P}^{(0,i_1)}(j_1, n_{0,i_1})\} =$$

$$1 + \mathcal{N}\{\mathbf{P}^{(0,1)}(1,2)\} \mathcal{N}\{\mathbf{P}^{(0,2)}(1,2)\} + \mathcal{N}\{\mathbf{P}^{(0,1)}(2,2)\} \times$$

$$\mathcal{N}\{\mathbf{P}^{(0,2)}(2,2)\} = 1 + (2 + 1 \times 1)(2 + 1 \times 1) +$$

$$(3 + 2 \times 1 + 1 \times 1)(3 + 2 \times 1 + 1 \times 1) = 46$$

All corresponding patterns are listed in Table 3.

Finally, it should be noted that, although the above formulas can be used to compute the total number of candidate patterns, the actual patterns themselves must still be created on the basis of the precedence order defined in the SOO. In order to alleviate the work load incurred in the manual enumeration process, a computer algorithm has been developed in this work to automatically generate all possible patterns associated with any given SOO. For the sake of completeness, this algorithm is provided in Appendix B.

**3.2. Reconciliation Procedure for Determining the Net Effects on On-line Symptoms.** As mentioned previously, the fault propagation behaviors derived from essentially any

**Table 4. Reconciliation Rules**

rule no.	individual effects	net effect
1	+10, +10	+10
2	+10, +1	+10
3	+10, 0	+10
4	+10, -1	+10/+1
5	+10, -10	+10/+1/0/-1/-10
6	+1, +1	+10/+1
7	+1, 0	+1
8	+1, -1	+1/0/-1
9	+1, -10	-1/-10
10	0, 0	0
11	0, -1	-1
12	0, -10	-10
13	-1, -1	-1/-10
14	-1, -10	-10
15	-10, -10	-10

SDG model can be characterized with the tree-shaped FPPs. The corresponding SOOs are therefore trees also. Since several distinct disturbance levels of the same variable may be expressed with separate nodes in a composite SOO, a reconciliation procedure is needed to determine their net effects at various instances. Three different scenarios are discussed in the sequel:

**3.2.1. Candidate Patterns Derived from a Tree-Shaped SDG.** Notice that the precedence order of the events associated with different values of the same variable is uniquely specified in the corresponding SOO in this case. Thus, according to the definition of SOO, the latest symptom should always override all the previous ones in a candidate pattern. For example, the candidate patterns listed in Tables 2 and 3 can be obtained by applying the pattern generation algorithm described in Appendix B and the overriding principle given here.

**3.2.2. Candidate Patterns Derived from an SDG with NFFLs.** As indicated previously, the symptoms associated with the ending node of an NFFL are located at different branches of the corresponding tree-shaped SOO. Thus, their precedence order is unspecified since the effects of any disturbance entering the starting node of an NFFL should propagate along the branch paths *independently*. In this study, these conflicting effects on the same variable are reconciled according to Table 4. For illustration convenience, let us consider the fictitious SOO in Figure 3 as an example. A total of 21 patterns can be generated by *directly* implementing the proposed algorithms. These patterns are presented in columns 2–9 of Table 5. Since the deviation values of  $Z^{(1)}$  and  $Z^{(2)}$  in columns 6 and 7 represent two different effects on Z, they can be reconciled according to rules given in Table 4. The resulting values are shown in column 10. Notice that the net outcomes of opposite effects (+1 and -1) on Z are evaluated in patterns 16, 17, 20, and 21. According to row 8 in Table 4, there are three possibilities in each of these four cases. Notice also that the net effects on W are listed in column 11 and, only in pattern 21, three alternative outcomes are produced. Consequently, there may be nine distinct combinations for pattern 21. However, it should be noted that Z is the ending node of an NFFL and W is its output. Since the reconciled patterns must also be consistent with the edge gain between them in the SDG model, the deviation values of Z and W listed in the same row in the parenthesis of this pattern should occur simultaneously. In other words, there can be only three alternatives for pattern 21.

**3.2.3. Candidate Patterns Derived from an SDG with NFBLs.** If the final steady-state values of the loop variables of

**Table 5. Candidate Patterns Derived from the FPP in Figure 3**

no.	X	Y	U	V	Z <sup>(1)</sup>	Z <sup>(2)</sup>	W <sup>(1)</sup>	W <sup>(2)</sup>	Z	W
1	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0
4	1	1	0	0	-1	0	0	0	-1	0
5	1	1	0	0	-1	0	-1	0	-1	-1
6	1	0	1	0	0	0	0	0	0	0
7	1	1	1	0	0	0	0	0	0	0
8	1	1	1	0	-1	0	0	0	-1	0
9	1	1	1	0	-1	0	-1	0	-1	-1
10	1	0	1	1	0	0	0	0	0	0
11	1	1	1	1	0	0	0	0	0	0
12	1	1	1	1	-1	0	0	0	-1	0
13	1	1	1	1	-1	0	-1	0	-1	-1
14	1	0	1	1	0	1	0	0	1	0
15	1	1	1	1	0	1	0	0	1	0
16	1	1	1	1	-1	1	0	0	$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$	0
17	1	1	1	1	-1	1	-1	0	$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$	-1
18	1	0	1	1	0	1	0	1	1	1
19	1	1	1	1	0	1	0	1	1	1
20	1	1	1	1	-1	1	0	1	$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$	1
21	1	1	1	1	-1	1	-1	1	$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$

**Table 6. Fuzzy Inference Rules Constructed According to the Candidate Patterns in Table 2**

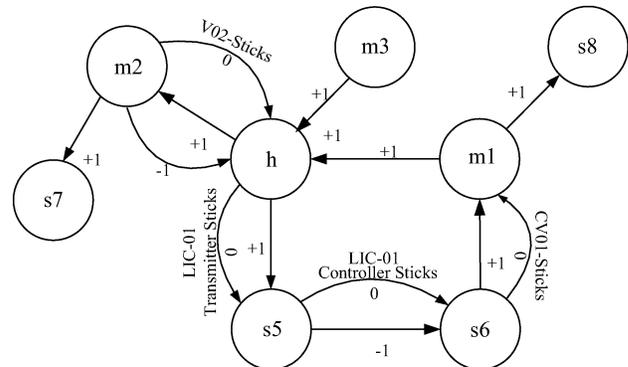
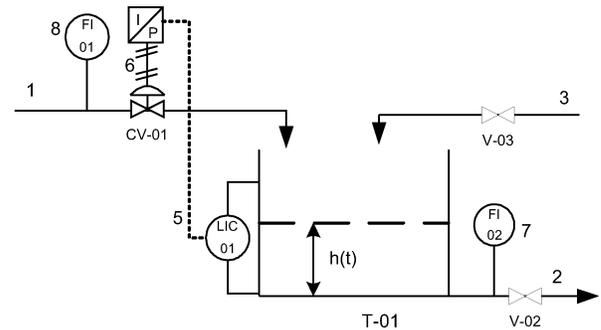
no.	IF			THEN
	X	Y	Z	cs
1	ZE	ZE	ZE	NOC
2	SP	ZE	ZE	UCT <sub>1</sub>
3	SP	SP	ZE	UCT <sub>2</sub>
4	SP	SP	SN	UCT <sub>3</sub>
5	LP	ZE	ZE	UCT <sub>2</sub>
6	LP	SP	ZE	UCT <sub>3</sub>
7	LP	SP	SN	UCT <sub>4</sub>
8	LP	LP	ZE	UCT <sub>4</sub>
9	LP	LP	SN	UCT <sub>5</sub>
10	LP	LP	LN	OCR

a NFBL can be determined on the basis of additional process knowledge, then the corresponding SOO can be constructed on the basis of eq 5. In such a case, the aforementioned overriding principle should also be applicable.

**3.3. Rule Compilation.** Each candidate pattern can be encoded into an IF–THEN rule to evaluate the existence potential (or occurrence index cs) of the corresponding fault origin. Specifically, the premises of this rule are constructed on the basis of the qualitative deviation values of the symptoms in the given pattern. These deviations are translated into linguistic values according to an interpretation function  $F_{in}$ , i.e.,

$$F_{in}(\delta_j) = \begin{cases} \text{LN} & \text{if } \delta_j = -10 \\ \text{SN} & \text{if } \delta_j = -1 \\ \text{ZE} & \text{if } \delta_j = 0 \\ \text{SP} & \text{if } \delta_j = +1 \\ \text{LP} & \text{if } \delta_j = +10 \end{cases} \quad (10)$$

In the above equation,  $\delta_j$  denotes the deviation value of the  $j$ th measurement ( $j = 1, 2, \dots, N_M$ ); LN, SN, ZE, SP, and LP denote respectively the linguistic values of -10, -1, 0, +1, and +10.



**Figure 5.** (a) Single-tank, level-control system. (b) Corresponding digraph model.

**Table 7. Fault Origins and Their Simulation Methods Adopted in Example 1**

case no.	$m_2$	$m_3$	CV-01 sticks	simulation method
1	-10			multiply $C_1$ in eq B.2 by 0% at 1000 s
2	-1			multiply $C_1$ in eq B.2 by 50% at 1000 s
3		+10		let $q_3 = 1000 \text{ cm}^3/\text{s}$ in eq B.1 at 1000 s
4		+1		let $q_3 = 450 \text{ cm}^3/\text{s}$ in eq B.1 at 1000 s
5	-1		Y	(1) multiply $C_1$ in eq B.2 by 25% at 1000 s (2) fix control-valve opening at 1000 s
6		+1	Y	(1) set $q_3 = 200 \text{ cm}^3/\text{s}$ in eq B.1 at 1000 s (2) fix control-valve opening at 1000 s
7	-1	+1		(1) multiply $C_1$ in eq B.2 by 50% at 1000 s (2) set $q_3 = 450 \text{ cm}^3/\text{s}$ in eq B.1 at 1000 s
8	-1	+1	Y	(1) multiply $C_1$ in eq B.2 by 25% at 1000 s (2) set $q_3 = 200 \text{ cm}^3/\text{s}$ in eq B.1 at 1000 s (3) fix control-valve opening at 1000 s

If the on-line symptoms are identical to those in an SOO, then it is highly possible that they are caused by the corresponding fault origin. To assert such a belief, the conclusion “cs is OCR” should be used in the inference rule. Here, OCR is the linguistic value of the occurrence index cs reflecting the highest confidence level in confirming the existence of the root cause(s). On the other hand, it is reasonable to disregard the possibility of a fault if none of the symptoms in the corresponding SOO can be observed. Thus, the conclusion in the inference rule for this scenario should be “cs is NOC”, where NOC is the linguistic value representing the lowest confidence. The conclusions of the remaining rules should be uncertain. Naturally, the confidence level of a particular candidate pattern in confirming the existence of the root cause(s) should be proportional to the number of matched (or occurred) symptoms in the SOO. In this

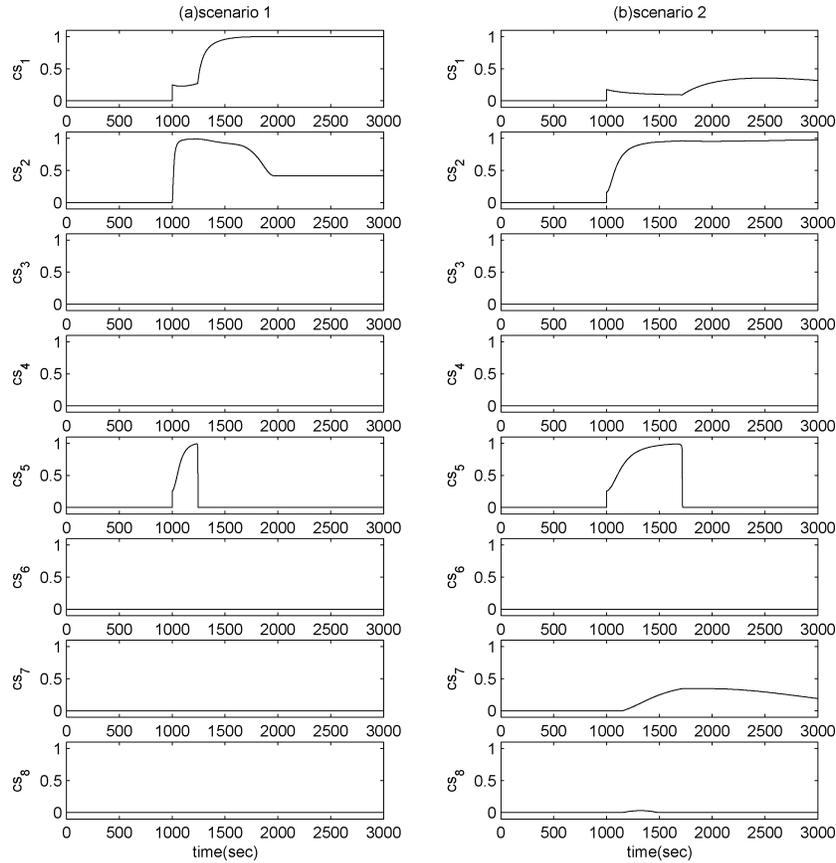


Figure 6. Diagnosis results of example 1. (a) Occurrence indices obtained when the actual fault origin is  $m_2(-10)$ . (b) Occurrence indices when the actual fault origin is  $m_2(-1)$ .

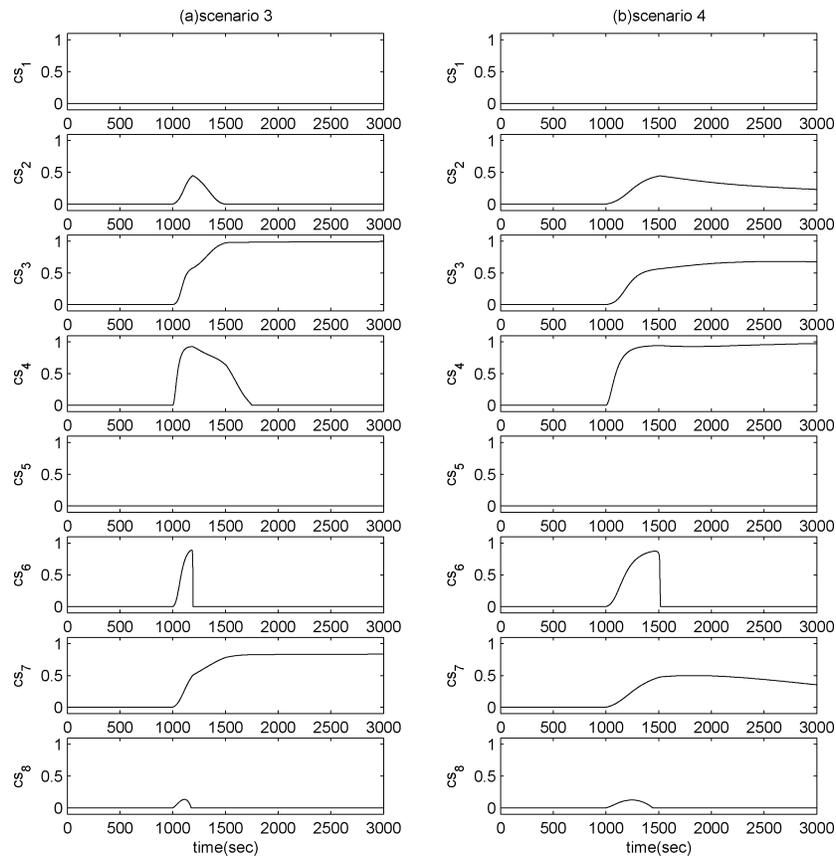
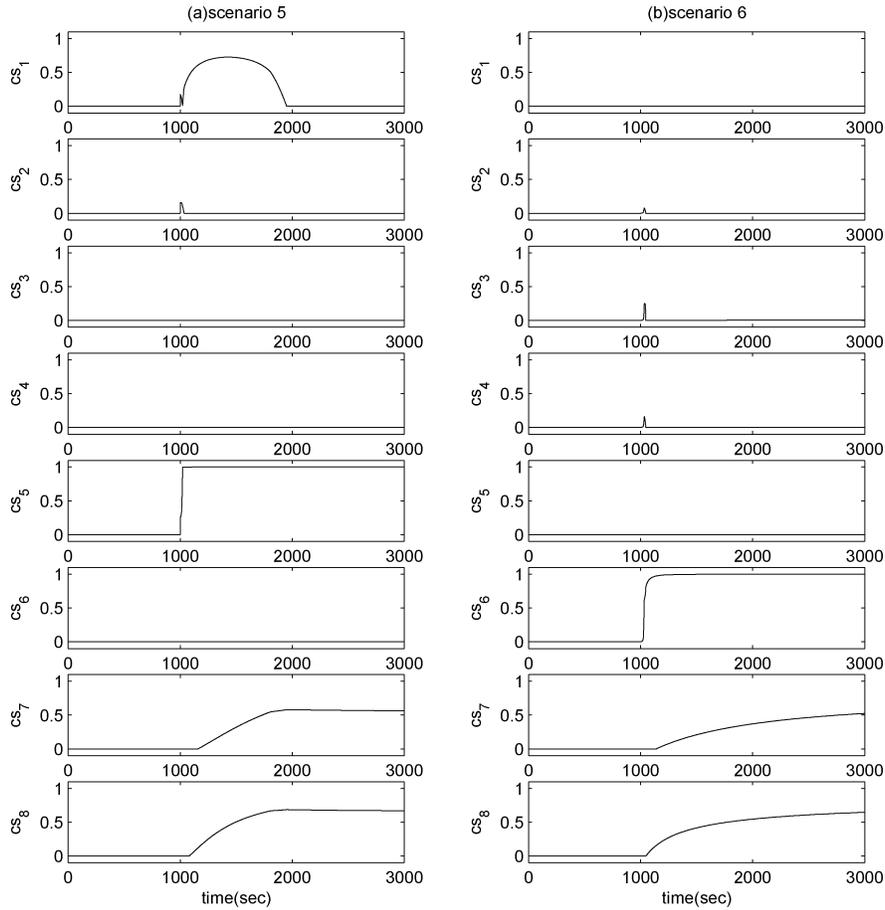
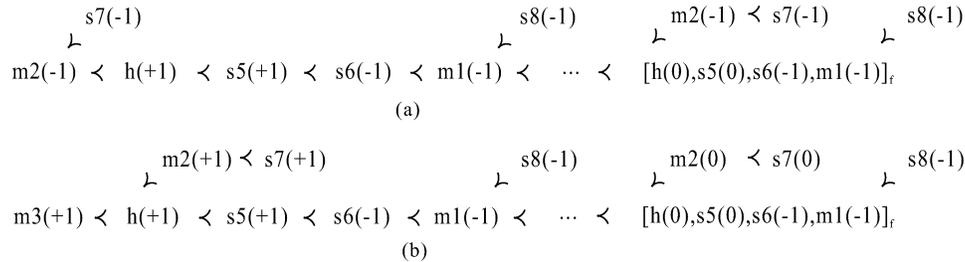


Figure 7. Diagnosis results of example 1. (a) Occurrence indices obtained when the actual fault origin is  $m_3(+10)$ . (b) Occurrence indices obtained when the actual fault origin is  $m_3(+1)$ .



**Figure 8.** Diagnosis results of example 1. (a) Occurrence indices obtained when the two actual fault origins are  $m_2(-1)$  and valve sticking. (b) Occurrence indices obtained when the two actual fault origins are  $m_3(+1)$  and valve sticking.



**Figure 9.** Two FPPs used for scenario 7 in example 1.

study, the latter value  $l$  is used directly as a qualitative measure of confidence. Since the events associated with different values of the same variable may be included as different nodes in an SOO, the latest among them always overrides all the previous ones in the on-line measurements. To account for the overridden symptoms implied in a candidate pattern, the following formula is used in this work for computing the confidence level  $l$ :

$$l = \sum_{j=1}^{N_M} c(\delta_j) \tag{11}$$

where  $c(\delta_j)$  denotes the number of symptoms which occurred in the  $j$ th measured variable when its current value is  $\delta_j$ . In other words,  $l$  is the total number of nodes in an SOO which have been confirmed with measurement data.

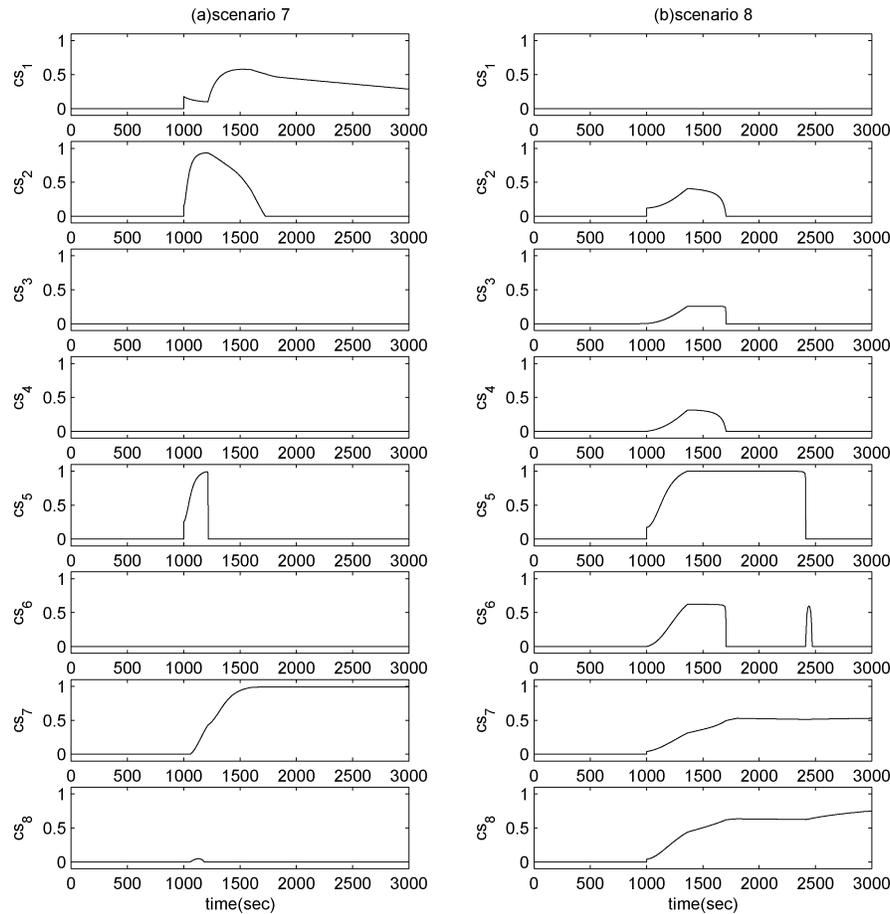
A second interpretation function  $F_{out}$  can be defined accordingly to determine the linguistic values of the occurrence index, i.e.

$$F_{out}(l) = \begin{cases} \text{NOC} & \text{if } l = 0 \\ \text{OCR} & \text{if } l = l_{max} \\ \text{UCT}_l & \text{otherwise} \end{cases} \tag{12}$$

where  $l_{max}$  is the confidence level associated with the fully developed candidate pattern of the given SOO. As an example, let us consider the candidate patterns in Table 2. These patterns can be converted to the fuzzy inference rules presented in Table 6 with the aforementioned interpretation functions.

#### 4. Case Studies

To verify the effectiveness of the proposed fault diagnosis approach, extensive numerical simulation studies have been



**Figure 10.** Diagnosis results of example 1. (a) Occurrence indices obtained when the two actual fault origins are  $m_2(-1)$  and  $m_3(+1)$ . (b) Occurrence indices obtained when the three actual fault origins are  $m_2(-1)$ ,  $m_3(+1)$ , and valve sticking.

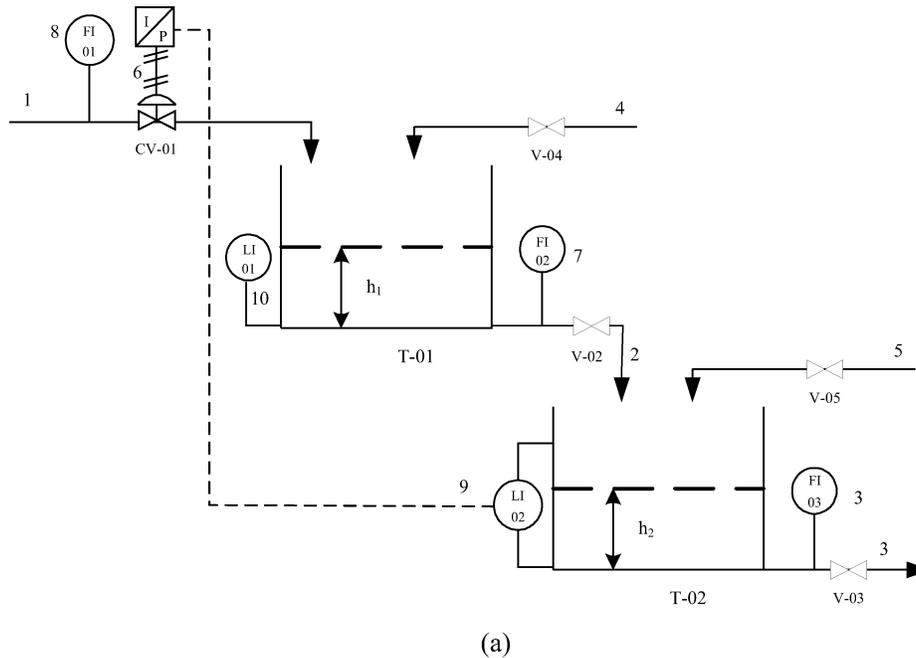
carried out in this work. The on-line measurement data of all fault propagation scenarios were generated with SIMULINK.<sup>26</sup> These data were then used in Sugeno's inference procedure with the fuzzy-logic module in the MATLAB toolbox.<sup>27</sup> Three typical examples are presented below. In all these studies, it has been assumed that a standard hazard assessment method, e.g., fault tree analysis (FTA), can be applied off-line to establish the scope of fault diagnosis. The most significant advantage of this practice is that the candidates of fault identification are restricted to only the causes of one or more given top events and, consequently, the diagnosis procedure can be greatly simplified.

**Example 1.** Let us consider the single-tank level-control system in Figure 5a and the corresponding SDG model in Figure 5b. The model equations and the numerical values of model parameters used in the simulation studies are listed in the appendix (see Table C.1). A total of eight possible scenarios have been identified with fault tree analysis for this example. The fault origins of these scenarios and their simulation procedures are listed in Table 7. Scenarios 1–4 are used to demonstrate the capability of the proposed diagnosis method in distinguishing different magnitudes of the same disturbance. The diagnosis results obtained by introducing the first two fault origins, i.e.,  $m_2(-10)$  and  $m_2(-1)$ , are presented in Figure 6a and b, respectively, while those associated with  $m_3(+10)$  and  $m_3(+1)$  are given in Figure 7a and b. The high  $CS_7$  value in Figure 7a is mainly due to the fact that the transient symptoms in scenario 7 happen to be very similar to those in scenario 3. In scenarios 5 and 6, a small disturbance (i.e.,  $m_2(-1)$  or  $m_3(+1)$ ) is introduced in the presence of a hardware failure (i.e., control valve CV-01 sticks). The corresponding results are

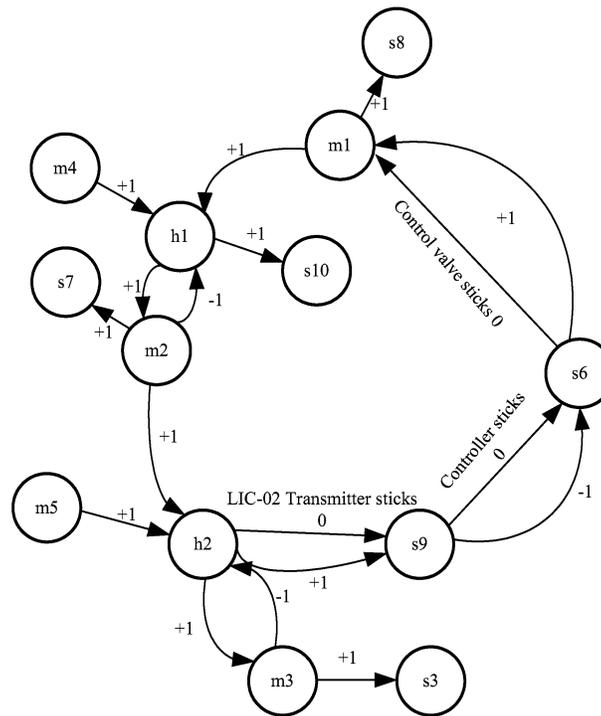
**Table 8. Fault Origins and Their Simulation Methods Adopted in Example 2**

case no.	fault origin(s)	simulation method
1	$m_4(+10)$	set $q_4 = 1000 \text{ cm}^3/\text{s}$ in eq B.4 at 1000 s
2	$m_4(+1)$	set $q_4 = 700 \text{ cm}^3/\text{s}$ in eq B.4 at 1000 s
3	$m_5(+10)$	set $q_5 = 1000 \text{ cm}^3/\text{s}$ in eq B.5 at 1000 s
4	$m_5(+1)$	set $q_5 = 700 \text{ cm}^3/\text{s}$ in eq B.5 at 1000 s
5	$m_4(+1), m_5(+1)$	(1) set $q_4 = 700 \text{ cm}^3/\text{s}$ in eq B.4 at 1000 s (2) set $q_5 = 700 \text{ cm}^3/\text{s}$ in eq B.5 at 1000 s

shown in Figure 8a and b. Next let us consider scenario 7 in which two external disturbances are present, i.e., a decrease in the output flow  $m_2$  and an increase in the input flow  $m_3$ . A FPP can be constructed for each disturbance *individually* according to the previously proposed procedures (see Figure 9). These two FPPs can be combined to describe the fault propagation mechanisms in the present scenario. Specifically, it can be postulated that the root nodes  $m_2(-1)$  and  $m_3(+1)$  of the FPPs in Figure 9a and b are the effects of a fictitious common cause. The corresponding SOO and candidate patterns can be obtained on the basis of this combined FPP. Notice that the final steady-state values of the loop variables must be evaluated in this case with additional process knowledge. Let us assume that the combined effects of  $m_2(-1)$  and  $m_3(+1)$  are equivalent to those caused by an uncontrollable disturbance. Consequently, the final states of loop variables can be assigned by following the approach suggested in the work of Ju et al.,<sup>24</sup> i.e.,  $h(+1)$ ,  $s_5(+1)$ ,  $s_6(-10)$ , and  $m_1(-10)$ . The corresponding simulation results are shown in Figure 10a. Finally, let us consider a similar scenario in which the two aforementioned disturbances occur after control valve CV-01 sticks. Under the



(a)



(b)

**Figure 11.** (a) Process flow diagram of a two-tank, level-control system. (b) Corresponding SDG model of a two-tank, level-control system.

condition that the control valve failure always exists, two separate FPPs can be constructed for the individual disturbances. The same approach can then be taken to merge these two FPPs and to generate the corresponding SOO and candidate patterns. The simulation results are presented in Figure 10b. It can be clearly observed that the diagnostic resolution is good enough to differentiate the multiorigin and multimagnitude scenarios.

**Example 2.** Next, let us consider the two-tank level-control system presented in Figure 11a and the corresponding SDG model in Figure 11b. The scenarios studied in this example are

listed in Table 8. In every scenario, the disturbance(s) is originated from either  $m_4$  or  $m_5$ . The magnitude of each disturbance is divided into two levels, i.e., 1 (small) and 10 (large). Notice that scenario 5 is concerned with two coexistent fault origins, i.e.,  $m_4(+1)$  and  $m_5(+1)$ . The model equations and parameters used in simulation studies can be found in the appendix (see Table C.2). The SOO obtained in each scenario is shown in Figure 12. The diagnosis results are presented in Figures 13–15.

**Example 3.** Finally, let us consider an exothermic continuously stirred tank reactor (CSTR) together with its temper-



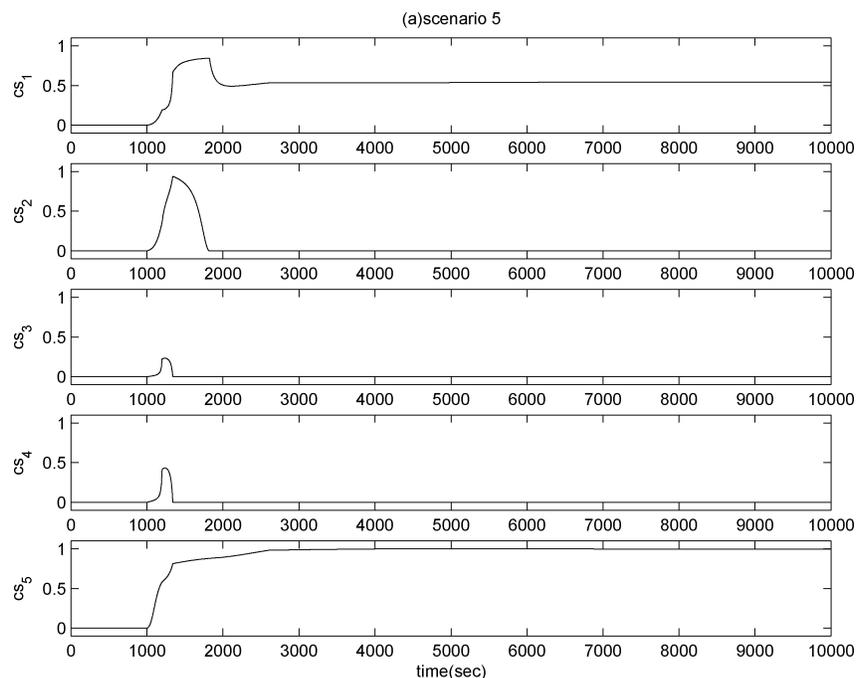


Figure 15. Occurrence indices obtained in example 2 when the actual fault origins are  $m_4(+1)$  and  $m_5(+1)$ .

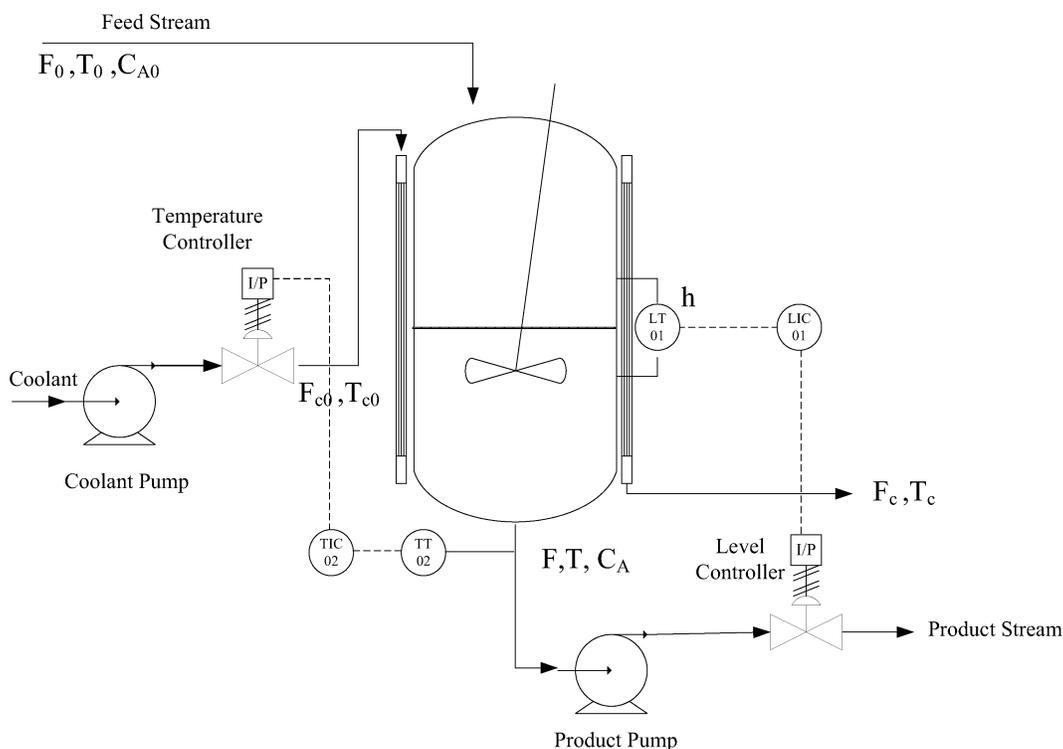


Figure 16. Process flow diagram of a CSTR with temperature- and level-control systems.

ature and level control loops<sup>23</sup> (see Figure 16). It is assumed that there are six measurable process variables, i.e., the height of the liquid level in the reactor ( $h$ ), the flow rate and outlet temperature of cooling water ( $F_c$  and  $T_c$ ), and the temperature, flow rate, and reactant concentration at the outlet of the CSTR ( $T$ ,  $F$ , and  $C_A$ ). For simplicity, it is further assumed that the variations in these variables are always accurately reflected in their measurements and, therefore, it is not necessary to distinguish a measured variable from its measurement signal in the digraph model. The resulting SDG is given in

Figure 17. It can be observed that the digraph configuration is quite complex and the feed forward and feedback loops in this system are highly coupled. The model equations and parameters used in this example are presented in the appendix (see Table C.3), and all simulated scenarios are summarized in Table 9. The initial FPPs corresponding to the fault origins are given in Figure 18, and the final states of process variables in every scenario are presented in Table 10. It should be noted that, due to the presence of complex process loops in this system, the final steady-state values of

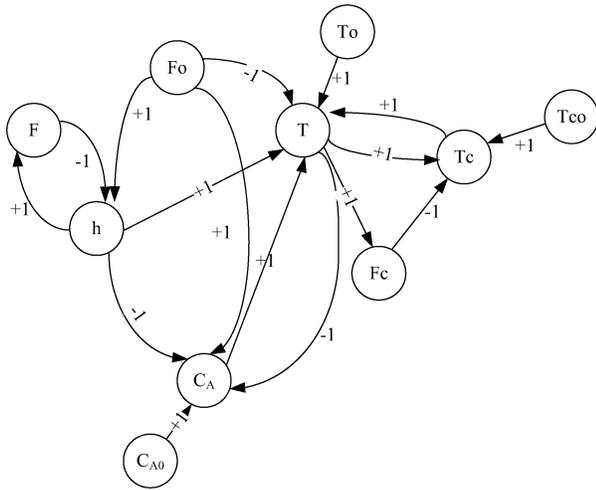


Figure 17. SDG model of CSTR system.

Table 9. Fault Origins and Their Simulation Methods Adopted in Example 3

case no.	fault origin(s)	simulation method
1	$F_0(+10)$	introduce an increase of 1000 ft <sup>3</sup> /h in $F_0$ at 3 h
2	$F_0(+1)$	introduce an increase of 550 ft <sup>3</sup> /h in $F_0$ at 3 h
3	$T_0(+10)$	introduce an increase of 120 °R in $T_0$ at 3 h
4	$T_0(+1)$	introduce an increase of 60 °R in $T_0$ at 3 h
5	$F_0(+1), T_0(+1)$	(1) introduce an increase of 550 ft <sup>3</sup> /h in $F_0$ at 3 h (2) introduce an increase of 60 °R in $T_0$ at 3 h

some of the variables cannot be uniquely determined with the SDG model alone. In these cases, all possible values are listed (see Table 10a). Notice that, by definition, the final value of each controlled variable can be predicted with a high degree of confidence. In particular, this value should be 0 if the external disturbance is controllable (with magnitude 1) and  $\pm 1$  if an uncontrolled disturbance (with magnitude 10) is introduced. Let us use scenario 1 as an example to illustrate this point. Since the fault origin in this case is  $F_0(+10)$  and multiple process NFFLs and NFBLs exist between nodes  $F_0$  and  $C_A$  in SDG, the net effect on the latter is really indeterminate on the basis of the digraph model only. Similarly, the net effects of fault origin on the temperature control loop and its loop variables (i.e.,  $T$ ,  $T_c$ , and  $F_c$ ) are also unpredictable. On the other hand, the final state of the liquid level should be +1 since  $h$  is located in a single control loop and this loop is affected only by an input  $F_0$ . Notice that the final states listed in rows 2–5 of Table 10a can also be determined in the same fashion. On the basis of the initial FPPs and also the incomplete information about the final states, a set of fuzzy inference rules can be obtained with the proposed approach. These rules have been tested with simulated data. The diagnosis results of scenarios 1 and 2 are shown in Figure 19. It is clear that these inference rules can be used to differentiate disturbances with different magnitudes, i.e.,  $F_0(+10)$  and  $F_0(+1)$ . Similarly, the same conclusion about  $T_0(+10)$  and  $T_0(+1)$  can be drawn from Figure 20. The feasibility of diagnosing multiple coexistent fault origins is demonstrated in Figure 21. In this scenario, disturbances  $F(+1)$  and  $T(+1)$  are introduced simultaneously at 3 h. Notice that the value of performance index  $cs_5$  becomes the largest among all five indices almost immediately after the faults occur. It should be noted that the occurrence indices of incorrect fault origins may not reach zero eventually. This phenomenon can be observed in scenarios 2–5. This is obviously due to the fact that some of the final steady-state values are uncertain. If the final

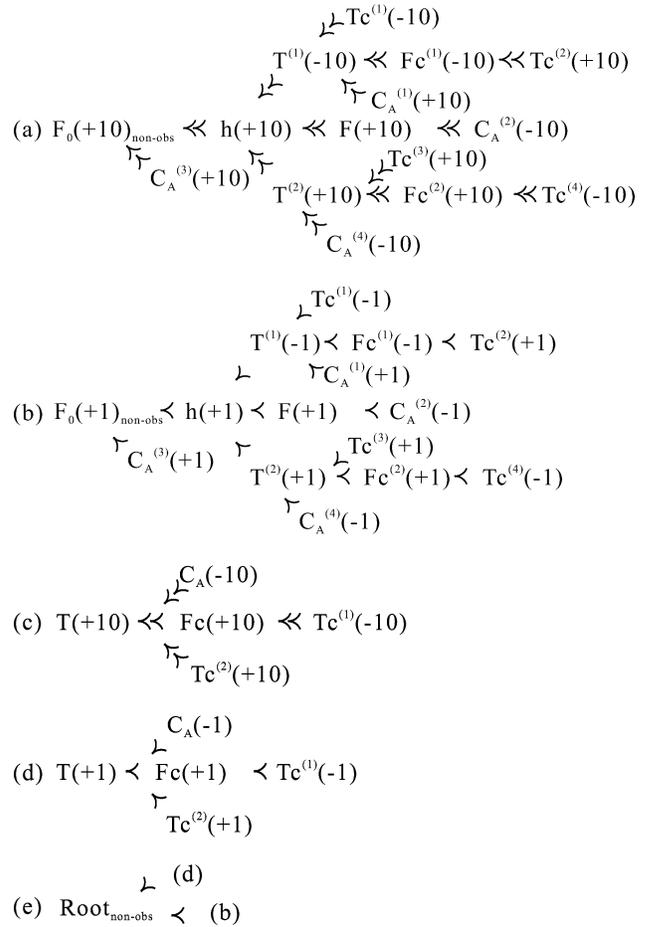


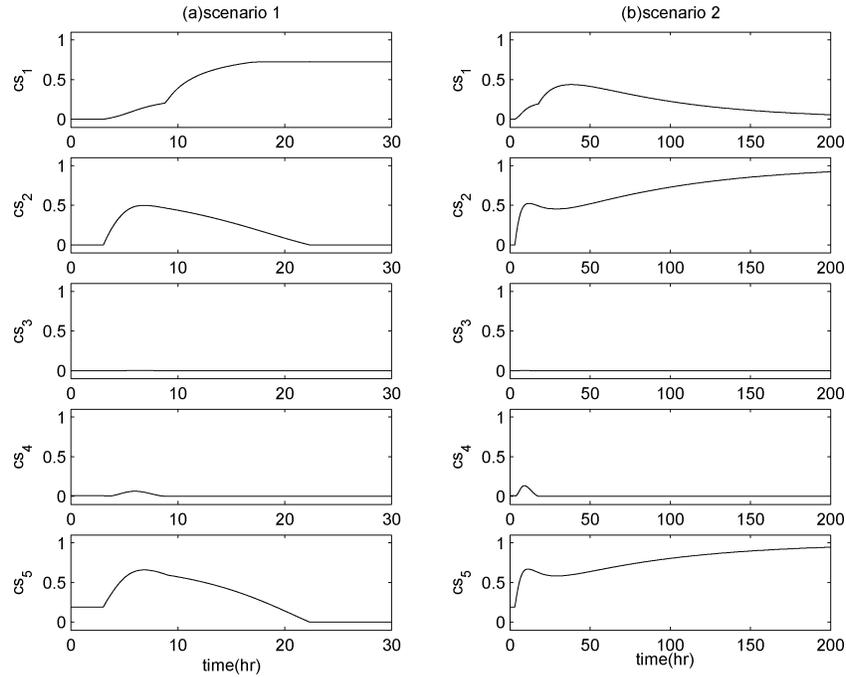
Figure 18. Initial SOOs associated with the fault origins considered in example 3.

Table 10. Partially Predicted Final Steady-State Values Adopted and Complete Final Steady-State Values Assumed in Every Scenario of Example 3

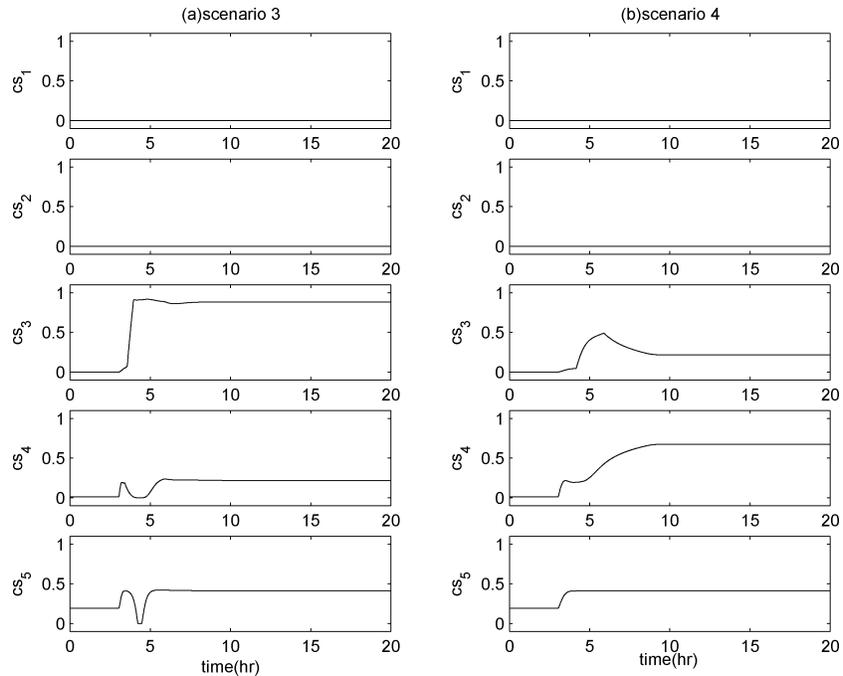
case no.	$h$	$C_A$	$T$	$T_c$	$F$	$F_c$
Partially Predicted Final Steady-State Values Adopted						
1	+1	+10/+1/0/ -1/-10	+1/0/-1	+1/0/-1	+10	+10/+1/0/ -1/-10
2	0	+1/0/-1	0	0	+1	+1/0/-1
3	0	+10/+1/0/ -1/-10	+1	+1	0	+10
4	0	+1/0/-1	0	0	0	+1
5	0	+10/+1/0/ -1/-10	-1/0/+1	-1/0/+1	+1	+10/+1/0/ -1/-10
Complete Final Steady-State Values Assumed						
1	+1	0	0	0	+10	+10
2	0	0	0	0	+1	0
3	0	0	+1	+1	0	+10
4	0	0	0	0	0	+1
5	0	0	0	0	+1	+10

states listed in Table 10b can be assumed to be available, then the diagnostic resolution of the proposed method can be further enhanced. In other words, the occurrence indices of the incorrect origins can be suppressed to very low levels in all scenarios.

Other than the feasibility problem discussed above, let us also address the issue of on-line computation load with additional data gathered in the simulation studies carried out for this example. A total of 5248 rules have been constructed for the five fault origins under consideration. To execute the computation required for fault diagnosis, the fuzzy inference module has been used on a desktop PC with an AMD Athlon 64



**Figure 19.** Diagnosis results of example 3. (a) Occurrence indices obtained when the actual fault origin is  $F_0(+10)$ . (b) Occurrence indices obtained when the actual fault is  $F_0(+1)$ .



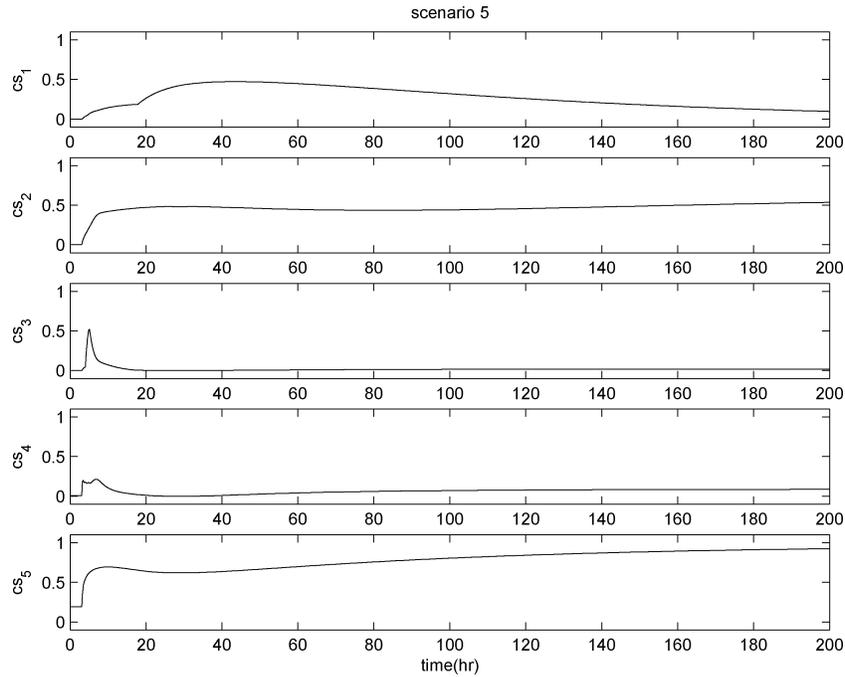
**Figure 20.** Diagnosis results of example 3. (a) Occurrence indices obtained when the actual fault origin is  $T_0(+10)$ . (b) Occurrence indices obtained when the actual fault origin is  $T_0(+1)$ .

processor 3000 + 1.81 GHz and 1.00 GB RAM. It was found that the computation time needed to process each on-line data set is less than 0.0017 s, which should be quite acceptable for practical implementation.

**5. Conclusions**

The fault propagation paths (FPPs) and symptom occurrence orders (SOOs) caused by multiple fault origins with one or more

possible magnitude levels are characterized systematically in this study according to (1) the degree of deviation of each process variable from its normal state and (2) the occurrence order of the abnormal deviations. Fuzzy diagnosis rules can then be generated automatically on the basis of the candidate patterns derived from SOOs. Mathematical theorems and the corresponding pattern generation algorithms are also developed to enumerate all possible on-line symptoms. The simulation results of several example are presented in this paper to demonstrate



**Figure 21.** Occurrence indices obtained in example 3 when the two actual fault origins are  $T_0(+1)$  and  $F_0(+1)$ .

the feasibility of the proposed approach. It can be observed that the fuzzy inference system constructed in this study is indeed suitable for fault diagnosis in multiorigin and/or multimagnitude scenarios.

#### Appendix A: Theorems Related to the Number of Candidate Symptom Patterns

**Theorem 1.** Consider a set of single-path composite SOOs denoted by  $\mathbf{P}(m, n)$ , where  $m$  is the number of disturbance levels and  $n$  is the number of measurement nodes. The total number of candidate patterns of  $\mathbf{P}(m, n)$ , denoted as  $N_{m,n}$ , can be computed with either one of the following two formulas:

$$N_{m,n} = \sum_{i_1=1}^{m+1} \sum_{i_2=1}^{i_1} \cdots \sum_{i_{n-2}=1}^{i_{n-3}} \sum_{i_{n-1}=1}^{i_{n-2}} i_{n-1} \quad m \geq 0 \quad n \geq 2 \quad (\text{A.1})$$

$$N_{m,n} = \sum_{j_1=1}^{n+1} \sum_{j_2=1}^{j_1} \cdots \sum_{j_{m-2}=1}^{j_{m-3}} \sum_{j_{m-1}=1}^{j_{m-2}} j_{m-1} \quad m \geq 2 \quad n \geq 0 \quad (\text{A.2})$$

**Proof 1.** At any time, the first measurement node in path  $\mathbf{P}(m, n)$  may assume one value among  $m + 1$  possible ones, i.e., those corresponding to the normal state and the  $m$  disturbance levels. Given a specific value (say level  $i_1 - 1$ ) of the first node, the second node can only assume values at the same level or the lower levels, i.e., those corresponding to the normal state and the disturbances from level 1 to level  $i_1 - 1$ . Similarly, the number of possible values of the remaining nodes can be determined on the basis of the same rationale. As indicated in eq A.1, the total number of candidate patterns of  $\mathbf{P}(m, n)$  is computed by summing all such possibilities.

On the other hand, there are also  $n + 1$  possible propagation lengths on the lowest disturbance level in  $\mathbf{P}(m, n)$ , i.e., from no nodes to  $n$  nodes. Given a specific length (say of  $j_1 - 1$  nodes) on the lowest level, the propagation length on the second lowest level should not be longer, i.e., from no nodes to  $j_1 - 1$  nodes. Similarly, the number of possible lengths on the remaining disturbance

levels can be determined according to the same approach. As indicated in eq A.2, the total number of candidate patterns of  $\mathbf{P}(m, n)$  is computed by summing all such possibilities.

**Theorem 2.** The total number of candidate patterns associated with a single-path composite SOO with  $m$  disturbance levels and  $n$  measurement nodes is

$$N_{\text{CP}} = N_{m,n} = \binom{m+n}{m} \quad (\text{A.3})$$

**Proof 2.** This theorem is proved by mathematical induction. The basis of induction is established according to the following steps:

- (1) Set  $N_{0,0} = 1$
- (2) The pattern numbers

$$N_{0,1} = N_{1,0} = \frac{1!}{1!0!} = 1$$

$$N_{1,1} = \frac{2!}{1!1!} = 2$$

can be determined by direct enumeration.

- (3) From eq A.1, one can obtain

$$N_{0,2} = \sum_{i_1=1}^1 i_1 = \frac{2!}{0!2!} = 1$$

$$N_{1,2} = \sum_{i_1=1}^2 i_1 = \frac{3!}{2!1!} = 3$$

- (4) From eq A.2, one can obtain

$$N_{2,0} = \sum_{j_1=1}^1 j_1 = \frac{2!}{0!2!} = 1$$

$$N_{2,1} = \sum_{j_1=1}^2 j_1 = \frac{3!}{2!1!} = 3$$

(5) From eq A.1 or A.2, one can obtain

$$N_{2,2} = \sum_{i_1=1}^3 i_1 = \sum_{j_1=1}^3 j_1 = \frac{4!}{2!2!} = 6$$

Thus, it can be hypothesized that the theorem is true for  $0 \leq m \leq m'$  and  $0 \leq n \leq n'$ , i.e.,

$$N_{m,n} = \binom{n+m}{m}$$

where,  $m' = 2, 3, \dots$  and  $n' = 2, 3, \dots$

Notice that eq A.1 can be written as

$$N_{m',n'+1} = N_{0,n'} + N_{1,n'} + \dots + N_{m',n'}$$

Substituting the induction hypothesis into the above equation yields

$$N_{m',n'+1} - N_{m',n'} = \sum_{i=0}^{m'-1} \binom{i+n'}{i}$$

By using the identity

$$\sum_{k=0}^K \binom{L+k}{k} = \binom{L+K+1}{K}$$

(where  $k, L$ , and  $K$  are non-negative integers), the above equation can be converted to

$$\begin{aligned} N_{m',n'+1} &= N_{m',n'} + \binom{m'+n'}{m'-1} = \\ &= N_{m',n'} + \binom{m'+n'+1}{m'} - \binom{m'+n'}{m'} = \\ &= \binom{m'+n'+1}{m'} = \binom{m'+n'+1}{n'+1} \end{aligned}$$

Similarly, eq A.2 can be written as

$$N_{m'+1,n'} = N_{m',0} + N_{m',1} + \dots + N_{m',n'}$$

By following the same derivation procedure, it can be shown that

$$\begin{aligned} N_{m'+1,n'} - N_{m',n'} &= \sum_{j=0}^{n'-1} \binom{m'+j}{j} = \binom{m'+n'}{n'-1} = \\ &= \binom{m'+n'+1}{n'} - \binom{m'+n'}{n'} \end{aligned}$$

$$N_{m'+1,n'} = \binom{m'+n'+1}{n'} = \binom{m'+n'+1}{m'+1}$$

Thus, eq A.3 should be valid for all non-negative integers.

**Theorem 3.** Let us consider a tree-shaped SOO characterized by  $m$  disturbance levels. If  $\mathbf{P}^{(0)}(m, n_0)$  denotes the initial path of this tree with  $n_0$  measurement nodes,  $\mathbf{P}^{(0,i_1)}(m, n_{0,i_1})$  ( $i_1 = 1, 2, \dots, B_0$ ) denotes the  $i_1$ th branch path connecting to the end of  $\mathbf{P}^{(0)}(m, n_0)$  with  $n_{0,i_1}$  measurement nodes,  $\mathbf{P}^{(0,i_1,i_2)}(m, n_{0,i_1,i_2})$  ( $i_2 = 1, 2, \dots, B_{0,i_1}$ ) denotes the  $i_2$ th branch path of length  $n_{0,i_1,i_2}$  connecting to the end of  $\mathbf{P}^{(0,i_1)}(m, n_{0,i_1})$ , etc., then the total number of candidate patterns  $N_{CP}$  can be computed according to the following equation

$$\begin{aligned} N_{CP} = \mathcal{N}\{\mathbf{P}_{n_0}^{(0)(m)}\} &= \binom{n_0 - 1 + m}{m} + \\ &+ \sum_{j_1=1}^m \binom{m - j_1 + n_0 - 1}{m - j_1} \prod_{i_1=1}^{B_0} \mathcal{N}\{\mathbf{P}^{(0,i_1)}(j_1, n_{0,i_1})\} \end{aligned} \quad (\text{A.4})$$

where  $\mathcal{N}\{\bullet\}$  denotes the counting operator of a given path.

The results of counting operation can be obtained recursively, i.e.,

$$\begin{aligned} \mathcal{N}\{\mathbf{P}^{(0,i_1,i_2,\dots,i_k)}(j_k, n_{0,i_1,i_2,\dots,i_k})\} &= \\ &= \binom{n_{0,i_1,i_2,\dots,i_k} - 1 + j_k}{j_k} + \sum_{j_{k+1}=1}^{j_k} \binom{j_k - j_{k+1} + n_{0,i_1,i_2,\dots,i_{k+1}}}{j_k - j_{k+1}} \times \\ &\times \sum_{i_{k+1}=1}^{B_{0,i_1,i_2,\dots,i_k}} \mathcal{N}\{\mathbf{P}^{(0,i_1,i_2,\dots,i_{k+1})}(j_{k+1}, n_{0,i_1,i_2,\dots,i_{k+1}})\} \end{aligned} \quad (\text{A.5})$$

where  $k = 1, 2, \dots$ . If there are no further branches connected to the end of the branch path  $\mathbf{P}^{(0,i_1,i_2,\dots,i_k)}(j_k, n_{0,i_1,i_2,\dots,i_k})$ , i.e.,  $B_{0,i_1,i_2,\dots,i_k} = 0$ , then

$$\prod_{i_{k+1}=1}^0 [\bullet] = 1$$

**Proof 3.** Let us first consider the candidate patterns of the abnormal symptoms appearing only on the initial path  $\mathbf{P}^{(0)}(m, n_0)$ . The pattern number of such a single-path composite SOO can be determined according to Theorem 2, i.e.

$$N_{CP} = N_{m,n_0} = \binom{m+n_0}{m}$$

Notice that the left-hand side can be rewritten as

$$\begin{aligned} \binom{m+n_0}{m} &= \sum_{l=0}^m \binom{l+n_0-1}{l} = \\ &= \sum_{j=0}^m \binom{m-j+n_0-1}{m-j} = \sum_{j=0}^m N_{m,n_0}^{(j)} \end{aligned}$$

where

$$N_{m,n_0}^{(j)} = \binom{m-j+n_0-1}{m-j}$$

represents the number of patterns in which the  $n_0$ th node assumes the  $j$ th-level disturbance value.

Let us next consider the candidate patterns of the abnormal symptoms appearing only on the initial path  $\mathbf{P}^{(0)}(m, n_0)$  and/or the subsequent second-tier branch paths  $\mathbf{P}^{(0,i_1)}(m, n_{0,i_1})$  ( $i_1 = 1, 2, \dots, B_0$ ). If the  $n_0$ th node on the initial path remains in its normal state, then none of the symptoms on  $\mathbf{P}^{(0,i_1)}(m, n_{0,i_1})$  are observable and the corresponding pattern number should be  $N_{m,n_0}^{(0)}$ . On the other hand, if the  $n_0$ th node on the initial path assumes the  $j_1$ th-level disturbance value ( $j_1 = 1, 2, \dots, m$ ), then the corresponding pattern

number can also be determined on the basis of Theorem 2, i.e.

$$N_{m,n_0}^{(j_1)} \prod_{i_1=1}^{B_0} N_{j_1,n_0,i_1}$$

where

$$N_{j_1,n_0,i_1} = \binom{j_1 + n_0, i_1}{j_1}$$

Thus, the total number of candidate patterns in this case should be

$$N_{CP} = N_{m,n_0}^{(0)} + \sum_{j_1=1}^m N_{m,n_0}^{(j_1)} \prod_{i_1=1}^{B_0} N_{j_1,n_0,i_1} = \binom{m+n_0-1}{m} + \sum_{j_1=1}^m \binom{m-j_1+n_0-1}{m-j_1} \prod_{i_1=1}^{B_0} N_{j_1,n_0,i_1}$$

Since  $N_{j_1,n_0,i_1}$  is again the pattern number of a single-path composite SOO with  $j_1$  disturbance levels and  $n_0, i_1$  measurement nodes, the above approach can be used repeatedly to derive the pattern number associated with a tree-shaped SOO having the third- and higher-tier branch paths.

Q.E.D.

### Appendix B: Pattern Generation Algorithm for the Tree-Shaped SOOs

To facilitate illustration of the proposed algorithm, it is necessary to first introduce the concept of *adjacent list*<sup>28</sup> for representation of the SOO structure. In particular, the neighboring nodes of every symptom in SOO are classified with two distinct lists, i.e., the precedent list and the succedent list. The former list contains all preceding nodes and the latter all succeeding nodes. For example, let us consider the fictitious SOO in Figure B.1a. In each node of this SOO, the capital letter is used to represent a measured variable and the lower-case letter

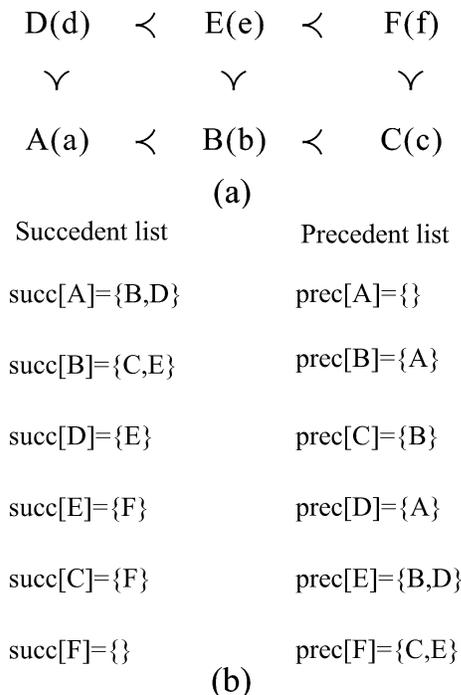


Figure B.1. (a) Fictitious SOO. (b) Corresponding succedent and precedent lists.

```

//initialize modified depth-first search
input parameters: pattern,label,prec,succ
output parameters: none
mdfs_initialization(pattern,label,prec,succ){
  for(i=1~pattern.node_number)
    visit[i]=false
  mdfs_recursion(pattern,label,prec,succ)
}

```

```

//perform modified depth-first search recursively
input parameters: pattern,label,prec,succ
output parameters: none
mdfs_recursion(pattern,label,prec,succ){
  pattern_generation(pattern,label,prec)
  visit[label]=true
  pointer=succ[label]
  while(pointer!=null){
    i=pointer.key
    if(!visit[i] and (check_preceding_nodes(i,prec)))
      mdfs_recursion(pattern,i,prec,succ)
    pointer=pointer.next
  }
}

```

```

//check if all preceding nodes of label have been visited
input parameters: label,prec
output parameters: boolean.variable1
check_preceding_nodes(label,prec){
  pointer=prec[label];
  while(pointer!=null){
    if(!visit[pointer.key])
      return false
    pointer=pointer.next}
  return true
}

```

```

//insert label into every pattern containing all its preceding nodes
input parameters: pattern,label,prec
output parameters: none
pattern_generation(pattern,label,prec){
  for(i=1~pattern.string_number){
    if(verify_pattern(pattern,i,label,prec))
      add_pattern(i,label,pattern)
  }
}

```

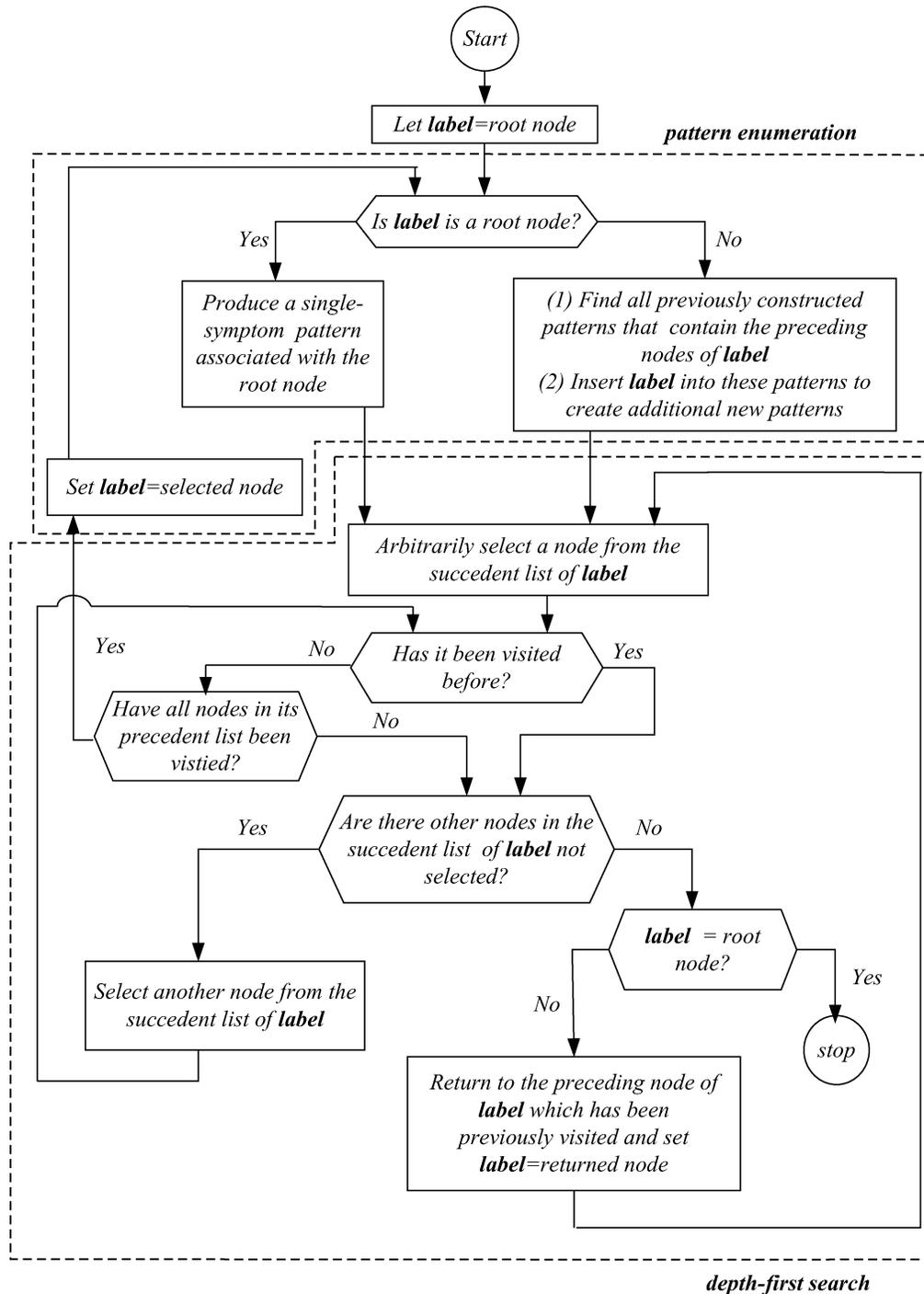
```

//verify if the preceding nodes of label are included in pattern i
input parameters: pattern,i,label,prec
output parameters: boolean.variable2
verify_pattern(pattern,i,label,prec){
  pointer=prec[label]
  while(pointer!=null){
    if(pattern.item[i][pointer.key]==1)
      pointer=pointer.next
    else
      return false
  }
  return true
}

```

in parenthesis denotes its qualitative deviation value. The corresponding adjacent lists can be found in Figure B.1b. Notice that the deviation values are dropped in these lists for the sake of brevity.

In essence, two basic tasks are performed with the pattern generation algorithm, i.e., depth-first search and pattern enumeration. The original procedure was proposed by Chen and Chang<sup>1</sup> for the simple SOOs with only one disturbance level. A modified algorithm has been developed in the present study



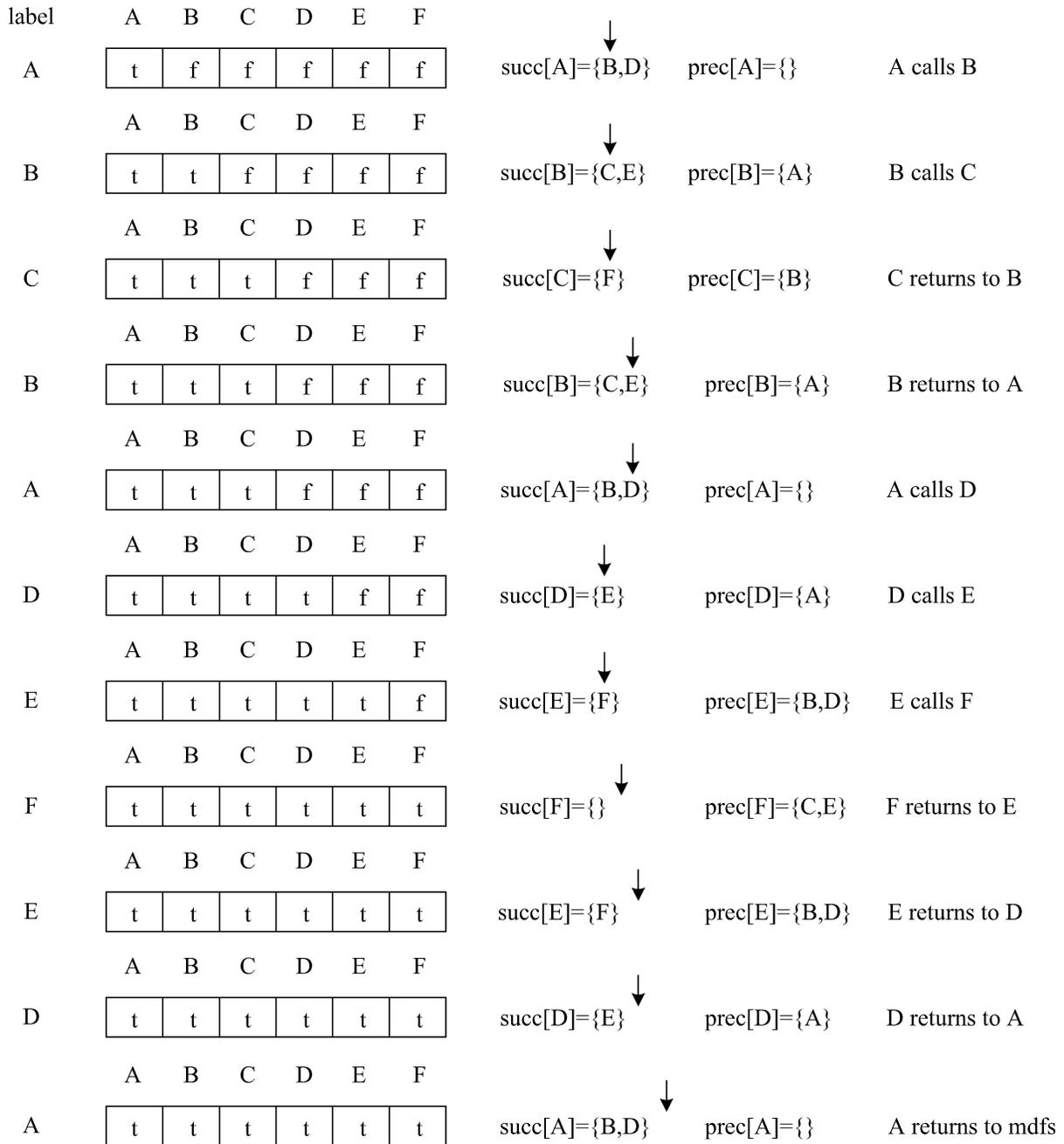
**Figure B.2.** Flowchart of pattern generation algorithm.

to generate all candidate patterns associated with any tree-shaped composite SOO. This updated version is described with the following pseudocode and also the flowchart given in Figure B.2.

It is assumed that parameter *pattern* in the above pseudo code can be characterized by a data structure with three separate fields, i.e., *pattern.node\_number* (representing the total number of nodes in the SOO), *pattern.string\_number* (representing the current number of constructed patterns), and *pattern.item[i][j]* (denoting if the *i*th pattern item is the *j*th node of the SOO). It is also assumed that the initial value of *label* is the root node and the precedent list and succedent list of every node in the SOO are given.

Let us first illustrate the modified depth-first search procedure with the example SOO shown in Figure B.1a. The corresponding

search steps are traced in Figure B.3. In the beginning of the search process, the function *mdfs\_initialization* is called according to a given root node, i.e., *A(a)*. All elements in array *visit* are then initialized to be false. Next, the function *mdfs\_recursion* is activated from *mdfs\_initialization* while assigning *A* to be the value of *label*. In the function *mdfs\_recursion*, *visit[A]* is set to be true and then function *pattern\_generation* is called to create all patterns related to node *A(a)*. The following step is to examine one of the succeeding nodes of *A(a)*, e.g., *B(b)*, using the *check\_preceding\_nodes* function. It is required in the proposed procedure that all preceding nodes of a node qualified for the next visitation must have already been visited in the previous steps. Since node *B(b)* has not been considered before and its only preceding node *A(a)* has already been visited, the function *mdfs\_recursion* should be called again. In the next call



**Figure B.3.** Iteration steps in the modified depth-first search procedure for the SOO in Figure B.1a.

of *mdfs\_recursion*, *label* = B and, thus, *visit*[B] should be set to be true. Consequently, *pattern\_generation* should be called and then one of the succeeding nodes of *B(b)*, e.g., *C(c)*, must be checked. Similarly, since *C(c)* has not been considered before and its only preceding node *B(b)* has already been visited, the function *mdfs\_recursion* must again be called. Notice that this procedure cannot be carried out further since there is only one node in the succeeding list of node *C(c)*, i.e., *F(f)*, but one of its preceding nodes *E(e)* has *not* been visited before. In such a case, the subsequent search steps should be carried out by returning to the previous caller *B(b)* in order to identify additional qualified nodes. Notice that *B(b)* has two succeeding nodes *C(c)* and *E(e)*. Since *C(c)* has already been visited while one of the preceding nodes of *E(e)* (i.e., *D(d)*) has not been visited, it is necessary to go further back to consider the succeeding nodes of *A(a)*. At this point, *D(d)* can be identified to be a qualified node. This process should then be continued

until no qualified nodes can be identified and the returned caller is the root node from function *mdfs\_initialization*.

Let us next illustrate the pattern enumeration process with the fictitious SOO in Figure B.1a. First of all, it should be noted that the possibility of a normal system state should be considered. In other words,  $\{A(0), B(0), C(0), D(0), E(0), F(0)\}$  should always be regarded as the first candidate pattern. The other candidate patterns are associated with the abnormal system states. By following the modified depth-first search procedure, the nodes in SOO may be visited according to the sequence *A(a)B(b)C(c)D(d)E(e)F(f)*. The abnormal-state candidate patterns can be generated by processing the items in this sequence one-by-one. Obviously, the initial pattern associated with the first item should be  $\{A(a), B(0), C(0), D(0), E(0), F(0)\}$ . This pattern can be written in an abbreviated form  $\{A\}$  (pattern 2) by removing all symptoms with zero deviations and also dropping the nonzero deviation value of the occurred symptom.

The candidate pattern corresponding to the second item in the search sequence can be obtained by inserting an additional symptom  $B(b)$  into a previously constructed pattern which contains all its preceding nodes, i.e.,  $A(a)$ . More specifically, this operation yields  $\{A, B\}$  (pattern 3). The candidate patterns associated with the third item  $C(c)$  can also be produced by inserting the corresponding symptom into the available pattern(s) containing its only predecessor  $B(b)$ . Since the completed patterns at this point are  $\{A\}$  and  $\{A, B\}$ , the latter can be adopted to generate a new pattern  $\{A, B, C\}$  (pattern 4). The fourth item  $D(d)$  in search sequence can be processed in a similar fashion. From Figure B.1a, it can be observed that the predecessor of the node corresponding to this item is  $A(a)$ . Thus, the new patterns in this case should be  $\{A, D\}$  (pattern 5),  $\{A, B, D\}$  (pattern 6), and  $\{A, B, C, D\}$  (pattern 7). Notice that the fifth item  $E(e)$  has two preceding nodes in the SOO, i.e.,  $B(b)$  and  $D(d)$ . The corresponding new patterns can be created by augmenting this item with pattern 5 and pattern 6, i.e.,  $\{A, B, D, E\}$  (pattern 8) and  $\{A, B, C, D, E\}$  (pattern 9). Finally, the pattern containing the last item can be produced by adding it to pattern 9, i.e.,  $\{A, B, C, D, E, F\}$  (pattern 10). Notice that, by substituting 2 and 3 for  $m$  and  $n$ , respectively, into eq 7, the resulting total number of candidate patterns is also 10. This pattern generation algorithm has been coded with Visual C++<sup>29,30</sup> in the present study and tested extensively with different SOOs.

## Appendix C: Process Models

### C.1. Mathematical Model Used for Simulating a Single-Tank, Level-Control System.

$$A \frac{dh}{dt} = q_1 + q_3 - q_2 \quad (C.1)$$

$$q_2 = C_1 \sqrt{h_1} \quad (C.2)$$

$$q_1 = q_{1s} + K_{cv} K_c \left[ (h_{set} - h) + \frac{1}{\tau_I} \int_0^t (h_{set} - h) \right] \quad (C.3)$$

Table TC.1. Model Parameters Used in Example 1

parameter	definition	(steady-state) value
$A$	cross-sectional area of tank	$10^4 \text{ cm}^2$
$h$	height of liquid level in tank	50 cm
$q_1$	input flow rate	$707.1 \text{ cm}^3/\text{s}$
$q_2$	output flow rate	$707.1 \text{ cm}^3/\text{s}$
$q_3$	additional input flow rate	$0 \text{ cm}^3/\text{s}$
$C_1$	proportional constant	$100 \text{ cm}^2\text{-}^5/\text{s}$
$K_{cv}$	control-valve gain	$15 \text{ cm}^3/\text{s} \%$
$h_{set}$	set point of level height	50 cm
$K_c$	proportional gain of level controller	0.02
$\tau_I$	integral time of level controller	20 s

### C.2. Mathematical Model Used for Simulating Two-Tank, Level-Control System.

$$A_1 \frac{dh_1}{dt} = q_1 + q_4 - q_2 \quad (C.4)$$

$$A_2 \frac{dh_2}{dt} = q_2 + q_5 - q_3 \quad (C.5)$$

$$q_2 = C_1 \sqrt{h_1} \quad (C.6)$$

$$q_3 = C_2 \sqrt{h_2} \quad (C.7)$$

$$q_1 = q_{1s} + K_{cv} K_c \left[ (h_{2,set} - h_2) + \frac{1}{\tau_I} \int_0^t (h_{2,set} - h_2) \right] \quad (C.8)$$

Table TC.2. Model Parameters Used in Example 2

parameter	definition	(steady-state) value
$A_1$	cross-sectional area of tank 1	$10^4 \text{ cm}^2$
$h_1$	height of liquid level in tank 1	50 cm
$A_2$	cross-sectional area of tank 2	$10^4 \text{ cm}^2$
$h_2$	height of liquid level in tank 2	50 cm
$q_1$	input flow rate of tank 1	$711.0 \text{ cm}^3/\text{s}$
$q_2$	output flow rate of tank 1	$711.0 \text{ cm}^3/\text{s}$
$q_3$	output flow rate of tank 2	$711.0 \text{ cm}^3/\text{s}$
$q_4$	additional input flow rate to tank 1	$0 \text{ cm}^3/\text{s}$
$q_5$	additional input flow rate to tank 2	$0 \text{ cm}^3/\text{s}$
$C_1, C_2$	proportional constants	$100.56 \text{ cm}^2\text{-}^5/\text{s}$
$K_{cv}$	control-valve gain	$14.22 \text{ cm}^3/\text{s} \%$
$h_{2,set}$	set point of the level height in tank 2	50 cm
$K_c$	proportional gain of level controller	1.281
$\tau_I$	integral time of level controller	$7.95 \times 10^{-4} \text{ s}$

### C.3. Mathematical Model Used for Simulating CSTR System.

$$\frac{dV}{dt} = F_0 - F \quad (C.9)$$

$$V = A_T h \quad (C.10)$$

$$r_A = k_0 e^{-(E/RT)} C_A \quad (C.11)$$

$$\frac{dC_A}{dt} = \frac{F_0}{V} (C_{A_0} X_A - C_A) - r_A \quad (C.12)$$

$$\frac{dT}{dt} = \frac{F_0}{V} (T_0 - T) + \frac{r_A (-\Delta H)}{\rho C_p} - \frac{UA(T - T_c)}{V \rho C_p} \quad (C.13)$$

$$\frac{dT_c}{dt} = \frac{F_c}{V_j} (T_{c_0} - T_c) + \frac{UA(T - T_c)}{V_j \rho_j C_j} \quad (C.14)$$

$$F_c = F_{cs} - K_c^T \left[ (T_{set} - T) + \frac{1}{\tau_I} \int_0^t (T_{set} - T) \right] \quad (C.15)$$

$$F = F_s - K_c^H \left[ (h_{set} - h) + \frac{1}{\tau_I} \int_0^t (h_{set} - h) \right] \quad (C.16)$$

Table TC.3. Model Parameters Used in Example 3

parameter	definition	(steady-state) value
$h$	height of liquid level in reactor	48 ft
$X_A$	molar fraction of reactant at reactor inlet	0.47
$V$	reactor volume	$4800 \text{ ft}^3$
$C_{A_0}$	reactant concentration in feed	$1 \text{ lb-mol}/\text{ft}^3$
$T$	reactor temperature	$537 \text{ }^\circ\text{R}$
$F_0$	feed flow rate	$4000 \text{ ft}^3/\text{h}$
$T_0$	feed temperature	$530 \text{ }^\circ\text{R}$
$C_A$	reactant concentration in reactor	$0.474 \text{ lb-mol}/\text{ft}^3$
$T_c$	outlet coolant temperature	$537 \text{ }^\circ\text{R}$
$F_c$	coolant flow rate	$4836 \text{ ft}^3/\text{h}$
$V_j$	volume of jacket	$385 \text{ ft}^3$
$k_0$	frequency factor	$7.08 \times 10^{10}/\text{h}$
$E$	activation energy	$30\,000 \text{ Btu}/\text{lb-mol}$
$R$	universal gas constant	$1.99 \text{ Btu}/(\text{lb-mol } ^\circ\text{R})$
$U$	overall heat transfer coefficient	$150 \text{ Btu}/(\text{h ft}^2 \text{ }^\circ\text{R})$
$A$	heat transfer area	$25\,000 \text{ ft}^2$
$T_{c_0}$	inlet coolant temperature	$530 \text{ }^\circ\text{R}$
$\Delta H$	heat of reaction	$-30\,000 \text{ Btu}/\text{lb-mol}$
$C_p$	heat capacity (process side)	$0.72 \text{ Btu}/(\text{lb}_m \text{ }^\circ\text{R})$
$C_j$	heat capacity (coolant side)	$1 \text{ Btu}/(\text{lb}_m \text{ }^\circ\text{R})$
$\rho$	density of process mixture	$50 \text{ lb}_m/\text{ft}^3$
$\rho_j$	Density of coolant	$62.3 \text{ lb}_m/\text{ft}^3$
$A_r$	cross-sectional area of reactor	$100 \text{ ft}^2$
$K_c^H$	proportional gain of level controller	10
$K_c^T$	proportional gain of temperature controller	80
$\tau_I^H$	integral time of level controller	89.286 h
$\tau_I^T$	integral time of temperature controller	0.6557 h
$h_{set}$	set point of the level height in tank	48 ft
$T_{set}$	set point of the temperature in tank	$537 \text{ }^\circ\text{R}$

## Literature Cited

- (1) Chen, J. Y.; Chang, C. T. Fuzzy diagnosis method for control systems with coupled feed forward and feedback loops. *Chem. Eng. Sci.* **2006**, *61*, 3105.
- (2) Chang, C. T.; Chen, J. W. Implementation issues concerning the ekf-based fault diagnosis technique. *Chem. Eng. Sci.* **1995**, *50*, 2861.
- (3) Petti, T. F.; Klein, J.; Dhurjati, P. S. Diagnostic model processor: Using deep knowledge for process fault diagnosis. *AIChE J.* **1990**, *36*, 565.
- (4) Carrasco, E.; Rodríguez, J.; Puñal, A.; Roca, E.; Lema, J. Diagnosis of acidification states in an anaerobic wastewater treatment plant using a fuzzy-based expert system. *Control Eng. Pract.* **2004**, *12*, 59.
- (5) Hoskins, J. C.; Kalivur, K. M.; Himmelblan, D. M. Fault diagnosis in complex chemical-plants using artificial neural networks. *AIChE J.* **1991**, *37*, 137.
- (6) Iri, M.; Aoki, K.; O'Shima, E.; Matsuyama, H. An algorithm for diagnosis of system failure in the chemical process. *Comput. Chem. Eng.* **1979**, *3*, 489.
- (7) Kramer, M. A.; Palowitch, B. L. A rule-based approach to fault diagnosis using the signed directed graph. *AIChE J.* **1987**, *33*, 1067.
- (8) Shiozaki, J.; Matsuyama, H.; O'Shima, E.; Iri, M. An improved algorithm for diagnosis of system failures in the chemical process. *Comp. Chem. Eng.* **1985**, *9*, 285.
- (9) Tsuge, Y.; Shiozaki, J.; Matsuyama, H.; O'Shima, E. Fault diagnosis algorithms based on the signed directed graph and its modifications. *Inst. Chem. Eng. Symp. Ser.* **1985**, *92*, 133.
- (10) Maurya, M. R.; Rengaswamy, R.; Venkatasubramanian, V. Application of signed digraph-based analysis for fault diagnosis chemical process flowsheets. *Eng. Appl. Artif. Intell.* **2004**, *17*, 501.
- (11) Tarifa, E.; Scenna, N. Fault diagnosis for msf dynamic states using a sdg and fuzzy logic. *Desalination* **2004**, *166*, 93.
- (12) Zhang, Z.; Wu, C.; Zhang, B.; Xia, T.; Li, A. SDG multiple fault diagnosis by real-time inverse inference. *Reliab. Eng. Syst. Saf.* **2005**, *87*, 173.
- (13) Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37.
- (14) Catelani, M.; Fort, A.; Alippi, C. A fuzzy approach for soft fault detection in analog circuits. *Measurement* **2002**, *32*, 73.
- (15) Wang, J.; Hu, H. Vibration-based fault diagnosis of pump using fuzzy technique. *Measurement* **2006**, *39*, 185.
- (16) Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S. N. A review of process fault detection and diagnosis, part: Quantitative model based methods. *Comput. Chem. Eng.* **2003**, *27*, 293.
- (17) Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. N. A review of process fault detection and diagnosis, part: Qualitative model and search strategies. *Comput. Chem. Eng.* **2003**, *27*, 313.
- (18) Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. N.; Yin, K. A review of process fault detection and diagnosis, part: Process history based methods. *Comput. Chem. Eng.* **2003**, *27*, 327.
- (19) Maurya, M. R.; Rengaswamy, R.; Venkatasubramanian, V. A signed directed graph-based systematic framework for steady state malfunction diagnosis control loops. *Chem. Eng. Sci.* **2006**, *61*, 1790.
- (20) Chang, S. Y.; Lin, C. R.; Chang, C. T. A fuzzy diagnosis approach using dynamic fault trees. *Chem. Eng. Sci.* **2002**, *57*, 2971.
- (21) Chang, S. Y.; Chang, C. T. A fuzzy-logic based fault diagnosis strategy for process control loops. *Chem. Eng. Sci.* **2003**, *58*, 3395.
- (22) Kuipers, B. *Qualitative Reasoning*; The MIT Press: Cambridge, MA, 1994.
- (23) Dash, S.; Rengaswamy, R.; Venkatasubramanian, V. Fuzzy-logic based trend classification for fault diagnosis of chemical process. *Comput. Chem. Eng.* **2003**, *27*, 347.
- (24) Ju, S. N.; Chen, C. L.; Chang, C. T. Constructing fault trees for advanced process control systems—Application to cascade control loops. *IEEE Trans. Reliab.* **2004**, *53*, 43.
- (25) Oyeleye, O. O.; Kramer, M. A. Qualitative simulation of chemical process systems: steady-state analysis. *AIChE J.* **1988**, *34*, 1441.
- (26) Mathworks *SIMULINK—Dynamic System Simulation for MATLAB Using Simulink*; The Mathworks Inc.: Natick, MA, 2002.
- (27) Mathworks *Fuzzy Logic Toolbox—User Guide*; The Mathworks Inc.: Natick, MA, 2002.
- (28) Johnsonbaugh, R.; Schaefer, M. *Algorithms*; Pearson Education: Upper Saddle River, NJ, 2004.
- (29) Petzold, C. *Programming Windows*; Microsoft Press: Redmond, WA, 1998.
- (30) LaJoie, J.; Lippman, S. B. *C++ Primer*, 3rd ed.; Addison Wesley Professional: Upper Saddle River, NJ, 1998.

Received for review August 21, 2006  
 Revised manuscript received March 12, 2007  
 Accepted March 16, 2007

IE061103E