PERGAMON

Chemical Engineering Science 58 (2003) 3395–3411

Chemical Engineering Science

www.elsevier.com/locate/ces

A fuzzy-logic based fault diagnosis strategy for process control loops

Sheng-Yung Chang, Chuei-Tin Chang*

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

Received 28 October 2002; received in revised form 24 April 2003; accepted 12 May 2003

Abstract

By considering the fault propagation behaviors in process systems *with* control loops, a fuzzy-logic based fault diagnosis strategy has been developed in the present work. The proposed fault diagnosis methods can be implemented in two stages. In the off-line preparation stage, the fault origins of a system hazard are identified by determining the minimal cut sets of the corresponding fault tree. The fault propagation patterns in a feedback loop are obtained on the basis of system digraph. The occurrence order of observable symptoms caused by each fault origin is derived accordingly and then encoded into a set of IF–THEN diagnosis rules. In the next on-line diagnosis stage, the occurrence indices of the top event and also the fault origins are computed in a fuzzy inference system based on real-time measurement data. Simulation studies have been carried out to demonstrate the feasibility of the proposed approach.

Keywords: Fault diagnosis; Feedback control; Symptom occurrence order; Candidate pattern; Fuzzy logic

1. Introduction

As a result of increasing complexity in the modern chemical processes, development of the computer-aided on-line fault diagnosis techniques has become an important issue for plant operation. A wide variety of different approaches have already been proposed in the literature, e.g. the expert systems (Petti, Klein, & Dhurjati, 1990), the state observers (Chang & Chen, 1995), the neural networks (Hoskins, Kalivur, & Himmeblau, 1991) and the signed directed graphs (SDG) (Iri, Aoki, O'Shima, & Matsuyama, 1979), etc. Ulerich and Powers (1988) reported the first attempt to perform fault diagnosis on the basis of fault trees. The most significant advantage of their approach is that the candidates of fault identification can be restricted to only the causes of one or more given top events and, consequently, the diagnosis procedure can be greatly simplified. By incorporating the propagation patterns of fault origins in a fuzzy inference system, Chang, Lin, and Chang (2002) developed a systematic fault diagnosis procedure for processes *without* control loops. This method was implemented in two stages: the off-line preparation stage and the on-line implementation stage. In the former case, a SDG system model was first constructed and the fault trees corresponding to the given top events were then synthesized according to the Lapp-and-Powers algorithm (Lapp & Powers, 1977). The symptom occurrence order caused by the basic events in each cut set can be easily determined with the qualitative simulation techniques on the basis of the SDG model (Chang & Hwang, 1992). In addition, two theorems were developed to facilitate enumeration of all possible symptom patterns that may be observed during operation. These candidate patterns were then translated into a set of IF–THEN inference rules for assessing the occurrence possibilities of the basic events in every cut set and also the top events. In the next stage, the on-line measurement data were first compared with the normal steady-state values of the process variables. The process deviations were determined accordingly and then used as the inputs to a fuzzy inference system for computing the occurrence indices of top events and cut sets in real time.

Since the dynamic responses of feedback control loops were not considered in the original fault diagnosis system mentioned above (Chang et al., 2002), a more general fault diagnosis strategy is developed in the present study to enhance its capability. Specifically, in addition to the initial and final states used in the previous studies, the transient state of each loop variable is introduced to describe the symptom

* Corresponding author. Tel.: +886-6-275-7575x62663; fax: +886-6-234-4496.

E-mail address: ctchang@mail.ncku.edu.tw (C.-T. Chang).

0009-2509/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0009-2509(03)00218-5

occurrence order more accurately. Three more theorems can then be developed to enumerate all possible symptom patterns by considering the loop dynamics. The effectiveness of this new feature has been verified with extensive simulation studies.

The rest of this paper is organized as follows. In Section 2, the procedure for enumerating all possible fault propagation mechanisms in a loop-free system is reviewed to facilitate later discussions. A recursive formula is also provided here to determine the total number of symptom patterns that may be observed on-line. Section 3 is concerned with the candidate patterns in systems *with* control loops. Three formal theorems are presented for counting these patterns. The architecture of fuzzy inference system is given in Section 4. The practices used in this study to classify measurement data and also diagnosis results are described in detail. In addition, a systematic encoding method is developed for translating the candidate patterns and other insights into four types of IF–THEN rules. Finally, The effectiveness of the proposed strategy is demonstrated with extensive simulation studies in Section 5.

2. The candidate patterns in loop-free systems

The effects of base event(s) identified in a fault tree usually propagate throughout the entire system sequentially. In general, a series of intermediate events may occur before the inception of a designated top event. Since the performance of a diagnosis scheme should be evaluated not only in terms of its correctness but also its timeliness, it is the intention of this research to develop a fault identification procedure taking both the eventual symptoms and also their *occurrence order* into consideration. To identify this *symptom occurrence order* (SOO) associated with a given fault origin, the following operations can be performed on the system digraph (Chang et al., 2002): (1) apply the techniques of qualitative simulation to identify the *fault propagation paths* (FPPs), (2) merge every pair of measured variable and its measurement signal in the FPPs, and then (3) eliminate the nodes representing the unmeasured variables.

To illustrate the above procedure, let us consider the level control system in Fig. 1 as an example. In this process, there are two input streams, i.e. streams 1 and 3, and one output stream, i.e. stream 2, connected to the liquid storage tank T-01. The level control loop consists of the proportional-integral (PI) controller LIC-01 and the control valve CV-01 on stream 1. The flow rates of streams 1 and 2 are monitored with flow meters FI-01 and FI-02, respectively. It is assumed that the gate valve on the stream 2 (V-02) is open and the globe valve on stream 3 (V-03) closed during normal operation.

A SDG model can be constructed according to Fig. 1 (see Fig. 2). The nodes in this digraph are mainly associated with the process variables, the measurement signals and the control signals. Generally speaking, the process variables

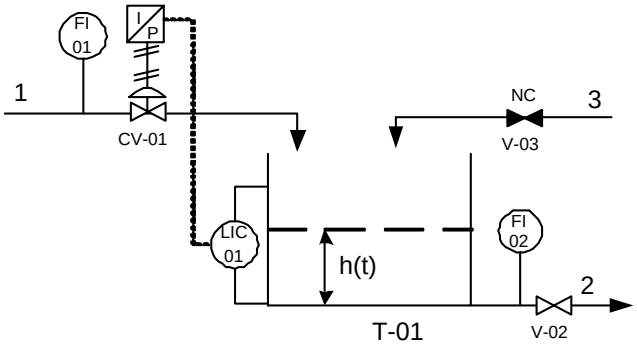

Fig. 1. Level control system.

considered in this case are mass flow rate and liquid level. They are represented respectively by combining an initial letter, i.e. m and h, with a numerical label or a subscript. The numerical labels are basically the stream numbers in the process flow diagram, while the subscripts are always associated with process units. Thus, h_{T-01} denotes the liquid level of the tank T-01 and $m1$ represent the mass flow rate of stream 1. Notice also that a measurement signal is represented by the tag of the corresponding sensor or controller with a superscript "m". The same approach is adopted to denote the controller outputs to drive the control valves, i.e. they are represented by the controller tags in process flow diagram with a superscript "c".

A set of five values, i.e. $\{-10,-1,0,+1,+10\}$, may be assigned to each node in a SDG to *qualitatively* represent deviation from the normal value of corresponding variable. "0" means that it is under 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 non-zero 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 condition 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, ±1 and ±10. The output value of an arc can be computed with the gain and its input value according to the following equation:

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

where g, v_{in} and v_{out} denote, respectively, the gain, input and output values. Finally, it should be noted that, other than the normal arc, one or more conditional arc may be added between two nodes. These arcs are valid only under the specified conditions.

Let us select the event "high liquid level in the tank," i.e. $h_{T-01}(+1)$, as one of the top events for diagnosis

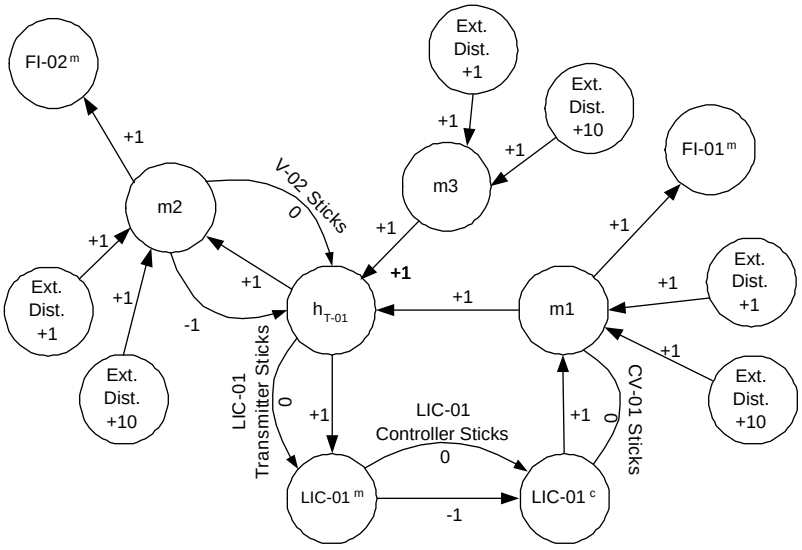

Fig. 2. Digraph model of level control system.

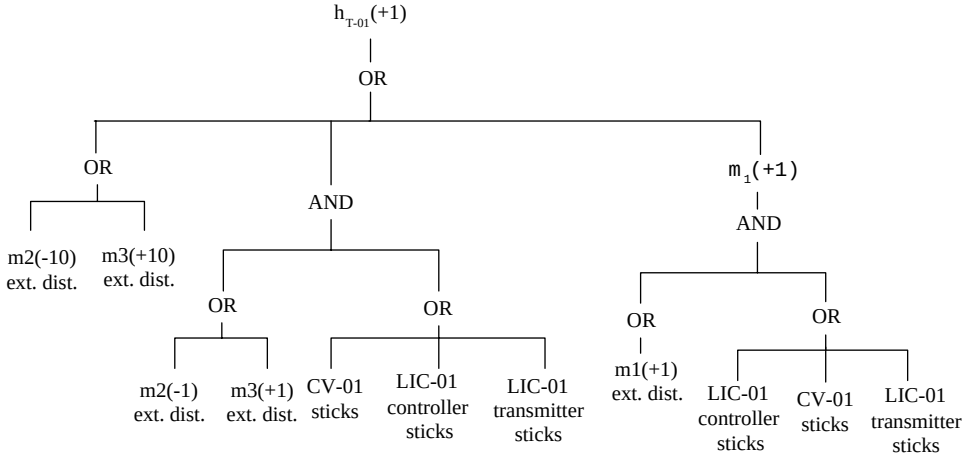

Fig. 3. Fault tree with top event "liquid level in tank T-01 is too high," i.e. $h_{T-01}(+1)$.

purpose. The conventional Lapp-and-Powers algorithm (Lapp & Powers, 1977) is adopted in this study to synthesize the corresponding fault tree on the basis of Fig. 2 (see Fig. 3). The resulting minimal cut sets can be found in Table 1. To illustrate the proposed method in determining the SOOs, let us examine the 9th cut set, i.e. $\{m_3(+1)$, CV-01 sticks$\}$. The corresponding SOO can be found according to the proposed procedure (see Fig. 4). It should be stressed that the digraph configuration is changed under the influence of the failure "CV-01 sticks". Since the gain between nodes LIC-01^c and $m1$ now becomes zero, the resulting system should be considered as *loop free*.

If all symptoms in a SOO can be observed on-line, then it is certainly reasonable to confirm the existence of corresponding fault origin(s). However, it is also possible to find that these symptoms are only partially developed during the incipient period of an eventual system hazard and, further, their pattern may vary at different instances during operation. To facilitate later discussions, let us define the collection of on-line symptoms at any time after the introduction of basic event(s) in a cut set as a *candidate pattern*. It is obvious that any candidate pattern can be considered

Table 1
The minimal cut sets of the fault tree in Fig. 3

MCS no.	Flow rate			CV-01 sticks	LIC-01 transmitter sticks	LIC-01 controller sticks
	m_1	m_2	m_3			
1		−10				
2			+10			
3	+1			Y		
4	+1				Y	
5	+1					Y
6		−1		Y		
7		−1			Y	
8		−1				Y
9			+1	Y		
10			+1		Y	
11			+1			Y

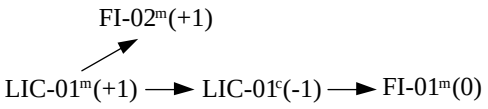

Fig. 4. SOO of cut set 9.

as an evidence for fault identification with a degree of confidence. Thus, it is important to enumerate all possible candidate patterns and evaluate their respective significance in the off-line preparation stage.

Notice that a SOO in general assumes the form of a tree in a loop-free system. The total number of candidate patterns in this case can be computed with a theorem developed by Chang et al. (2002):

Theorem 1. *Consider a tree-shaped SOO* **T**. *If* $\mathbf{P}^{(0)}(n_0)$ *denotes the initial path of length* n_0 *in* **T**, $\mathbf{P}^{(0,i_1)}(n_{0,i_1})$ $(i_1 = 1,2,\ldots,N_0)$ *denotes the* i_1*th branch path of length* n_{0,i_1} *connecting to the end of* $\mathbf{P}^{(0)}(n_0)$, $\mathbf{P}^{(0,i_1,i_2)}(n_{0,i_1,i_2})$ $i_2 = 1,2,\ldots,N_{0,i_1})$ *denotes the* i_2*th branch path of length* n_{0,i_1,i_2} *connecting to the end of* $\mathbf{P}^{(0,i_1)}(n_{0,i_1})$, *etc., then the total number of candidate patterns* N_{CP} *can be computed according to the following equation*:

$$N_{CP} = \mathcal{N}_1\{\mathbf{P}^{(0)}(n_0)\} = n_0 + \prod_{i_1=1}^{N_0} \mathcal{N}_1\{\mathbf{P}^{(0,i_1)}(n_{0,i_1})\}, \quad (2)$$

where $\mathcal{N}_1\{\bullet\}$ *denotes the counting operator for a path in a tree-shaped SOO. The result of this operation should be generated recursively, i.e.*

$$\mathcal{N}_1\{\mathbf{P}^{(0,i_1,i_2,\ldots,i_k)}(n_{0,i_1,i_2,\ldots,i_k})\} = n_{0,i_1,i_2,\ldots,i_k} + \prod_{i_{k+1}=1}^{N_{0,i_1,i_2,\ldots,i_k}} \mathcal{N}_1\left\{\mathbf{P}^{(0,i_1,i_2,\ldots,i_{k+1})}(n_{0,i_1,i_2,\ldots,i_k,i_{k+1}})\right\} \quad (3)$$

and $k = 1,2,\ldots$.

If there are no further branches connected to the end of the branch path $\mathbf{P}^{(0,i_1,i_2,\ldots,i_k)}(n_{0,i_1,i_2,\ldots,i_k})$, *i.e.* $N_{0,i_1,i_2,\ldots,i_k} = 0$, *then*

$$\prod_{i_{k+1}=1}^{0} [\bullet] = 1. \quad (4)$$

The proof of this theorem is omitted in this paper for the sake of brevity. Instead, a simple example is provided below to illustrate the enumeration procedure:

Example 1. Consider the SOO of cut set 9 shown in Fig. 4. The total number of candidate patterns can be calculated using Theorem 1, i.e.

$$\begin{aligned} N_{CP} &= \mathcal{N}_1\{\text{LIC-01}^m\} \\ &= 1 + \mathcal{N}_1\{\text{FI-02}^m\} \cdot \mathcal{N}_1\{\text{LIC-01}^c \rightarrow \text{FI-02}^m\} \\ &= 1 + \left(1 + \prod_{}^{0}[\bullet]\right) \cdot \left(2 + \prod_{}^{0}[\bullet]\right) \\ &= 1 + (1+1)\cdot(2+1) \\ &= 7. \end{aligned}$$

Table 2
The candidate patterns of cut set 9

No.	LIC-01^m	LIC-01^c	FI-01^m	FI-02^m
1	0	0	0	0
2	+1	0	0	0
3	+1	0	0	+1
4	+1	−1	0	0
5	+1	−1	0	+1
6	+1	−1	0*	0
7	+1	−1	0*	+1

The corresponding candidate patterns are listed in Table 2. Notice that there are two pairs of seemingly identical patterns. The first is associated with rows 4 and 6, and the second can be found in rows 5 and 7. This is mainly due to the fact that "FI − $01^m(0)$" is treated as a symptom in SOO. In other words, the patterns in each pair should be interpreted differently. In rows 6 and 7, asterisks are used to denote the fully developed symptoms. On the other hand, the corresponding unmarked "0" in row 4 or 5 indicates that the effects of abnormal disturbances have not reached FI-01.

3. The candidate patterns in process control loops

In a loop-free process, the net effect of fault propagation on a process variable can be viewed qualitatively as the direct transition from the normal system state to another new state. However, if a process contains feedback loops, the intermediate transient states caused by the compensation action of the controller must be considered. For illustration purpose, let us again consider the level control system in Fig. 1 and examine the scenario caused by the basic event in cut set 2 (see Table 2). The transient response of the control system can be simulated with SIMULINK (Mathworks, 2000b). In the simulation studies, the height of tank wall is taken as 100 cm and the outlet flow rate is assumed to be proportional to the square root of liquid-level height. The initial values of the liquid level and the flow rate of streams 3 are 50 cm and 0 g/s, respectively. It is also assumed that the system is at steady state initially and the initial flow rates of streams 1 and 2 are the same, i.e. 707 g/s.

Typical simulation results can be found in Figs. 5(A) and (B). Notice that these results were generated with two different sets of controller parameters, i.e. $K_p = 8$ and $\tau_I = 5$ were used in Case (A) and $K_p = 0.02$ and $\tau_I = 20$ in Case (B). It can be observed that the dynamic behavior of the system is dependent upon the compensation speed of controller. If

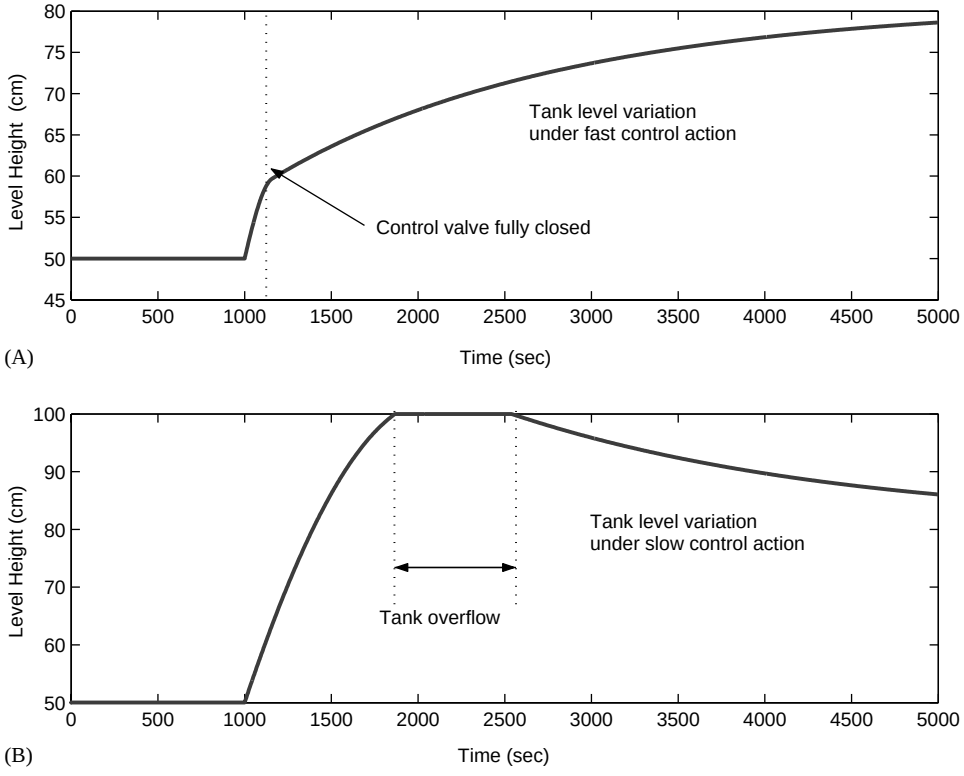

Fig. 5. Transient responses of level control system caused by cut set 2: (A) fast compensation; (B) slow compensation.

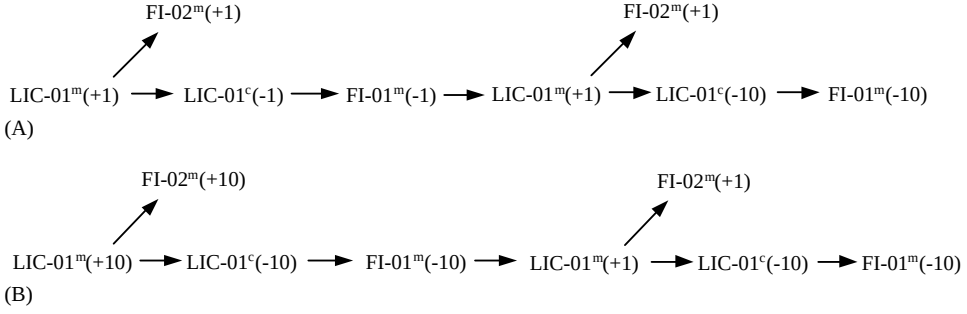

Fig. 6. Single-valued SOO of cut set 2: (A) fast compensation; (B) slow compensation.

the compensation speed is "fast," i.e. Case (A), the control valve is fully closed shortly after the occurrence of an uncontrollable disturbance (accidental opening of valve V-03 during operation) and the height of liquid level rises gradually to an abnormally high value. The symptoms of fault propagation can be described qualitatively with the SOO in Fig. 6(A). On the other hand, a "slow" controller may fail to prevent tank overflow despite the fact that the flow of stream 1 is blocked eventually (Case B). This scenario can be characterized with the SOO in Fig. 6(B). To facilitate a concise representation of the candidate patterns, these two alternative SOOs are written in a unified two-valued format as shown in Fig. 7. The two values in each parenthesis denote, respectively, the transient and final states of the corresponding loop variable.

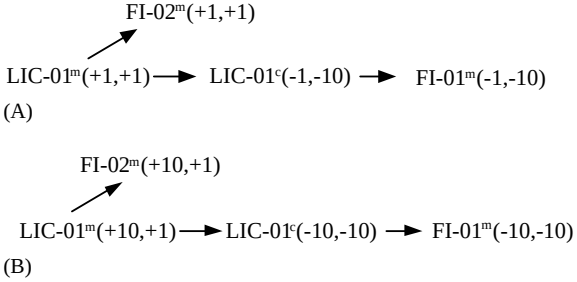

Fig. 7. Two-valued SOOs of cut set 2: (A) fast compensation; (B) slow compensation.

Let us first consider the simplest case of a single-path SOO on the feedback loop, i.e. $s_1(\delta_1^t, \delta_1^f) \rightarrow s_2(\delta_2^t, \delta_2^f) \rightarrow \cdots \rightarrow$

$s_n(\delta_n^t, \delta_n^f)$. A detailed analysis of the candidate patterns in Fig. 6 reveals that,

(1) The transient state of any measured variable should always be observed before its final state. More specifically, the symptom $s_j(\delta_j^t)$ should appear before $s_j(\delta_j^f)$ and $j = 1, 2, \ldots, n$.
(2) The transient and final states of measured variables in the single-path SOO can only be confirmed sequentially. In other words, the symptom $s_j(\delta_j^t)$ should occur earlier than $s_{j+1}(\delta_{j+1}^t)$ and $s_j(\delta_j^f)$ earlier than $s_{j+1}(\delta_{j+1}^f)$. Here, $j = 1, 2, \ldots, n-1$.
(3) As a result of the inherent feedback mechanism, the final state of the first measured variable in SOO cannot be detected until after the last measured variable reaches its transient state, i.e. $s_1(\delta_1^f)$ follows *only* $s_n(\delta_n^t)$.

Consequently, a set of rules can be developed for generating the candidate patterns:

- The normal state of one-loop variable and the final state of another cannot coexist in the same candidate pattern.
- The states of adjacent loop variables must either be the same or follow their precedence order in time. More specifically, there are only five possibilities: (1) $\{s_j(\delta_j^f), s_{j+1}(\delta_{j+1}^f)\}$; (2) $\{s_j(\delta_j^t), s_{j+1}(\delta_{j+1}^t)\}$; (3) $\{s_j(0), s_{j+1}(0)\}$; (4) $\{s_j(\delta_j^f), s_{j+1}(\delta_{j+1}^t)\}$; (5) $\{s_j(\delta_j^t), s_{j+1}(0)\}$.

On this basis of the above two pattern generation rules, the following theorem can be derived:

Theorem 2. *Let* $\mathbf{Y}$ *be the set of all measured variables and* $\mathbf{\Delta} = \{-10, -1, +10, +1\}$. *If* $\tilde{\mathbf{S}}(n) = s_1(\delta_1^t, \delta_1^f) \rightarrow s_2(\delta_1^2, \delta_f^2) \rightarrow \cdots \rightarrow s_n(\delta_n^t, \delta_n^f)$ *(where* $\forall s_j \in \mathbf{Y}$, $\forall \delta_j^t \in \mathbf{\Delta}$, $\forall \delta_j^f \in \mathbf{\Delta}$ *and* $j = 1, 2, \ldots, n$*) denotes a two-valued single-path SOO on a feedback control loop, then the total number of* candidate patterns N_{CP} *can be computed with the following equation*:

$$N_{CP} = 2n + 1. \tag{5}$$

This theorem is proved in Appendix A. On the other hand, from the two-valued SOOs in Fig. 7, it can also be observed that some of the branch paths may *not* be located on the feedback loop. The variables on these branches may experience time-variant disturbances emanating from the loop. Thus, it is necessary to generalize Theorem 1 to account for the possibility of multiple disturbances propagating in a tree. It should be noted that the first candidate generation rule mentioned above is no longer valid in this case. Furthermore, the possibility of an additional pattern between two adjacent variable, i.e. $\{s_j(\delta_j^f), s_{j+1}(0)\}$, should also be included in implementing the second rule. Consequently, the following theorem can be derived to compute the total number of candidate patterns:

Theorem 3. *Consider a tree-shaped two-valued SOO* $\hat{\mathbf{T}}$. *If* $\hat{\mathbf{P}}^{(0)}(n_0)$ *denotes the initial path of length* n_0 *in* $\hat{\mathbf{T}}$, $\hat{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})$ $(i_1 = 1, 2, \ldots, N_0)$ *denotes the* i_1*th branch path of length* n_{0,i_1} *connecting to the end of* $\hat{\mathbf{P}}^{(0)}(n_0)$, $\hat{\mathbf{P}}^{(0,i_1,i_2)}(n_{0,i_1,i_2})$ $(i_2 = 1, 2, \ldots, N_{0,i_1})$ *denotes the* i_2*th branch path of length* n_{0,i_1,i_2} *connecting to the end of* $\hat{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})$, *etc., then the total number of candidate patterns* N_{CP} *can be determined by computing* $\mathcal{N}_3\{\hat{\mathbf{P}}^{(0)}(n_0)\}$, *i.e.* $N_{CP} = \mathcal{N}_3\{\hat{\mathbf{P}}^{(0)}(n_0)\}$ *and*

$$\mathcal{N}_3\{\hat{\mathbf{P}}^{(0)}(n_0)\} = \frac{n_0(n_0+1)}{2} + n_0 \prod_{i_1=1}^{N_0} \mathcal{N}_1\{\hat{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\} + \prod_{i_1=1}^{N_0} \mathcal{N}_3\{\hat{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\}, \tag{6}$$

where $\mathcal{N}_1\{\bullet\}$ *denotes the counting operator defined in Theorem* 1, *and* $\mathcal{N}_3\{\bullet\}$ *denotes a new counting operator for a path in the tree-shaped two-valued SOO. The result of this operation should be generated recursively, i.e.*

$$\begin{aligned}\mathcal{N}_3\{\hat{\mathbf{P}}^{(0,i_1,i_2,\ldots,i_k)}(n_{0,i_1,i_2,\ldots,i_k})\} &= \frac{n_{0,i_1,i_2,\ldots,i_k}(n_{0,i_1,i_2,\ldots,i_k}+1)}{2} \\ &+ n_{0,i_1,i_2,\ldots,i_k} \prod_{i_{k+1}=1}^{N_{0,i_1,i_2,\ldots,i_k}} \mathcal{N}_1\{\hat{\mathbf{P}}^{(0,i_1,i_2,\ldots,i_{k+1})}(n_{0,i_1,i_2,\ldots,i_k,i_{k+1}})\} \\ &+ \prod_{i_{k+1}=1}^{N_{0,i_1,i_2,\ldots,i_k}} \mathcal{N}_3\{\hat{\mathbf{P}}^{(0,i_1,i_2,\ldots,i_{k+1})}(n_{0,i_1,i_2,\ldots,i_k,i_{k+1}})\}\end{aligned} \tag{7}$$

and $k = 1, 2, \ldots$.

If there are no further branches connected to the end of the branch path $\hat{\mathbf{P}}^{(0,i_1,i_2,\ldots,i_k)}(n_{0,i_1,i_2,\ldots,i_k})$, *i.e.* $N_{0,i_1,i_2,\ldots,i_k} = 0$, *then*

$$\prod_{i_{k+1}=1}^{0} [\bullet] = 1. \tag{8}$$

The proof of this theorem is provided in Appendix B. Furthermore, it should be noted that a hybrid SOO, i.e. one that contains a feedback loop and also branch paths not on the loop, is often encountered in practical applications. To facilitate enumeration of candidate patterns in this situation, the above two theorems have been combined to produce the following theorem:

Theorem 4. *Consider a tree-shaped two-value hybrid SOO* $\tilde{\mathbf{T}}$ *containing branch paths both on and off a feedback control loop. If* $\tilde{\mathbf{P}}^{(0)}(n_0)$ *denotes the initial path of length* n_0 in $\tilde{\mathbf{T}}$, $\tilde{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})$ $(i_1 = 1, 2, \ldots, N_0)$ *denotes the* i_1*th branch path of length* n_{0,i_1} *connecting to the end of* $\tilde{\mathbf{P}}^{(0)}(n_0)$,

$\tilde{\mathbf{P}}^{(0,i_1,i_2)}(n_{0,i_1,i_2})$ $(i_2 = 1,2,\ldots,N_{0,i_1})$ *denotes the* i_2*th branch path of length* n_{0,i_1,i_2} *connecting to the end of* $\tilde{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})$, *etc., and* $\tilde{\mathbf{P}}^{(0)}(n_0)$, $\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})$, $\tilde{\mathbf{P}}^{(0,1,1)}(n_{0,1,1}),\ldots$ *are branch paths on the loop, then the total number of candidate patterns* N_{CP} *can be determined according to the following equation*:

$$N_{CP} = n_0 + [(n_0 - 1)\mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})\} + \mathcal{N}_1\{\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})\}]\prod_{i_1=2}^{N_0}\mathcal{N}_1\{\tilde{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\} + \mathcal{N}_{4,(2)}\{\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})\}\prod_{i_1=2}^{N_0}\mathcal{N}_3\{\tilde{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\}, \quad (9)$$

where $\mathcal{N}_1\{\bullet\}$ *and* $\mathcal{N}_3\{\bullet\}$ *denotes the counting operators defined in Theorems* 1 *and* 3, *respectively, and* $\mathcal{N}_{4,(1)}\{\bullet\}$ *and* $\mathcal{N}_{4,(2)}\{\bullet\}$ *denote two new counting operators for a loop path in the tree-shaped hybrid SOO. The results of these operations can be generated recursively, i.e.*

$$\mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1,\ldots,1)}(n_{0,1,\ldots,1})\} = \mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1,\ldots,1,1)}(n_{0,1,\ldots,1,1})\} \times \prod_{i_{k+1}=2}^{N_{0,1,\ldots,1}}\mathcal{N}_1\{\tilde{\mathbf{P}}^{(0,1,\ldots,1,i_{k+1})}(n_{0,1,\ldots,1,i_{k+1}})\}, \quad (10)$$

$$\mathcal{N}_{4,(2)}\{\tilde{\mathbf{P}}^{(0,1,\ldots,1)}(n_{0,1,\ldots,1})\} = n_{0,1,\ldots,1}\mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1,\ldots,1,1)}(n_{0,1,\ldots,1,1})\} \times \prod_{i_{k+1}=2}^{N_{0,1,\ldots,1}}\mathcal{N}_1\{\tilde{\mathbf{P}}^{(0,1,\ldots,1,i_{k+1})}(n_{0,1,\ldots,1,i_{k+1}})\} + \mathcal{N}_{4,(2)}\{\tilde{\mathbf{P}}^{(0,1,\ldots,1,1)}(n_{0,1,\ldots,1,1})\} \times \prod_{i_{k+1}=2}^{N_{0,1,\ldots,1}}\mathcal{N}_3\{\tilde{\mathbf{P}}^{0,1,\ldots,1,i_{k+1}}(n_{0,1,\ldots,1,i_{k+1}})\}. \quad (11)$$

If there are no further branches connected to the end of the branch path $\tilde{\mathbf{P}}^{(0,1,\ldots,1)}(n_{0,1,\ldots,1})$, *i.e.* $N_{0,1,\ldots,1} = 0$, *then*

$$\mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1,\ldots,1)}(n_{0,1,\ldots,1})\} = 1, \quad (12)$$

$$\mathcal{N}_{4,(2)}\{\tilde{\mathbf{P}}^{(0,1,\ldots,1)}(n_{0,1,\ldots,1})\} = n_{0,1,\ldots,1} + 1. \quad (13)$$

A proof of the above theorem can be found in Appendix C. The following example demonstrates the use of Theorem 4:

Example 2. Let us again consider cut set 2 in Table 1. If the compensation speed is "fast," the candidate patterns of cut set can be determined according to Fig. 7(A). We can then compute the total number of candidate patterns with Theorem 4:

$$N_{CP} = 1 + [(1-1) + (2+1)](1+1) + (2+1) \times \left[\frac{1\times(1+1)}{2} + 1\times 1 + 1\right] = 1 + 6 + 9 = 16. \quad (14)$$

Table 3
The candidate patterns of cut set 2

No.	LIC-01^m	LIC-01^c	FI-01^m	FI-02^m
1	0	0	0	0
2	+1 (t)	0	0	0
3	+1 (t)	0	0	+1 (t)
4	+1 (t)	−1 (t)	0	0
5	+1 (t)	−1 (t)	0	+1 (t)
6	+1 (t)	−1 (t)	−1 (t)	0
7	+1 (t)	−1 (t)	−1 (t)	+1 (t)
8	+1 (f)	−1 (t)	−1 (t)	0
9	+1 (f)	−1 (t)	−1 (t)	+1 (t)
10	+1 (f)	−1 (t)	−1 (t)	+1 (f)
11	+1 (f)	−10 (f)	−1 (t)	0
12	+1 (f)	−10 (f)	−1 (t)	+1 (t)
13	+1 (f)	−10 (f)	−1 (t)	+1 (f)
14	+1 (f)	−10 (f)	−1 (f)	0
15	+1 (f)	−10 (f)	−1 (f)	+1 (t)
16	+1 (f)	−10 (f)	−1 (f)	+1 (f)

The corresponding candidate patterns are presented in Table 3. Notice that, in the parenthesis next to each deviation value, a label "*t*" is used to denote the transient state and "f" the final state.

4. Fuzzy inference system

In this work, the final product prepared off-line is a *fuzzy inference system* (FIS). A sketch of its framework is presented in Fig. 8. If this system is to be implemented on-line, the measurement data must be first converted to a set of process deviations with respect to the given reference values and, then, used as inputs to FIS. The core of FIS is a

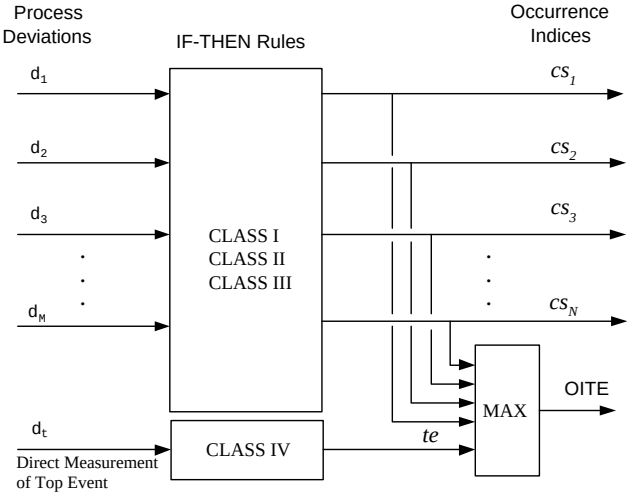

Fig. 8. Framework of fuzzy inference system.

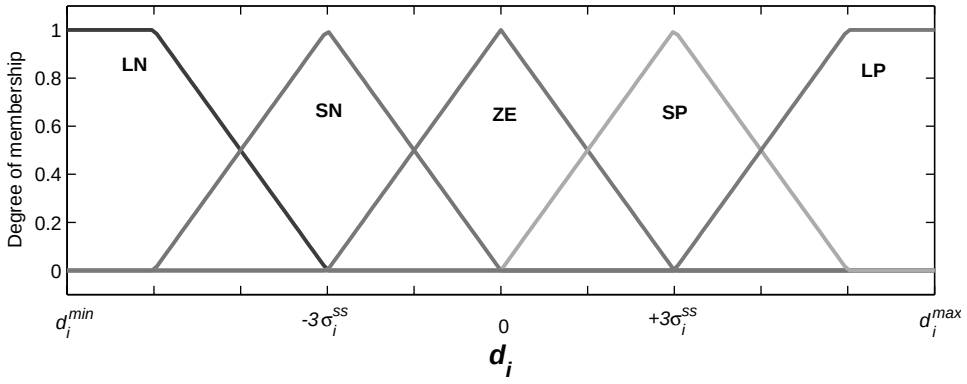

Fig. 9. Classification of process deviations.

collection of IF–THEN rules, which can be further divided into to four distinct classes. The outputs of FIS are the occurrence index of top event OITE and also that of every minimal cut set cs_k ($k = 1, 2, \ldots$). The former can be used to forecast a potential incident and the latter are adopted to rank all possible fault origins. The inference mechanism used in this system can be found in standard textbooks, e.g., George and Yuan (1995) and Ross (1995). The main elements of FIS are described in detail in the following subsections.

4.1. Classification of process deviations

Since the deviation values associated with disturbances and/or failures in a typical fault tree are *qualitative* in nature, it is necessary to introduce a more consistent description in fault diagnosis applications. In particular, a set of membership functions are used in this work for the purpose of classifying the on-line measurements into several fuzzy sets. Specifically, let us denote the ith process measurement as y_i and the same measurement at steady state as y_i^{ss}. The process deviation of the ith measured variable, d_i, can be determined according to the following equation:

$$d_i = y_i - \mu_i^{ss}, \tag{15}$$

where μ_i^{ss} denotes the mean of y_i^{ss}. In this study, five linguistic values, i.e. LN (large negative), SN (small negative), ZE (zero) and SP (small positive) and LP (large positive), are assigned to each process deviation to describe the qualitative concepts of $-10, -1, 0, +1$ and $+10$. The corresponding membership functions can be constructed according to Fig. 9. In particular, the linguistic values SN, ZE and SP are represented with standard triangular membership functions. The locations of their apexes are determined on the basis of the standard deviation of y_i^{ss}, i.e.

$$V_i^{SN} = -3\sigma_i^{ss}, \tag{16}$$

$$V_i^{ZE} = 0, \tag{17}$$

$$V_i^{SP} = +3\sigma_i^{ss}, \tag{18}$$

where V_i^{SN}, V_i^{ZE} and V_i^{SP} represent the apex locations and σ_i^{ss} is the standard deviation of y_i^{ss}. On the other hand, the membership functions of LP and LN are trapezoids. The locations of their upper interior corners are chosen on the basis of process knowledge and/or operation experience. For example, in the level control system described previously, the membership value of LP for LIC-01^m can be considered as 1 if the level measurement exceeds 100 cm. As another example, the upper and lower alarm limits (if exist) of a measured variable can be used for the same purpose.

The bottom corners of the triangular membership functions can be conveniently set to the apex and/or upper-corner locations of the neighboring functions. Specifically,

$$B_{i,l}^{SN} = V_i^{LN}, \quad B_{i,r}^{SN} = V_i^{ZE}, \tag{19}$$

$$B_{i,l}^{ZE} = V_i^{SN}, \quad B_{i,r}^{ZE} = V_i^{SP}, \tag{20}$$

$$B_{i,l}^{SP} = V_i^{ZE}, \quad B_{i,r}^{SP} = V_i^{LP}. \tag{21}$$

In the above equations, the symbol B on the left is used to denote the locations of the bottom corners. The corresponding linguistic values are specified as its superscript. Notice that two subscripts are used to distinguish these bottom corners. The first is the label of the measurement variable and the second denotes its position relative to the apex, i.e. left or right. Also notice that V_i^{LN} and V_i^{LP} represent, respectively, the locations of upper interior corners of the trapeziods corresponding to LN and LP. The locations of interior corners at the bottom of LN and LP can be determined in the same fashion, i.e.

$$B_{i,r}^{LN} = V_i^{SN}, \tag{22}$$

$$B_{i,l}^{LP} = V_i^{SP}. \tag{23}$$

Finally, it is assumed that the smallest and largest possible values of the ith process deviation, i.e. $d_i^{\min}$ and $d_i^{\max}$, can always be determined in advance. The locations of two remaining corner points of the trapezoidal functions for LN are both set at $d_i^{\min}$ and those for LP are set at $d_i^{\max}$.

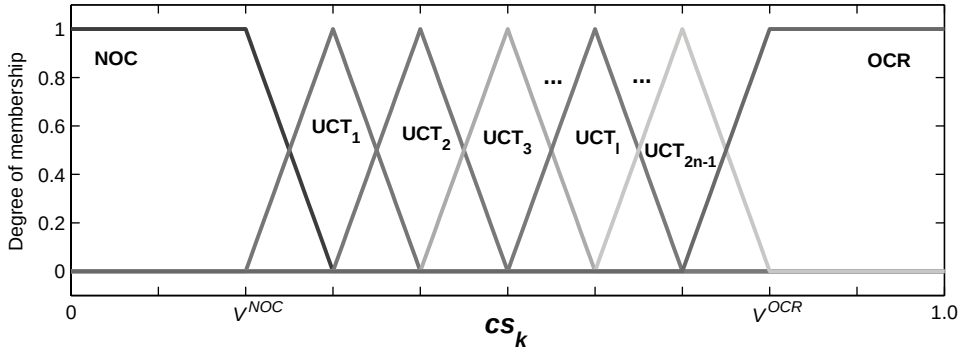

Fig. 10. Classification of occurrence indices.

4.2. Classification of occurrence indices

As mentioned previously, an occurrence index can be viewed as the diagnosis concerning a top event or the basic events in a cut set. In general, three types of fuzzy sets, i.e. OCR, NOC and UCT$_\ell$ are adopted in this work for its characterization. The symbol OCR is used to denote the belief that the corresponding event(s) will occur soon or have already occurred and, on the other hand, NOC is concerned with the opposite conclusion, i.e. the possibility of fault can be ruled out. Finally, UCT$_\ell$ is essentially a less definite statement between OCR and NOC. The subscript ℓ is used to reflect the degree of confidence toward confirmation of the corresponding event(s).

Membership functions used in this work for classifying the occurrence indices are either triangular or trapezoidal as shown in Fig. 10. Notice that all diagnostic conclusions can be characterized with the same set of membership functions. The upper left corner of NOC and the upper right corner of OCR are placed at 0 and 1, respectively. The locations of their upper interior corners, i.e., V^{NOC} and V^{OCR}, should be selected before other triangular membership functions can be determined. If the number of measurements in a two-valued SOO is n, then a total of $2n-1$ triangular membership functions are used to characterize UCT$_\ell$s. The locations of their apexes (V_ℓ^{UCT}) are determined according to the following equation:

$$V_\ell^{UCT} = V^{NOC} + \ell\,\frac{V^{OCR} - V^{NOC}}{2n}, \tag{24}$$

where $\ell = 2\ell_f + \ell_t$, and ℓ_f and ℓ_t denote the numbers of matched final and transient symptoms, respectively. The locations of bottom corners of each triangle are again chosen to be the apex locations of the neighboring functions.

4.3. Generation of inference rules

As mentioned previously, the IF–THEN inference rules can be divided into four different classes. The classes I and II rules are used for evaluating the existence potential of all basic events in a cut set, i.e. cs_k ($k = 1, 2, 3, \ldots$). The class III rules are mainly used to enhance diagnostic resolution. Finally, the class IV rules are used to compute a preliminary index for the top event, i.e. *te*.

The class I rules can be derived from the candidate patterns with the help of Theorem 4. To facilitate later discussions, let us first define a *linguistic interpretation function* F as follows:

$$F(\delta_j^*) = \begin{cases} \text{LN} & \text{if } \delta_j^* = -10, \\ \text{SN} & \text{if } \delta_j^* = -1, \\ \text{ZE} & \text{if } \delta_j^* = 0, \\ \text{SP} & \text{if } \delta_j^* = +1, \\ \text{LP} & \text{if } \delta_j^* = +10, \end{cases} \tag{25}$$

where $\delta_j^* = \delta_j^t$ or δ_j^f, and $j = 1, 2, \ldots, n$. If the on-line symptoms are identical to those in a SOO, then it is highly possible that they are caused by the corresponding fault origin. Consequently, the following rule is incorporated in FIS to assert such a belief:

IF d_1 is $F(\delta_1^f)$ AND d_2 is $F(\delta_2^f)$ AND
$\cdots$ AND d_n is $F(\delta_n^f)$
THEN cs_k is OCR. (26)

On the other hand, it is quite reasonable to disregard the possibility of a fault if none of the symptoms in the corresponding SOO can be observed. Thus, the following rule is also included:

IF d_1 is ZE AND d_2 is ZE AND $\cdots$ AND d_n is ZE
THEN cs_k is NOC. (27)

The remaining $N_{CP} - 2$ candidate patterns are translated into rules with uncertain conclusions, i.e. UCT$_\ell$ and $\ell = 1, 2, \ldots, 2n-1$. The premises of each rule can be determined by substituting the qualitative deviation values in a candidate pattern into the linguistic interpretation function F. The number of matched transient and final symptoms in the candidate pattern is used to compute the degree of confidence ℓ in conclusion UCT$_\ell$.

Since five deviation values are used to characterize each symptom, the total number of *all* possible patterns should be 5^n. Obviously, there are $5^n - N_{CP}$ non-candidate patterns and the conclusions of the corresponding rules should all be NOC. If all such rules are included in FIS, the resulting computation load may become overwhelming. On the other hand, the inference system is bound to perform poorly without them. To be more specific, let us consider the situation when none of the candidate patterns can be matched. Obviously, the membership value of cs_k obtained by firing the class I rules should be very small at every point in the interval $[0, 1]$ and thus its centroid must be around 0.5. In other words, the diagnosis is uncertain. This is unacceptable since the correct occurrence index should be 0, i.e. the possibility of cut set k should be ruled out if none of the candidate patterns are matched. To solve this dilemma, a *single* class II rule is used in the FIS instead. This rule can be written as

$$\begin{aligned}&\text{IF } d_1 \text{ is NOT ZE} \quad \text{AND} \quad d_2 \text{ is NOT ZE AND} \cdots \\ &\quad \text{AND } d_n \text{ is NOT ZE} \\ &\text{THEN } cs_k \text{ is NOC}, \quad (\omega_k),\end{aligned} \tag{28}$$

where $0 < \omega_k \ll 1$. Notice that ω_k here denotes a weighting factor introduced in firing the rule. If it is the only triggered rule, then the resulting occurrence index should be 0 (not 0.5). If one or more additional rule is also triggered, then the impact of class II rule should be negligible due to this very small weighting factor. Implementation of the class II rule is realized with the Fuzzy Logic Toolbox in MATLAB (Mathworks, 2000a) in this work.

Usually not all measured variables in a system are affected by the basic events in a cut set. In this study, the set of all measurements included in the SOO of the kth cut set is referred to as its *range of influence* $\mathbf{ROI}_k$. Thus, if a measurement variable is not a member of $\mathbf{ROI}_k$, then it should remain unaffected by the corresponding basic event(s) in cut set k. In this work, the class III rules are used mainly to reduce the chance of misdiagnosis due to a different fault origin, say the basic events in cut set k', causing the same candidate patterns within $\mathbf{ROI}_k$. A set of corresponding class III rules can be written as

$$\begin{aligned}&\text{IF } d_u \text{ is SP THEN } cs_k \text{ is NOC} \\ &\text{IF } d_u \text{ is SN THEN } cs_k \text{ is NOC} \\ &\text{IF } d_u \text{ is LP THEN } cs_k \text{ is NOC} \\ &\text{IF } d_u \text{ is LN THEN } cs_k \text{ is NOC}\end{aligned} \tag{29}$$

or simply

$$\text{IF } d_u \text{ is NOT ZE} \quad \text{THEN } cs_k \text{ is NOC}, \tag{30}$$

where cs_k is the occurrence index of the kth cut set and $u \notin \mathbf{ROI}_k$. If cut set k is indeed the correct root cause, then the rule in Eq. (30) is not supposed to be triggered. On the other hand, in the case of cut set k' occurring, these rules should definitely lower the value of cs_k. Thus, by assuming that the possibility of more than one fault origin occurring is negligible, the class III rules can enhance the diagnostic resolution. However, the drawback of including them in FIS is that the possibility of a non-minimal cut set or multiple cut sets may also be ignored. In other words, if the basic events in cut set k form a subset of all existing events, these rules may cause a decrease in the occurrence index cs_k. Therefore, this trade-off between resolution and comprehensiveness in diagnosis must be evaluated carefully before introducing the class III rules in practical applications. In this study, they are included on the ground that the probability of additional events occurring is extremely low.

Finally, if the top event is monitored directly with a sensor, then this information should be encoded with class IV rules. Since there is only one measurement involved, five inference rules can be constructed to cover all possibilities. As an example, by assuming that "d_t is SP" is the event causing undesirable consequences, one can express the corresponding rules as

$$\begin{aligned}&\text{IF } d_t \text{ is LP THEN } te \text{ is OCR} \\ &\text{IF } d_t \text{ is SP THEN } te \text{ is OCR} \\ &\text{IF } d_t \text{ is ZE THEN } te \text{ is NOC} \\ &\text{IF } d_t \text{ is SN THEN } te \text{ is NOC} \\ &\text{IF } d_t \text{ is LN THEN } te \text{ is NOC}\end{aligned} \tag{31}$$

Example 3. Let us consider the 2nd cut set in Table 1 and assume that the corresponding SOO can be described with Fig. 7(A). The class I rules in this case can be found in Table 4. The class II and class IV rules can be written as

- Class II:

$$\begin{aligned}&\text{IF LIC-01}^m \text{ is NOT ZE AND LIC-01}^c \text{ is NOT ZE AND} \\ &\quad \text{FI-01}^m \text{ is NOT ZE AND FI-02}^m \text{ is NOT ZE} \\ &\text{THEN } cs_2 \text{ is NOC (0.001)}\end{aligned} \tag{32}$$

- Class IV:

$$\begin{aligned}&\text{IF LIC-01}^m \text{ is LP THEN } te \text{ is OCR} \\ &\text{IF LIC-01}^m \text{ is SP THEN } te \text{ is OCR} \\ &\text{IF LIC-01}^m \text{ is ZE THEN } te \text{ is NOC} \\ &\text{IF LIC-01}^m \text{ is SN THEN } te \text{ is NOC} \\ &\text{IF LIC-01}^m \text{ is LN THEN } te \text{ is NOC}\end{aligned} \tag{33}$$

Finally, notice that the class III rules do not exist since all measurements are within $\mathbf{ROI}_2$.

Table 4
The IF–THEN inference rules of cut set 2

	IF				
No.	LIC-01^m	LIC-01^c	FI-01^m	FI-02^m	**THEN**
1	ZE	ZE	ZE	ZE	NOC
2	SP	ZE	ZE	ZE	UCT_1
3	SP	ZE	ZE	SP	UCT_2
4	SP	SN	ZE	ZE	UCT_2
5	SP	SN	ZE	SP	UCT_3
6	SP	SN	SN	ZE	UCT_3
7	SP	SN	SN	SP	UCT_4
8	SP	SN	SN	ZE	UCT_4
9	SP	SN	SN	SP	UCT_5
10	SP	SN	SN	SP	UCT_6
11	SP	LN	SN	ZE	UCT_5
12	SP	LN	SN	SP	UCT_6
13	SP	LN	SN	SP	UCT_7
14	SP	LN	LN	ZE	UCT_6
15	SP	LN	LN	SP	UCT_7
16	SP	LN	LN	SP	OCR

4.4. Forecast of top event

If the existence of the basic event(s) in any cut set is confirmed, then one can deduce logically that the top event is bound to occur. Thus, the combined outputs of firing classes I, II and III rules, i.e. cs_k ($k=1,2,\ldots$), can also be treated as indications of system hazard. On the other hand, the class IV rules exist only when the *direct* measurement of the variable associated with top event d_t is available. If this is the case, the final occurrence index of top event (OITE) should be obtained by taking the maximum of cs_ks and te, i.e.

$$\mathrm{OITE} = \max(cs_1, cs_2, \ldots, te). \tag{34}$$

5. Case studies

To demonstrate the effectiveness of the proposed approach, extensive simulation studies on the level control system in Fig. 1 have been carried out in this work. From the results of fault tree analysis given in Table 1, it can be observed that there are basically two types of failure mechanisms. The first two cut sets represent scenarios caused by the uncontrollable disturbances. Typical dynamic responses in these situations can be found in Figs. 5(A) and (B). The fault origins listed in remaining cut sets can be generalized as the combination of two basic events, i.e. (1) a controllable disturbance and (2) an equipment failure causing the control loop inactive.

Let us first consider the scenario caused by cut set 2, i.e. $\{m_3(+10)\}$, which is an uncontrollable disturbance. For simulation purpose, it is assumed that the system is originally at steady state and valve V-03 is mistakenly opened at time 1000 s. Consequently, a flux of 900 g/s is introduced into the storage tank via stream 3. Since this flow is larger than that of stream 1 at the normal steady state, it is apparent that its effects cannot be compensated with the level control system. The corresponding simulation data in this case were generated with SIMULINK (Mathworks, 2000b). In all simulation runs, the controller parameters associated with the "fast" control actions in Fig. 5(A) were adopted, i.e. $K_p=8$ and $\tau_I=5$. It is assumed that the inferior control performance presented in Fig. 5(B) can always be avoided with proper tuning procedure. The diagnosis results of FIS in this case are shown in Fig. 11. It can be seen that the performance of the proposed fuzzy-logic based method is quite satisfactory. Notice from Fig. 11(A) that the early symptoms of the basic event $m_3(+10)$ can be clearly detected. This observation indicates that the current compensation action of the controller is indeed fast enough for preventing the controlled variable, i.e. h_{T-01}, to exceed the height of tank wall even under the influence of large external disturbances. The occurrence index of top event is shown in Fig. 11(B). Notice that the occurrence of top event is confirmed long before a positive identification of the fault origin cs_2. This is due to the facts that (1) the level measurement LIC-01^m is the first symptom in the SOO given in Fig. 7(A), (2) the class IV rules specified in Eq. (31) should be triggered early in the fault propagation process, and (3) the occurrence index OITE is the maximum value of cs_ks and te.

Next let us consider a similar case in which a controllable disturbance is introduced, i.e. the globe valve V-03 is partially opened at 1000 s. Since the flow rate in stream 3 in this case is only 200 g/s, the liquid level can be eventually controlled at its original set point, i.e. 50 cm, by reducing the flow rate in stream 1. Thus, this flow increase in stream 3 should be expressed qualitatively as $m_3(+1)$. The simulated dynamic response of level height is shown in Fig. 12(A) and the corresponding inference results of FIS using only the IF–THEN rules for cs_2 can be found in Fig. 12(B). It can be observed that the initial finding of the diagnosis system is that the uncontrollable disturbance $m_3(+10)$ may be a candidate fault origin (with a 20% certainty). However, as the effects of disturbance $m_3(+1)$ are gradually compensated, the level height is brought back to the normal steady state. Consequently, the occurrence index cs_2 also returns to a very low value near zero around 1200 s and the occurrence possibility of $m_3(+10)$ can therefore be correctly eliminated.

The third case discussed in this paper is concerned with the second type of failure mechanisms listed in Table 1. Specifically, let us consider the scenario caused by the basic events in cut set 11, i.e. (1) a controllable disturbance is introduced in the flow rate of stream 3 and (2) the controller LIC-01 fails. Samples of simulation results for four different cut sets, i.e. cs_2, cs_9, cs_{10} and cs_{11}, are shown in Figs. 13(A)–(D), respectively. Notice first from Figs. 13(A) and (B) that the diagnostic conclusions concerning the 2nd and 9th cut sets are both uncertain. This is due to the fact the final symptoms of cut set 11 only partially match those caused

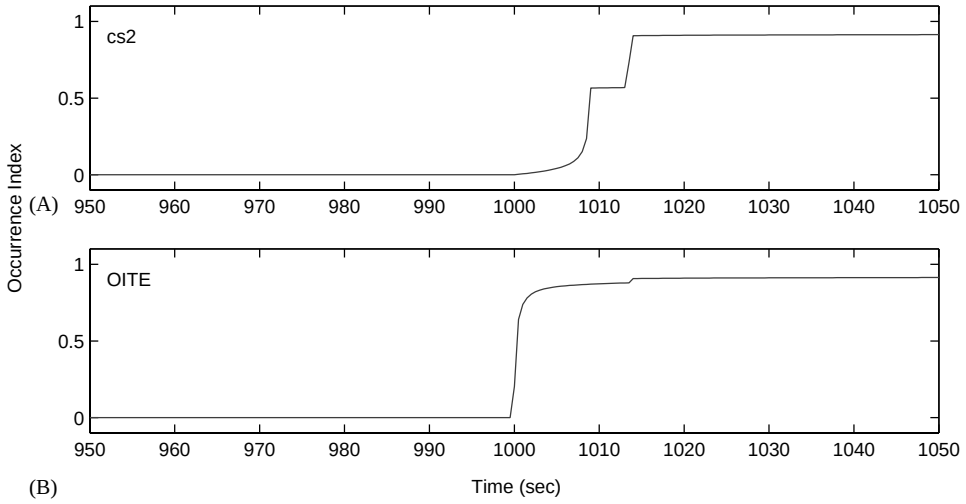

Fig. 11. Simulation results of case 1: (A) occurrence index of the 2nd cut set; (B) occurrence index of top event.

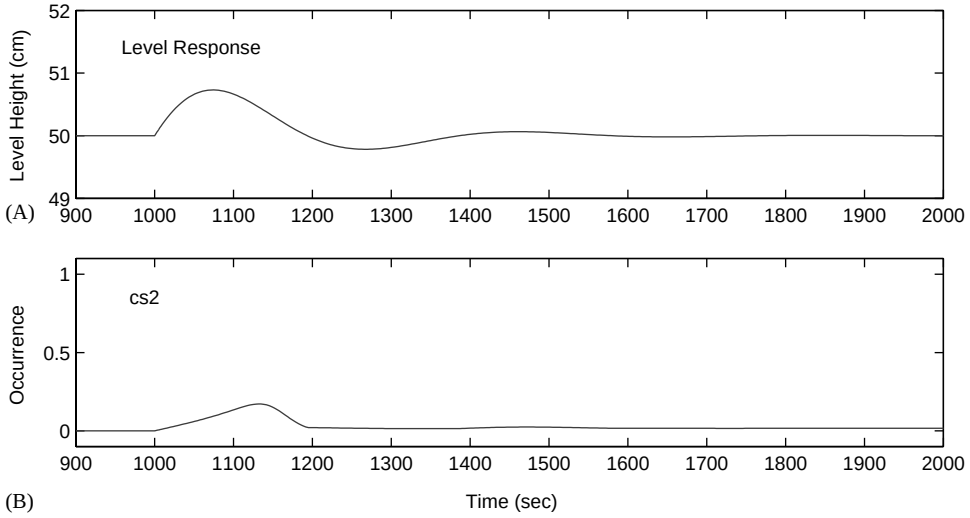

Fig. 12. Simulation results of case 2: (A) level response to a controllable disturbance; (B) occurrence index of the 2nd cut set.

by cut sets 2 and 9. More specifically, the fully developed candidate pattern of the 11th cut set is essentially the same as the 3rd candidate pattern listed in Table 3 and also the 3rd row in Table 2. In addition, notice that the eventual degree of confidence achieved in Fig. 13(A) is lower than that in Fig. 13(B). This difference can be easily explained by comparing the membership functions of the occurrence indices UCT_is used in the corresponding IF–THEN rules. From Table 1, one can also see that the cause of control-loop failure in the 10th cut set is the malfunction of sensor/transmitter and that in the 11th cut set is controller failure. The simulation results in Fig. 13(C) show that the on-line symptoms of the basic events in cut set 10 are completely matched during the early stage of diagnosis process. However, the fuzzy inference system rapidly lowers its confidence in confirming the 10th cut set after the symptoms of cut set 11 are fully developed at about 2600 s. Finally, the diagnosis results in Fig. 13(D) indicate that the actual fault origin, i.e. the basic events in cut set 11, can be correctly identified in a timely manner.

6. Conclusion

By capturing the dynamic characteristics of abnormal system behaviors with four different classes of fuzzy inference rules, an effective fault diagnosis system has been developed in this work. Four theorems have been derived to facilitate enumeration of all possible candidate patterns for diagnosis in systems with and without feedback control loops. The simulation results show that the proposed strategy can be used to identify not only the correct fault origins but also the corresponding fault propagation mechanisms as well.

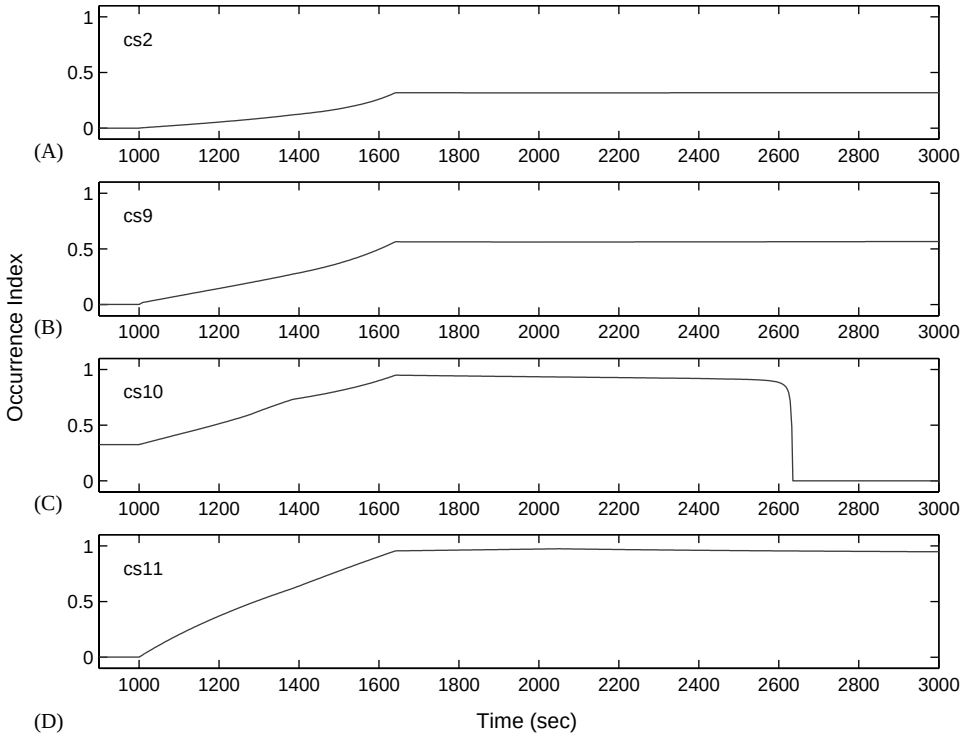

Fig. 13. Simulation results of case 3: (A) occurrence index of cut set 2; (B) occurrence index of cut set 9; (C) occurrence index of cut set 10; (D) occurrence index of cut set 11.

Notation

$B_{i,l}^{SN}/B_{i,r}^{SN}$	the location of left/right bottom corner of triangular membership function SN
$B_{i,l}^{ZE}/B_{i,r}^{ZE}$	the location of left/right bottom corner of triangular membership function NE
$B_{i,l}^{SP}/B_{i,r}^{SP}$	the location of left/right bottom corner of triangular membership function SP
$B_{i,r}^{LN}$	the location of right bottom corner of trapezoidal membership function LN
$B_{i,l}^{LP}$	the location of left bottom corner of trapezoidal membership function LP
cs_k	the occurrence index of kth cut set
d_i	the process deviation of y_i
$d_i^{\max}$	the maximum value of d_i
$d_i^{\min}$	the minimum value of d_i
F	the linguistic interpretation function
^{h}T-01	the liquid level of the tank T-01
ℓ_t/ℓ_f	the number of matched transient/final symptoms
LN	large negative—a linguistic value of the process deviation
LP	large positive—a linguistic value of the process deviation
m_i	the mass flow rate of ith stream
N_{CP}	the number of candidate patterns
NOC	not occurred—a linguistic value of the diagnostic conclusion
OCR	occurred—a linguistic value of the diagnostic conclusion
OITE	occurred index of top event
$\mathbf{P}^{(0)}(n_0)$	the initial path (of length n_0) in a single-valued tree-shaped SOO
$\mathbf{P}^{(0,i)}(n_{0,i})$	the ith branch path (of length $n_{0,i}$) from $\mathbf{P}^{(0)}$
$\mathbf{P}^{(0,i,j)}(n_{0,i,j})$	the jth branch path (of length $n_{0,i,j}$) from $\mathbf{P}^{(0,i)}$
$\hat{\mathbf{P}}^{(0)}(n_0)$	the initial path (of length n_0) in a two-valued tree-shaped SOO
$\hat{\mathbf{P}}^{(0,i)}(n_{0,i})$	the ith branch path (of length $n_{0,i}$) from $\hat{\mathbf{P}}^{(0)}$
$\hat{\mathbf{P}}^{(0,i,j)}(n_{0,i,j})$	the jth branch path (of length $n_{0,i,j}$) from $\hat{\mathbf{P}}^{(0,i)}$
$\tilde{\mathbf{P}}^{(0)}(n_0)$	the initial path (of length n_0) in a two-valued tree-shaped hybrid SOO
$\tilde{\mathbf{P}}^{(0,i)}(n_{0,i})$	the ith branch path (of length $n_{0,i}$) from $\tilde{\mathbf{P}}^{(0)}$
$\tilde{\mathbf{P}}^{(0,i,j)}(n_{0,i,j})$	the jth branch path (of length $n_{0,i,j}$) from $\tilde{\mathbf{P}}^{(0,i)}$
$\mathbf{ROI}_k$	the range of influence of the kth cut set
s_i	the ith variable in a single-path two-valued SOO
$\tilde{\mathbf{S}}$	a two-valued single-path SOO on a feedback control loop
SN	small negative—a linguistic value of the process deviation
SP	small positive—a linguistic value of the process deviation
$\mathbf{T}$	a single-valued tree-shaped SOO

$\hat{\mathbf{T}}$	a two-valued tree-shaped SOO
$\tilde{\mathbf{T}}$	a two-valued tree-shaped hybrid SOO
te	preliminary occurrence index of top event
UCT_ℓ	uncertain with confidence of degree ℓ— a linguistic value of the diagnostic conclusion
V^{NOC}	the location of upper interior corner of the trapezoidal membership function NOC
V^{OCR}	the location of upper interior corner of trapezoidal membership function OCR
V_ℓ^{UCT}	the apex location of triangular membership function UCT_ℓ
V_i^{LP}	the location of upper interior corner of the trapezoidal membership function LP for d_i
V_i^{LN}	the location of upper interior corner of the trapezoidal membership function LN for d_i
V_i^{SP}	the apex location of triangular membership function SP for d_i
V_i^{SN}	the apex location of triangular membership function SN for d_i
V_i^{ZE}	the apex location of triangular membership function ZE for d_i
y_i	the ith process measurement
y_i^{ss}	the steady-state value of y_i
$\mathbf{Y}$	the set of all measured variables
ZE	zero—a linguistic value of process deviation

Greek letters

δ_i^t	the qualitative value representing the transient state of s_i
δ_i^f	the qualitative value representing the final state of s_i
$\mathbf{\Delta}$	the set of all deviation states, i.e., $\mathbf{\Delta} = \{-10, -1, 0, +1, +10\}$
μ_i^{ss}	the mean of y_i^{ss}
σ_i^{ss}	the standard deviation of y_i^{ss}

Abbreviations

FIS	fuzzy inference system
FPP	fault propagation path
ROI	range of influence
SDG	signed directed graph
SOO	symptom occurrence order

Appendix A. Proof of Theorem 2

Consider first the case when $n = 1$, i.e. $\tilde{\mathbf{S}}(1) = s_1(\delta_1^t, \delta_1^f)$. Since all three possible states of s_1, i.e. 0, δ_1^t and δ_1^f, may be observed on-line, these three candidate patterns can be described collectively with a logic statement, i.e.

$$s_1(0) \vee s_1(\delta_1^t) \vee s_1(\delta_1^f),$$

where the symbol $\vee$ denotes the logic operator "OR" and the number of terms connected by $\vee$ is the total number of candidate patterns. Since $N_{CP} = 2 \times 1 + 1 = 3$, we find Eq. (5) is satisfied in this case.

If $n = 2$, then $\tilde{\mathbf{S}}(2) = s_1(\delta_1^t, \delta_1^f) \rightarrow s_2(\delta_2^t, \delta_2^f)$. All possible candidate patterns in this case can be identified with the pattern generation rules. They can be expressed with the following statement:

$$[s_1(0) \wedge s_2(0)] \vee [s_1(\delta_1^t) \wedge s_2(0)] \vee [s_1(\delta_1^t) \wedge s_2(\delta_2^t)],$$

$$\vee [s_1(\delta_1^f) \wedge s_2(\delta_2^t)] \vee [s_1(\delta_1^f) \wedge s_2(\delta_2^f)],$$

where the symbol $\wedge$ denotes the logic operator "AND". Again, the number of terms connected by $\vee$ in this logic statement is the total number of candidate patterns. Notice that $N_{CP} = 2 \times 2 + 1 = 5$ and thus Eq. (5) is still valid.

Let us further assume that the theorem is valid for a particular case when $n = k$ $(k = 1, 2, 3, \ldots)$ and then consider if is also true when $n = k + 1$. If $n = k$, the corresponding $2k + 1$ candidate patterns can be divided into three types according to the state of final measured variable s_k. A more detailed description is provided below:

- Type I: If its state is δ_k^f, then there is only one possibility, i.e.

  $$s_1(\delta_1^f) \wedge s_2(\delta_2^f) \wedge \cdots \wedge s_k(\delta_k^f).$$

- Type II: If its state is δ_k^t, then there are k possible patterns, i.e.

  $$s_1(\delta_1^t) \wedge s_2(\delta_2^t) \wedge s_3(\delta_3^t) \wedge \cdots \wedge s_{k-1}(\delta_{k-1}^t) \wedge s_k(\delta_k^t),$$

  $$s_1(\delta_1^f) \wedge s_2(\delta_2^t) \wedge s_3(\delta_3^t) \wedge \cdots \wedge s_{k-1}(\delta_{k-1}^t) \wedge s_k(\delta_k^t),$$

  $$s_1(\delta_1^f) \wedge s_2(\delta_2^f) \wedge s_3(\delta_3^t) \wedge \cdots \wedge s_{k-1}(\delta_{k-1}^t) \wedge s_k(\delta_k^t),$$

  $$\vdots$$

  $$s_1(\delta_1^f) \wedge s_2(\delta_2^f) \wedge s_3(\delta_3^f) \wedge \cdots \wedge s_{k-1}(\delta_{k-1}^f) \wedge s_k(\delta_k^t).$$

- Type III: If its state is 0, then another k possible patterns can be identified, i.e.

  $$s_1(0) \wedge s_2(0) \wedge s_3(0) \wedge \cdots \wedge s_{k-1}(0) \wedge s_k(0),$$

  $$s_1(\delta_1^t) \wedge s_2(0) \wedge s_3(0) \wedge \cdots \wedge s_{k-1}(0) \wedge s_k(0),$$

  $$s_1(\delta_1^t) \wedge s_2(\delta_2^t) \wedge s_3(0) \wedge \cdots \wedge s_{k-1}(0) \wedge s_k(0),$$

  $$\vdots$$

  $$s_1(\delta_1^t) \wedge s_2(\delta_2^t) \wedge s_3(\delta_3^t) \wedge \cdots \wedge s_{k-1}(\delta_{k-1}^t) \wedge s_k(0).$$

Next let us consider the case when $n = k + 1$. The corresponding SOO can be written as $\tilde{\mathbf{S}}(k+1) = \tilde{\mathbf{S}}(k) \rightarrow s_{k+1}(\delta_{k+1}^t, \delta_{k+1}^f)$. Thus all possible candidate patterns can be generated with the proposed pattern generation rules according to the patterns identified in the previous case when $n = k$. If the pattern associated with $\tilde{\mathbf{S}}(k)$ is type I, the state of s_{k+1} may be either δ_{k+1}^t or δ_{k+1}^f. Here, the possibility of

$s_{k+1}(0)$ is excluded on the basis of the second pattern generation rule. If, on the other hand, one of the last $k-1$ type-II patterns of $\tilde{\mathbf{S}}(k)$ is used for building the candidate patterns of $\tilde{\mathbf{S}}(k+1)$, then $s_{k+1}(\delta^t_{k+1})$ is the only allowed symptom. When every variable in $\tilde{\mathbf{S}}(k)$ is at the transient state, an additional possibility $s_{k+1}(0)$ should also be included. Finally, if the k type-III patterns of $\tilde{\mathbf{S}}(k)$ are considered for the same purpose, then $s_{k+1}(0)$ is the only choice for all of them.

The total number of candidate patterns can be determined by summing all possibilities mentioned above, i.e.

$$N_{CP} = 1 \times 2 + (k \times 1 + 1 \times 1) + k \times 1 = 2(k+1) + 1. \quad \square$$

Appendix B. Proof of Theorem 3

If $N_0 = 0$ then the symptom occurrence order is a single path, i.e. $\hat{\mathbf{T}} = \hat{\mathbf{P}}^{(0)}(n_0)$, and Eq. (6) can be written as

$$N_{CP} = \mathcal{N}_3\{\hat{\mathbf{P}}^{(0)}(n_0)\} = \frac{n_0(n_0+1)}{2} + n_0 + 1 = \frac{(n_0+1)(n_0+2)}{2}. \tag{B.1}$$

A formal proof of this equation is given below:

Let us first consider the case when $n_0 = 1$. Since only one measured variable s_1 is included in the path $\hat{\mathbf{P}}^{(0)}(n_0)$, all candidate patterns can be expressed as $s_1(0) \vee s_1(\delta^t_1) \vee s_1(\delta^f_1)$. Thus the number of candidate patterns is $N_{CP} = 3 = (1+1)(1+2)/2$. If $n_0 = 2$, then $\hat{\mathbf{P}}^{(0)}(2) = s_1(\delta^t_1, \delta^f_1) \to s_2(\delta^t_2, \delta^f_2)$. Since s_1 and s_2 are not on a negative feedback loop, the candidate patterns can be described with the following logic statement:

$$[s_1(0) \wedge s_2(0)] \vee [s_1(\delta^t_1) \wedge s_2(0)] \vee [s_1(\delta^t_1) \wedge s_2(\delta^t_2)] \times \vee [s_1(\delta^f_1) \wedge s_2(0)] \vee [s_1(\delta^f_1) \wedge s_2(\delta^t_2)], \times \vee [s_1(\delta^f_1) \wedge s_2(\delta^f_2)].$$

Thus, the total number of candidate patterns in the above statement is the same as that computed with Eq. (B.1), i.e. $N_{CP} = 6 = (2+1)(2+2)/2$. Notice that the fourth pattern is not allowed if s_1 and s_2 are loop variables.

Let us further assume that Eq. (B.1) is valid for a particular case when $n_0 = k$ $(k = 1, 2, 3, \ldots)$ and then consider if is also true when $n_0 = k+1$. If $n_0 = k$, the corresponding $(k+1)(k+2)/2$ candidate patterns can be classified into three types according to the state of final measurement s_k. By following the pattern generation rules, it can be easily deduced that:

- the symptom $s_k(0)$ is embedded in $k(k+1)/2$ patterns;
- the symptom $s_k(\delta^t_k)$ can be identified in another k patterns;
- only one pattern contains the symptom $s_k(\delta^f_k)$.

Next let us consider the case when $n_0 = k+1$. All possible candidate patterns can be generated on the basis of the patterns identified in the previous case when $n = k$. In particular, it can be determined by applying the pattern generation rules that the numbers of possible states of s_{k+1} for the three types of candidate patterns listed above should be 1, 2 and 3, respectively. The total number of candidate patterns in the present case can be determined accordingly, i.e.

$$N_{CP} = \frac{k(k+1)}{2} \times 1 + k \times 2 + 1 \times 3 = \frac{(k+2)(k+3)}{2}.$$

The validity of Eq. (B.1) is therefore established.

The second part of this proof is concerned with the general case when $N_0 > 1$, $N_{0,i_1} > 1 \exists i_1$, $N_{0,i_1,i_2} > 1 \exists i_2, \ldots$, etc. Notice that the classification of candidate patterns on each path in the tree-shaped SOO should be exactly the same as that given to a single path. Let us consider the last symptom in initial path $\hat{\mathbf{P}}^{(0)}(n_0)$. Specifically,

(1) If the symptom is $s_{n_0}(0)$, then none of the later symptoms in the tree-shaped SOO can be developed, i.e. their states should all be 0. The corresponding number of candidate patterns in this case should be $n_0(n_0+1)/2 \times 1 = n_0(n_0+1)/2$.

(2) If the symptom is $s_{n_0}(\delta^t_{n_0})$, then the later symptoms can only developed partially. In other words, the variables on the tree branches of SOO can only reach their respective transient, but not the final, states. Thus, by excluding the initial path, the remaining tree can be viewed as a single-valued SOO and Theorem 1 is applicable in this situation. Since the number of patterns associated with the initial path is n_0, the number of corresponding candidate patterns of the entire SOO should be

$$n_0 \prod_{i_1=1}^{N_0} \mathcal{N}_1\{\hat{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\}.$$

(3) If the symptom is $s_{n_0}(\delta^f_{n_0})$, then all later symptoms may be fully developed. In particular, any variable on a tree branch of SOO may reach either its transient or final state in a corresponding pattern. A two-valued counting operator $\mathcal{N}_3\{\bullet\}$ is thus required to compute the number of candidate patterns associated with the branch paths connecting to the initial path, i.e. $\hat{\mathbf{P}}(n_{0,i_1})$ and $i_1 = 1, 2, \ldots, N_0$. Since there is only one pattern associated with the initial path in this case, the number of corresponding patterns of the entire SOO can be determined according to

$$1 \times \prod_{i_1=1}^{N_0} \mathcal{N}_3\{\hat{\mathbf{P}}(n_{0,i_1})\}.$$

It obvious that Eq. (6) can be obtained by adding the above three numbers together. Finally, notice that each path $\hat{\mathbf{P}}^{0,i_1}(n_{0,i_1})$ $(i_1 = 1, 2, \ldots, N_0)$ can be viewed as the initial path of the branch paths connecting to its end. Thus, the counting operator $\mathcal{N}_3\{\hat{\mathbf{P}}(n_{0,i_1})\}$ can be evaluated recursively with Eq. (7).

Appendix C. Proof of Theorem 4

Since $\tilde{\mathbf{P}}^{(0)}(n_0)$ is a loop path, Theorem 2 should be applicable in this case. Thus, the number of corresponding patterns is $2n_0+1$ and these patterns can also be classified into three types according to the last symptom on this path. In particular, there are n_0 type I patterns (i.e. the last measurement value is 0), n_0 type II patterns (i.e. the last measurement is at the transient state), and only one type III pattern (i.e. the last measurement is at the final state). Let us consider the symptoms on the branch paths connecting to the end of this initial path $\tilde{\mathbf{P}}^{(0)}(n_0)$:

(1) If the pattern associated with initial path is of type I, then none of the later symptoms in the hybrid SOO can be developed, i.e. their states should all be 0. Therefore, the number of corresponding candidate patterns for the entire SOO should be $n_0 \times 1 = n_0$.

(2) If the pattern associated with initial path is of type II, then two different situations should be considered:

- If all measurements on the initial path are at the transient states, then the measurements on all branch paths of SOO can only reach their respective transient, but not the final, states. Thus, Theorem 1 can be applied to count the candidate patterns associated with all branch paths. Since there is only one possible pattern on the initial path, the number of corresponding candidate patterns of the entire SOO should be

$$1 \times \prod_{i_1=1}^{N_0} \mathcal{N}_1\{\hat{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\}.$$

- If the first measurement on initial path is at the final state, then the patterns on the loop path $\tilde{\mathbf{P}}^{(0,1)}$ and the other branch paths $\tilde{\mathbf{P}}^{(0,i_1)}$ $(i_1=2,3,\ldots,N_0)$ should not be the same. In the former case, let us temporarily assume that the number of the corresponding patterns can be computed with an operator $\mathcal{N}_{4,(1)}\{\bullet\}$. On the other hand, it should be noted that Theorem 1 is still applicable in the latter case for counting the patterns associated with the branch paths not on the loop., i.e. $\tilde{\mathbf{P}}^{(0,i_1)}$ $(i_1 = 2,3,\ldots,N_0)$. Since the number of patterns on initial path is n_0-1 in this case, the number of corresponding patterns of the entire SOO should be computed according to the following formula:

$$(n_0-1) \times \left[\mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1)}\} \cdot \times \prod_{i_1=2}^{N_0} \mathcal{N}_1\{\tilde{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\} \right].$$

(3) If the pattern associated with initial path is of type III, the patterns on the loop path $\tilde{\mathbf{P}}^{(0,1)}$ and the other branch paths $\tilde{\mathbf{P}}^{(0,i_1)}$ $(i_1=2,3,\ldots,N_0)$ should be obtained with different pattern generation rules. Let us assume that the former patterns can be counted with an operator $\mathcal{N}_{4,(2)}\{\bullet\}$. On the other hand, since the measurements may reach their final states on $\tilde{\mathbf{P}}^{(0,i_1)}$ $(i_1=2,3,\ldots,N_0)$, Theorem 3 should be applicable in those cases. Thus, the number of corresponding patterns of the entire SOO should be

$$1 \times \left[\mathcal{N}_{4,(2)}\{\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})\} \cdot \prod_{i_1=2}^{N_0} \mathcal{N}_3\{\tilde{\mathbf{P}}^{(0,i_1)}(n_{0,i_1})\} \right].$$

Notice that Eq. (9) can be obtained by summing the above numbers. In order to use this formula, the operators $\mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})\}$ and $\mathcal{N}_{4,(2)}\{\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})\}$ must be evaluated in advance.

The first operator should be evaluated under the conditions that the patterns on the initial path is of type II and the first measurement is at its final state. Consequently, only one pattern is possible on $\tilde{\mathbf{P}}^{(0,1)}$, i.e. all measurements on this path should be at their transient states. This is the result of applying the pattern generation rules used in Theorem 2. However, there is still a need to account for the possible patterns on the branch paths connecting to the end of $\tilde{\mathbf{P}}^{(0,1)}$. Although there is still only one possibility associated with $\tilde{\mathbf{P}}^{(0,1,1)}$, the patterns on the other paths, i.e. $\tilde{\mathbf{P}}^{(0,1,i_2)}$ and $i_2 = 2,3,\ldots,N_{0,1}$, should be determined with Theorem 1. If there are further branch paths connecting to the end of $\tilde{\mathbf{P}}^{(0,1,1)}$, then essentially the same computation steps can be adopted to determine the number of corresponding candidate patterns. This recursive procedure is described concisely in Eq. (10). The computation procedure is terminated when the further connecting branch paths cannot be found at the end of a loop path, i.e. $N_{0,1,\ldots,1}=0$. Then the operator value in this situation should be determined with Eq. (12) since there is only one possible patterns on this loop path.

In order to evaluated the second operator, it is necessary to examine the candidate patterns on $\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})$ under the constraint that the patterns on the initial path is of type III. It should be noted that only types II and III patterns are possible on this branch path. This is due to the proposed pattern generation rules used for loop paths. Let us consider these two cases separately:

- If the pattern on $\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})$ is of type II, then all measurements on the further loop path $\tilde{\mathbf{P}}^{(0,1,1)}(n_{0,1,1})$ must be at their transient states. This is due to the pattern generation rules used in Theorem 2. Consequently, the operator $\mathcal{N}_{4,(1)}\{\bullet\}$ should be applicable for determining the corresponding patterns, i.e

$$n_{0,1}\mathcal{N}_{4,(1)}\{\tilde{\mathbf{P}}^{(0,1,1)}(n_{0,1,1})\}\prod_{i_2=2}^{N_{0,1}} \mathcal{N}_1\{\tilde{\mathbf{P}}^{(0,1,i_2)}(n_{0,1,i_2})\}.$$

- If the pattern on $\tilde{\mathbf{P}}^{(0,1)}(n_{0,1})$ is of type III, then the rationale for generating the candidate patterns should be identical to that for the corresponding case when the pat-

tern on $\tilde{\mathbf{P}}^{(0)}(n_0)$ is of type III. Thus, operator $\mathcal{N}_{4,(2)}\{\bullet\}$ should be applied again in a formula of the same format to compute the pattern number, i.e.

$$\mathcal{N}_{4,(2)}\{\tilde{\mathbf{P}}^{(0,1,1)}(n_{0,1,1})\}\prod_{i_2=2}^{N_{0,1}}\mathcal{N}_3\{\tilde{\mathbf{P}}^{(0,1,i_2)}(n_{0,1,i_2})\}.$$

Notice that Eq. (11) can be obtained by adding the above two numbers together and obviously this equation can be applied repeatedly if there are further branch paths connecting to the end of $\tilde{\mathbf{P}}^{(0,1,1)}(n_{0,1,1})$. This recursive process is terminated when $N_{0,1,\ldots,1}=0$. The operator value in this situation should be computed with Eq. (13) since, according to the pattern generation rules for loop paths, the measurements on $\tilde{\mathbf{P}}^{(0,1,\ldots,1)}(n_{0,1,\ldots,1})$ should have already reached their transient or final states. In other words, the possibilities of normal measurements appearing in the corresponding patterns can be ruled out completely.

References

Chang, C. T., & Chen, J. W. (1995). Implementation issues concerning the EKF-based fault diagnosis technique. *Chemical Engineering Science*, *50*, 2861.

Chang, C. T., & Hwang, H. C. (1992). New developments of the digraph-based techniques for fault-tree synthesis. *I & EC Research*, *31*, 1490.

Chang, S. Y., Lin, C. R., & Chang, C. T. (2002). A fuzzy diagnosis approach using dynamic fault trees. *Chemical Engineering Science*, *57*, 2971.

George, J. K., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic, theory and application*. Englewood Cliffs, NJ: Prentice-Hall.

Hoskins, J. C., Kalivur, K. M., & Himmeblau, D. M. (1991). Fault diagnosis in complex chemical-plants using artificial neural networks. *A.I.Ch.E. Journal*, *37*, 137.

Iri, M., Aoki, K., O'Shima, E., & Matsuyama, H. (1979). An algorithm for diagnosis of system failure in the chemical process. *Computers Chemical Engineering*, *3*, 489.

Lapp, S. A., & Powers, G. J. (1977). Computer-aided synthesis of fault trees. *IEEE Transactions on Reliability*, *R-26*, 2.

Mathworks (2000a). *Fuzzy logic toolbox-user guide*. Natick, MA: The Mathworks Inc.

Mathworks (2000b). *SIMULINK-dynamic system simulation for MATLAB-using simulink*. Natick, MA: The Mathworks Inc.

Petti, T. F., Klein, J., & Dhurjati, P. S. (1990). Diagnostic model processor: Using deep knowledge for process fault diagnosis. *A.I.Ch.E. Journal*, *36*, 565.

Ross, T. (1995). *Fuzzy logic with engineering applications*. New York: Mc-Graw Hill.

Ulerich, N. H., & Powers, G. J. (1988). On-line hazard aversion and fault diagnosis in chemical processes: The digraph + fault tree method. *IEEE Transactions on Reliability*, *37*, 171.