Systematic Development of Automata Generated Languages for Fault Diagnosis in Continuous Chemical Processes

Chuei-Tin Chang. Jung-Yang Chen

National Cheng Kung University, Tainan, Taiwan 70101,
ROC (Tel:886-6-275-7575 ext. 62663; e-mail: ctchang@mail.ncku.edu.tw).

Abstract: A SDG-based simulation procedure is presented in this study to qualitatively predict all possible effects of one or more fault propagating in a given process system. All possible state evolution behaviors are characterized with an automaton model. By selecting a set of on-line sensors, the corresponding diagnoser can be constructed and the diagnosability of every fault origin can be determined accordingly. Furthermore, it is also possible to construct a formal diagnostic language on the basis of this diagnoser. Every string (word) in the language is then encoded into an IF-THEN rule and, consequently, a comprehensive fuzzy inference system can be synthesized for on-line diagnosis. The feasibility of this approach is demonstrated with a simple example in this paper.

Keywords: fault diagnosis, automata, signed directed graph, formal language, fuzzy logic.

1. INTRODUCTION

The fault diagnosis methods have been widely recognized as indispensable tools for enhancing process safety. Generally speaking, they could be classified into three distinct groups, i.e., the model based approaches, the knowledge based approaches, and the data-analysis based approaches (Venkatasubramanian et al., 2003a, b). However, in order to carry out these strategies on-line, it is usually necessary to first analyze the historical data and/or operational experiences obtained during *every* serious accident. This requirement cannot always be satisfied in practice.

To circumvent the above drawbacks, a qualitative cause-and-effect model, i.e., the signed directed graph (SDG), is used in the present study to characterize fault propagation mechanisms. The advantage of this modelling approach is mainly due to the fact that the causal relations in process systems can always be established according to generic engineering principles without any quantitative knowledge. On the other hand, it should be noted that such causal models are basically static in nature. Many SDG-based fault identification techniques were therefore implemented on the basis of the *steady-state* symptoms only, e.g., Maurya et al. (2006). Since the effects of fault(s) and/or failure(s) usually propagate throughout the entire system dynamically in sequence, a series of intermediate events may occur before the inception of catastrophic consequences. Thus, the performance of a qualitative diagnosis scheme should be evaluated not only in terms of its correctness but also its timeliness.

To enhance diagnostic efficiency, it is obviously necessary to consider the precedence order (in time) of various fault propagation effects derived from the qualitative models. Extensive studies have already been carried out to develop effective diagnosis strategies by incorporating both the eventual symptoms and also their *occurrence order* into a fuzzy inference system (FIS). This approach has been applied successfully to a number of loop-free processes (Chang et al., 2002) and also to systems with feedback and/or feed forward control loops (Chang and Chang, 2003; Chen and Chang, 2006; 2007).

Despite the fact that diagnostic performance can be significantly improved with the aforementioned technique, the representation, analysis and synthesis of inference systems are still very cumbersome. In particular, many different versions of the symptom occurrence orders can often be deduced from a single fault origin on the basis of SDG model. Manual enumeration of all such scenarios for all origins may become intractable even for a moderately complex system. Furthermore, the diagnosability issues concerning the resulting FIS have never been systematically addressed in the past. Thus, there is a definite need to develop a unified theoretical framework to extract the intrinsic features of dynamic fault propagation mechanisms. Our concern here is primarily with the sequence of system states visited after the occurrence of fault origin(s) and also the associated events causing the state transitions. A systematic procedure is proposed in this paper to construct automata and language models for the purpose of representing these sequences accurately and succinctly. As a result, additional insights can be revealed and, also, more compact inference rules can be produced accordingly. A simple example is provided at the end of this paper to demonstrate the feasibility and effectiveness of the proposed procedures for FIS synthesis and for fault diagnosis.

2. AUTOMATA CONSTRUCTON

2.1 *Qualitative Simulation Procedure*

Although other qualitative models may be equally acceptable, the SDG is adopted in the present study to simulate (or predict) the effects of faults and failures. This is due to the fact that the needed implementation procedure is conceptually straightforward. Notice first that the fault origins can usually be associated with the primal nodes, i.e., the nodes without inputs. A set of five values, i.e., {-10, -1, 0, +1, +10}, may be assigned to every node in the digraph to represent deviation from the normal value of corresponding variable. The value 0 represents the normal steady state. The negative values are used to denote the lower-than-normal states and the positive values signify the opposite. The magnitudes of non-zero deviations, i.e., 1 or 10, can be interpreted qualitatively as "small" and "large" respectively. The causal relation between two variables can be characterized with a directed arc and the corresponding gain. Each gain may also assume one of the five qualitative values mentioned above. The output value of every arc in digraph can be computed with the gain and its input value according to the following equation:

$$v_{out} = \begin{cases} g \times v_{in} & \text{if } -10 \le g \times v_{in} \le +10 \\ +10 & \text{if } g \times v_{in} > +10 \\ -10 & \text{if } g \times v_{in} < -10 \end{cases} \tag{1}$$

where g, v_{in} and v_{out} denote respectively the gain, input and output values. It is obvious that the deviation values of all variables affected by one or more fault origin can always be computed with this formula, 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*. In essence, this is the most basic assumption adopted in this study. Notice that, if the precedence order of various fault propagation effects is to be considered in fault diagnosis, a large number of different versions of qualitative simulation results may be generated accordingly. All such scenarios can be captured with the automaton model described in the sequel.

2.2 *System Automata*

A formal definition of a deterministic automaton $\mathcal{A}$ can be found in Cassandras and Lafortune (1999). Specifically, it is a six-tuple

$$\mathcal{A} = \left(\mathbb{X}, \mathbb{E}, f, \Gamma, x_0, \mathbb{X}_m\right) \tag{2}$$

where, $\mathbb{X}$ is the set of system states; $\mathbb{E}$ is the finite set of events associated with the transitions in automaton; $f : \mathbb{X} \times \mathbb{E} \to \mathbb{X}$ is the transition function; $\Gamma : \mathbb{X} \to 2^{\mathbb{E}}$ is the active event function; x_0 is the initial system state; $\mathbb{X}_m \subseteq \mathbb{X}$ is the set of marked states. In the present application, each system state $x \in \mathbb{X}$ is either a collection of node values at a particular instance after an initiating failure occurs or the initial state itself. Every event $e \in \mathbb{E}$ represents a previously nonexistent fault effect. Notice that the precedence order of these events must be consistent with the basic assumption mentioned above. The active event function $\Gamma(x)$ is used to specify the events which could change the system state x, while the transition function $f(x,e)$ is used for stipulating the resulting state caused by $e \in \Gamma(x)$. Finally, it should be noted that the initial state x_0 in this study is always associated with the *normal* condition and the set $\mathbb{X}_m$ contains the final steady states reached in all possible fault propagation scenarios.

To facilitate illustration of the automaton construction steps, let us consider the most fundamental digraph configuration, i.e., tree. More specifically, let us use the fictitious SDG model in Figure 1 as an example and also assume that a positive deviation in the upstream variable d, i.e., $d(+1)$, is the only possible fault origin in this case. Notice that, although the precedence order of any two effects along the same branch path in this digraph can be uniquely identified with the proposed qualitative simulation procedure, the order of two distinct events located on *separate* branches should be considered as indeterminable. The corresponding automaton can thus be described with the state transition diagram presented in Figure 2. Every system state here is characterized with a collection of the qualitative values of all variables in the digraph and all of them are listed in Table 1. Three equally possible event sequences between the initial and final system states can be identified from this automaton model, i.e.,

1. $d(+1)x(+1)y(+1)z(-1)u(+1)$,
2. $d(+1)x(+1)y(+1)u(+1)z(-1)$,
3. $d(+1)x(+1)u(+1)\,y(+1)\,z(-1)$.

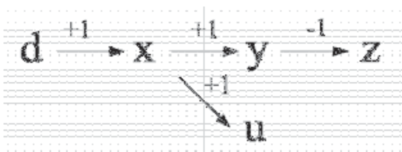

Fig. 1. A tree-shaped SDG model.

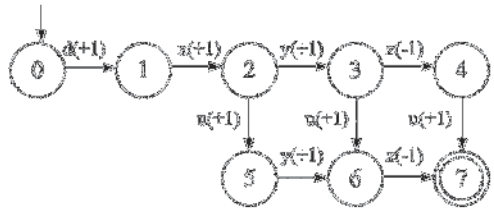

Fig. 2. The state transition diagram of automaton derived from Figure 1.

The automaton resulting from a "large" disturbance can be obtained by following a similar procedure. An auxiliary assumption is introduced in this work to facilitate an accurate description of the fault propagation mechanism, i.e., *the smaller deviation of a process variable must occur before* *reaching a larger one of the same variable*. Thus, the automaton in Figure 2 can be revised to incorporate this requirement (see Figure 3).

Table 1. System states in Figure 2.

State	d	x	y	z	u
0	0	0	0	0	0
1	+1	0	0	0	0
2	+1	+1	0	0	0
3	+1	+1	+1	0	0
4	+1	+1	+1	-1	0
5	+1	+1	0	0	+1
6	+1	+1	+1	0	+1
7	+1	+1	+1	-1	+1

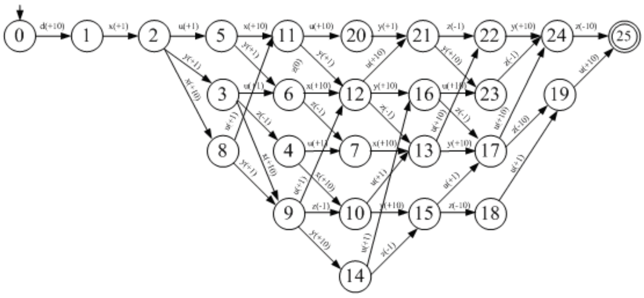

Fig. 3. The automaton resulted from $d(+10)$ in Figure 1

2.3 Diagnoser and Diagnisability

In realistic applications, the fault origins (i.e., failures or upsets) and some of the process variables cannot be monitored on-line. Thus, the event set of an automaton model can be further divided into the observable and unobservable event subsets, i.e., $\mathbb{E} = \mathbb{E}_o \cup \mathbb{E}_{uo}$. To check diagnosability of each fault origin and also facilitate diagnostic inference with the available sensors, the system automaton $\mathcal{A}$ should be converted to a *diagnoser* $\mathcal{A}_{diag}$, which is in essence a transformed automaton with the observable subset $\mathbb{E}_o$ as its event set. Although a systematic construction procedure has already been developed by Sampath et al. (1996) for the discrete event systems in general, the diagnosers for the present applications are built with an intuitive but more convenient alternative approach. Specifically, if a state is reached immediately after an unobservable event, then this state is merged with its predecessor(s) in the original automaton model. For example, let us assume that $d(+1)$ is the fault origin and $y(+1)$ is not observable in Figure 2. The corresponding diagnoser can be easily obtained by applying the aforementioned principle (see Figure 4). The numerical node labels here are the same as those in Figure 2, while the subscript of each label is used to reflect whether or not the fault origin has occurred at the corresponding state.

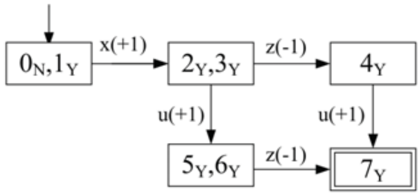

Fig. 4. The diagnoser obtained by assuming $y(+1)$ in Figure 2 is unobservable.

It should be noted that this construction method is applicable even when multiple scenarios are possible. For example, let us consider the SDG model in Figure 5 and assume that there are two measured variables, i.e., y and z , and four potential fault origins, i.e., (1) $d_x(+1)$, (2) $d_y(+1)$, (3) $d_z(+1)$ and (4) $d_u(+1)$.

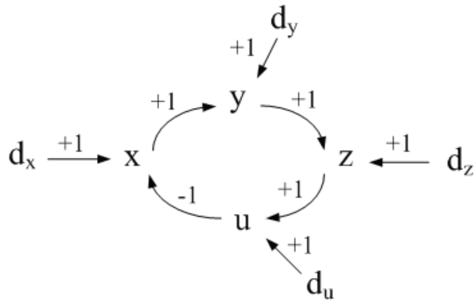

Fig. 5. A SDG model with negative feedback loop

The automaton model of this system and the corresponding diagnoser can be found in Figures 6(A) and 6(B) respectively. Obviously, the issue of diagnosability becomes important in this situation. Although the formal necessary and sufficient conditions of system diagnosability has also been derived and proven rigorously by Sampath et al. (1995), the identifiability of each fault origin in our studies can be determined simply by inspecting the diagnoser. In particular, the diagnosability of a fault origin can be established if it is the unique cause of at least one diagnoser state. Otherwise, the corresponding on-line symptoms should be indistinguishable from those of one or more scenarios caused by other origins. It can be determined on the basis of this criterion that fault origins (3) and (4) are both diagnosable, while the observable event sequences in scenarios (1) and (2) are identical and thus cannot be differentiated from one another.

The feasibility of this simple checking procedure is attributed mainly to the fact that the automata used in the present applications form a special subclass of those for modelling the discrete event systems. More specifically, since the *continuous* chemical processes are considered in this work, the corresponding automata can be characterized with the following unique features:

1. The initial automaton state is always associated with the normal system condition.
2. Every initial state transition is triggered by

failure event(s).

3. Recurrence of system state is not possible, i.e., the automaton is free of any feedback loop. Notice that this feature is due to our assumption that a final steady state is reachable in every possible scenario.

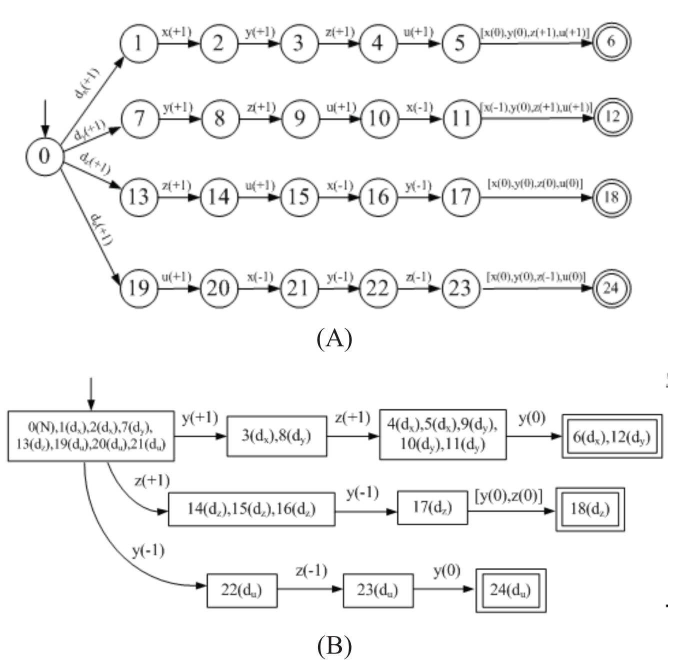

Fig 6. The automaton (A) and diagnoser (B) constructed according to the SDG in Figure 5

3. LANUGAE GENERATION

A *language* $\mathfrak{L}$ is regarded in this work as a collection of finite-length event sequences. These sequences are referred to as *strings* or *words*. The set of all possible events (alphabets) is the set $\mathbb{E}$ defined in equation (2). An additional set $\mathbb{E}^*$ is also utilized here to include all possible strings (including the empty string ε) constructed over $\mathbb{E}$. Thus, it is obvious that $\mathfrak{L} \subseteq \mathbb{E}^*$.

Since fault diagnosis can only be performed according to the on-line symptoms, the automaton $\mathcal{A}_{diag}$ (not $\mathcal{A}$) is used to generate a diagnostic language for the purpose of enumerating all *observable* event sequences caused by a given fault origin. Specifically,

$$\mathfrak{L}\left(\mathcal{A}_{diag}\right) = \left\{ s \in \mathbb{E}^* \mid f\left(x_0, s\right) \text{ is defined by } \mathcal{A}_{diag} \right\} \quad (3)$$

The transition function $f\left(x_0, s\right)$ here can be evaluated recursively according to the following rules:

$$\begin{aligned} f\left(x, \varepsilon\right) &= x \\ f\left(x, se\right) &= f\left(f\left(x, s\right), e\right) \end{aligned} \quad (4)$$

where, $s \in \mathbb{E}^*$ and $e \in \mathbb{E}$. In addition, the *marked* language of automaton $\mathcal{A}_{diag}$ can be defined as

$$\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}\right) = \left\{ s \in \mathfrak{L}\left(\mathcal{A}_{diag}\right) \mid f\left(x_0, s\right) \in \mathbb{X}_m \right\} \quad (5)$$

Notice that an automaton-based language can be synthesized by first identifying the longest strings and then obtaining all their *prefixes*. Since the marked states in the present application are always terminal, $\mathfrak{L}\left(\mathcal{A}_{diag}\right)$ can be produced by taking the prefix closure of $\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}\right)$ (Cassandras and Lafortune, 1999), i.e.

$$\mathfrak{L}\left(\mathcal{A}_{diag}\right) = \overline{\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}\right)} \quad (6)$$

where, $\overline{\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}\right)}$ denotes the set of all prefixes of the strings in $\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}\right)$. From equation (6), it can be shown that every diagnoser considered in this study must be *nonblocking*, i.e., any string $s \in \mathfrak{L}\left(A_{diag}\right)$ can be always extended by another string t such that $st \in \mathfrak{L}_{\mathrm{m}}\left(A_{diag}\right)$.

Let us use the diagnoser in Figure 4 as an example to illustrate the proposed approach. The two languages marked and generated respectively by $\mathcal{A}_{diag}$ in this case should be

$$\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}\right) = \left\{ x(+1)z(-1)u(+1),\ x(+1)u(+1)z(-1) \right\} \quad (7)$$

$$\mathfrak{L}\left(\mathcal{A}_{diag}\right) = \left\{ \begin{matrix} \varepsilon, x(+1),\ x(+1)z(-1),\ x(+1)u(+1), \\ x(+1)z(-1)u(+1),\ x(+1)u(+1)z(-1) \end{matrix} \right\} \quad (8)$$

If the possibilities of multiple fault origins are incorporated in a diagnoser, then it is necessary to further generate a sublanguage specific to every fault origin, i.e.

$$\mathfrak{L}\left(\mathcal{A}_{diag}\right) = \bigcup_i \mathfrak{L}\left(\mathcal{A}_{diag}^{F_i}\right) \quad (9)$$

where, $\mathcal{A}_{diag}^{F_i}$ is an automaton obtained by removing all the abnormal states in A_{diag} which are *not* caused by the i th fault origin F_i. The marked sublanguages of the fault origins in Figure 6(B) can be easily produced with this method, i.e.,

$$\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{d_x(+1)}\right) = \mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{d_y(+1)}\right) = \left\{ y(+1)z(+1)y(0) \right\} \quad (10)$$

$$\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{d_z(+1)}\right) = \left\{ z(+1)y(-1)\left[y(0), z(0)\right] \right\} \quad (11)$$

$$\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{d_u(+1)}\right) = \left\{ y(-1)z(-1)y(0) \right\} \quad (12)$$

4. FUZZY INFERENCE SYSTEM

Every string in $\mathfrak{L}\left(\mathcal{A}_{diag}\right)$ is encoded with an IF-THEN rule in this work. These rules can be incorporated in a fuzzy inference system to evaluate the *existence potential* of the corresponding fault origin. In particular, if at least one event sequence in the marked sublanguage $\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{F_i}\right)$ can be confirmed, then it is highly possible that they are caused by the corresponding fault origin F_i. To assert such a belief, the fuzzy conclusion "cs_i is OCR" is adopted in the inference rule, where OCR is the linguistic value of the *occurrence index* cs_i reflecting the highest confidence level in confirming the existence of F_i. More specifically, this rule can be written as

$$\text{IF } s_o \in \mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{F_i}\right) \text{ THEN } cs_i = \text{OCR}$$

where s_o denotes the observed event string.

On the other hand, it is certainly reasonable to disregard the possibility of a fault if none of the corresponding event strings in $\mathfrak{L}\left(\mathcal{A}_{diag}^{F_i}\right)$ can be observed. Thus, the diagnosis for this scenario should be "cs_i is NOC", where NOC is the linguistic value representing the lowest level of confidence. In other words,

$$\text{IF } s_o \notin \mathfrak{L}\left(\mathcal{A}_{diag}^{F_i}\right) \text{ THEN } cs_i = \text{NOC}$$

The diagnostic conclusion for each of the remaining strings should be UCT_ℓ, i.e., uncertain with confidence level ℓ. In particular, this rule can be written as

$$\text{IF } s_o \in \mathfrak{L}\left(\mathcal{A}_{diag}^{F_i}\right) \setminus \mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{F_i}\right) \text{ THEN } cs_i = \mathrm{UCT}_\ell$$

In this study, the confidence level ℓ in confirming the existence of the root cause(s) is assumed to be proportional to the string length. The highest possible confidence level is of course assigned to the strings in $\mathfrak{L}_{\mathrm{m}}\left(\mathcal{A}_{diag}^{F_i}\right)$.

Finally, it should be noted that the aforementioned IF-THEN rules can be implemented with the two-layer fuzzy inference framework developed by Chen and Chang (2006).

5. CASE STUDY

Let us consider the level control system presented in Figure 7 and the corresponding SDG model in Figure 8. All on-line signals, i.e., s_5 - s_8, are assumed to be available for fault diagnosis in this example. For illustration convenience, only two possible scenarios are considered here, i.e., (1) a moderate (controllable) increase in the flow rate of stream 3 while control valve CV-01 sticks and (2) an uncontrollable increase in the flow rate of stream 3.

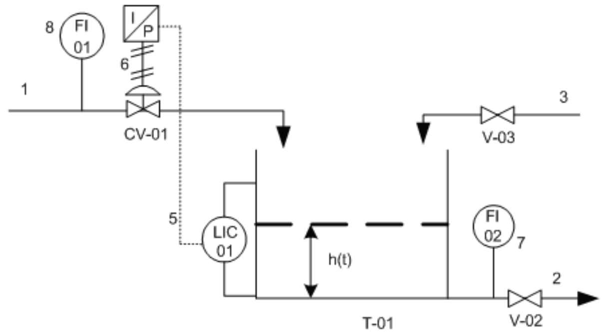

Fig. 7. A level-control system.

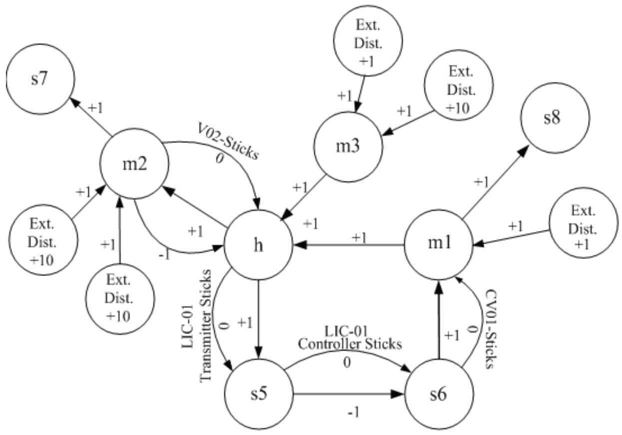

Fig. 8. The SDG model of level-control system.

The diagnoser for these two fault origins can be found in Figure 9. Notice that this automaton is presented in two parts for clarity. States 0 and $0'$ are used to represent the combined states of the normal condition and the system conditions reached immediately after the occurrence of fault origin in scenario 1 and scenario 2 respectively. These two states, i.e., 0 and $0'$, should be lumped into a single one in the actual diagnoser.

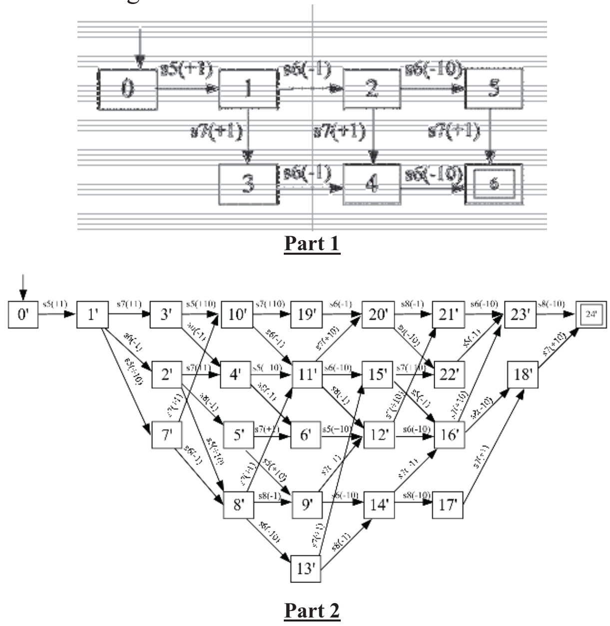

Fig. 9. The diagnoser used for level-control system

To verify the effectiveness of the proposed fault diagnosis approach, extensive numerical simulation studies have been carried out in this work. The on-line measurement data of all fault propagation scenarios were generated with SIMULINK. These data were then used in Sugeno's inference procedure with the fuzzy-logic module of MATLAB toolbox. As an example, let us first examine the occurrence index of the event $m3(+10)$ in scenario 2. It can be observed from Figure 10(A) that the diagnosis is clearly swift and quite accurate. Specifically, the existence of fault origin is detected almost immediately and fully confirmed at about 500 second after its introduction. On the other hand, the occurrence index of the incorrectly assumed fault origin in scenario 2, i.e., $m3(+10)$, is presented in Figure 10(B). Notice that the nonzero occurrence index in the period between 1000 and 2600 sec can be attributed to the fact that the observed event strings caused by the two fault origins can be matched partially during the *initial stage*. More specifically, the set of matched strings is

$$\left\{\overline{s5(+1)s6(-1)s7(+1)}, \overline{s5(+1)s7(+1)s6(-1)}\right\}$$

As the on-line symptoms developed further, none of the longer strings generated by the first part of automaton in Figure 9 can be used to characterize the measurement data obtained after 2600 sec and thus the occurrence possibility of the second fault origin was rejected with the proposed inference mechanisms (Chen and Chang, 2006)..

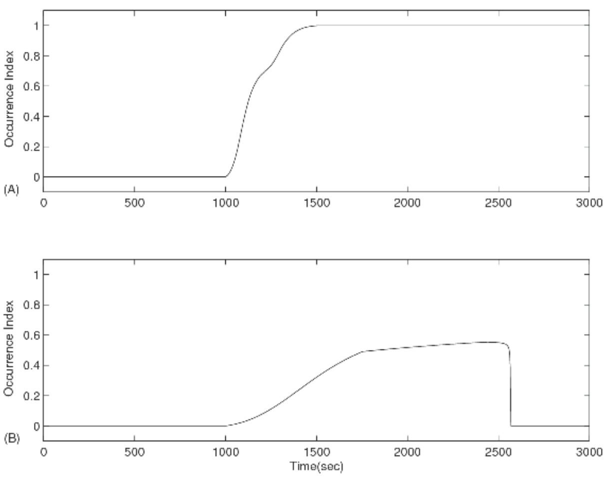

Fig. 10. Diagnosis results of two different scenarios in the level control system. (A) Occurrence index of the second fault origin using simulation data obtained by introducing the same event; (B) Occurrence index of the second fault origin using simulation data obtained by introducing the basic events in the first scenario.

6. CONCLUSIONS

In this study, a SDG-based reasoning procedure is proposed to qualitatively predict all possible symptom patterns and also their progression sequences. These intrinsic features of symptom evolution patterns are captured with automata and language models. The resulting IF-THEN rules can be incorporated in a fuzzy inference system and this system can be installed on-line to identify not only the locations of fault origins but also their magnitude levels with relatively high resolution.

REFERENCES

Cassandras, C.G., and Lafortune S. (1999). *Introduction to Discrete Event Systems*, Kluwer Academic Publisher. Boston.

Chang, S.Y., and Chang, C.T. (2003). A fuzzy-logic based fault diagnosis strategy for process control loops. *Chem. Eng. Sci.,* 58, 3395.

Chang, S.Y., Lin, C.R., and Chang, C.T. (2002). A fuzzy diagnosis approach using dynamic fault trees. *Chem. Eng. Sci.,* 57, 2971.

Chen, J.Y., and Chang, C.T. (2006). Fuzzy diagnosis method for control systems with coupled feed forward and feedback loops. *Chem. Eng. Sci.,* 61, 3105.

Chen, J.Y., and Chang, C.T. (2007). Systematic enumeration of fuzzy diagnosis rules for identifying multiple faults in chemical processes. *Ind. Eng. Chem. Res.,* 46, 3635.

Ju, S.N., Chen, C.L., and Chang, C.T. (2004). Constructing fault trees for advanced process control systems - application to cascade control loops. *IEEE Trans on Reliab*., 53, 43.

Maurya, M.R., Rengaswamy, R., and Venkatasubramanian, V. (2004). Application of signed digraph-based analysis for fault diagnosis chemical process flowsheets. *Engineering Applications of Artificial Intelligence.,* 17, 501.

Maurya, M.R., Rengaswamy, R., and Venkatasubramanian, V. (2006). A signed directed graph-based systematic framework for steady state malfunction diagnosis control loops. *Chem. Eng. Sci.*, 61, 1790.

Oyeleye, O.O., and Kramer, M.A. (1988). Qualitative simulation of chemical process systems: steady-state analysis. *AIChE Journal.*, 34, 1441.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1996). Failure Diagnosis using discrete-event models. *IEEE Trans. Cont. Syst. Tech.,* 4, 105.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Trans. Auto. Cont.*, 40, 1555.

Venkatasubramanian. V., Rengaswamy, R., Yin, K., and Kavuri, S. (2003a). A review of process fault detection and diagnosis, part 1: quantitative model based methods. *Comput & Chem Engng.*, 27, 293.

Venkatasubramanian, V., Rengaswamy, R., and Kavuri, S. (2003b). A review of process fault detection and diagnosis, part 2: qualitative model and search strategies. *Comput. & Chem. Engng.*, 27, 313.