Contents lists available at ScienceDirect

Chemical Engineering Research and Design

journal homepage: www.elsevier.com/locate/cherd

An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes

Ming-Li Yeh, Chuei-Tin Chang[*]

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

ABSTRACT

Online diagnosis has been considered as an important measure for improving operational safety in many batch chemical plants. Specifically, the state transition behaviors of all hardware items (components) in the given batch process and their failure mechanisms are modeled systematically with automata. The system model is then assembled by connecting the component models on the basis of a generic hierarchical structure. A "diagnoser" can be constructed accordingly for the purpose of determining various qualitative and quantitative performance indices. Guided by these indices, two performance enhancement approaches can be effectively applied: (1) installing additional sensors which are not included in the piping and instrumentation diagram (P&ID) and (2) executing extra operation steps which are not specified in the sequential function chart (SFC). Three examples are presented in this paper to demonstrate the feasibility of the proposed approach.

© 2011 The Institution of Chemical Engineers. Published by Elsevier B.V. All rights reserved.

Keywords: Automata; Batch operation; Diagnosability; Fault diagnosis; Fault propagation trace; Shannon entropy

1. Introduction

Hardware failures in chemical plants should be viewed as unavoidable but random events. They can usually be attributed to controller malfunctions, actuator failures (e.g., valve sticks, pump failures and compressor failures), containment failures (e.g., tank leakages), sensor failures, design errors and operator mistakes, etc. Any combination of these events may result in drastic decrease in productivity, significant deterioration in product quality and, in the worst cases, catastrophic outcomes such as explosions, fires, or toxic releases. Since offline hazard assessment can reduce the total expected loss of accidents only to a certain degree, *online* fault diagnosis should be regarded as an alternative means for improving the operational safety of chemical processes.

According to Venkatasubramanian et al. (2003a,b,c), the available fault diagnosis methods could be classified into three general types: (1) quantitative model-based approaches; (2) qualitative model-based approaches; (3) process history based approaches. These methods were developed primarily for the *continuous* chemical processes in the past, while significantly less effort has been devoted to the batch operations. Notice that online failure identification in the latter case is a much more difficult task. This is mainly due to the fact that a continuous process is supposed to be maintained at steady state, but the batch system state usually changes with time. Various different diagnostic strategies have already been adopted to cope with this time-variant nature and several of them are briefly reviewed in the sequel. Nomikos and MacGregor (1994, 1995) developed a multi-way principal component analysis method for batch process monitoring, which has later been extended for diagnosis applications (Kourti and Macgregor, 1995; Kourti et al., 1995; Lee et al., 2004; Undey et al., 2003; Chen and Jiang, 2011). In addition, fault identification techniques based on artificial neural networks, knowledge-based expert systems, observers and Petri nets have also been proposed for the batch operations (Ruiz et al., 2001a,b; Pierri et al., 2008; Hashizume et al., 2008; Caccavale et al., 2009). Although satisfactory results were reported in these studies, none of them addressed the important issue of performance assessment for the overall diagnostic system.

[*] *Corresponding author*. Tel.: +886 6 2757575x62663; fax: +886 6 2344496.
E-mail address: ctchang@mail.ncku.edu.tw (C.-T. Chang).

Received 8 February 2011; Received in revised form 24 April 2011; Accepted 3 May 2011

0263-8762/$ – see front matter © 2011 The Institution of Chemical Engineers. Published by Elsevier B.V. All rights reserved.

doi:10.1016/j.cherd.2011.05.007

It can also be observed from the above-mentioned studies for both continuous and batch systems that, in order to facilitate effective online fault identification, the fault propagation mechanisms must be clearly described with a quantitative or qualitative model and the resulting symptom evolution patterns must also be adequately characterized or *predicted* in advance. Although the digraph model is by far the most popular choice for this purpose (Maurya et al., 2004; Zhang et al., 2005; Chang and Chen, 2007; Chen and Chang, 2009), it has been used mostly in applications concerning the continuous processes. This is because digraph is not suitable for representing the dynamic causal relationships among time, events, equipment states and system configurations in the semi-batch or batch processes.

To circumvent the above-mentioned drawbacks, Viswanathan et al. (2002) adopted a hybrid framework to incorporate two different types of popular models, i.e., Petri nets and digraphs, for offline hazard assessment. Chen et al. (2010) also developed several Petri-net based algorithms in a recent study to configure online fault diagnosis systems for the batch processes. Although satisfactory results in simple examples were reported in this study, there are still critical issues that must be addressed before actual implementation:

- Since the event sequences (or traces) in multi-failure scenarios cannot be efficiently enumerated with the Petri-net models, the scope of fault diagnosis was mostly limited to the single-failure accidents only.
- A single intuitive index, i.e., the percentage of diagnosable traces, was adopted to evaluate the diagnostic performance of every given batch operation. This somewhat arbitrary approach is not enough for fully characterizing various complex features of the diagnosis results.
- The Petri-net based procedure was not tested rigorously with realistic batch operations in practical applications.

These deficiencies are eliminated in the present work with automata. The automaton model was originally used by Sampath et al. (1995, 1996) and also in a series of subsequent studies as the basis for online diagnosis in discrete-event systems (Zad et al., 2003; Qiu and Kumar, 2006, 2008; Cerutti et al., 2007; Liu et al., 2008; Rigatos, 2009; Wang et al., 2010; Zineb et al., 2010) and also in continuous chemical processes (Chang and Chen, 2011). It should be noted that an ad hoc approach was used in these studies to synthesize automata. There are thus definite needs for a concrete procedure to build the component models and also a universal model structure to incorporate these components.

Although a hierarchical structure of the batch operations has already been given in S88 (Fleming et al., 1998), it can be utilized for the synthesis of *normal* operating procedures only (Viswanathan et al., 1998a,b). To fulfill the additional requirements in the present work, a different model hierarchy and a set of specific modeling-building steps have been devised to construct automata for characterizing both the normal behaviors and also a wide spectrum of failure mechanisms in batch chemical processes. A so-called “diagnoser” can then be constructed accordingly to predict all observable event sequences in the given system, to determine diagnosable failures and fault origins, and to compute quantitative performance measures based on the well-established concept of Shannon entropy (Shannon, 1948). These qualitative and quantitative assessment results can then be used as the basis for introducing design changes to enhance the diagnostic performance. Two specific options can be considered: (1) identifying and installing additional sensors which are not included in the P&ID and (2) synthesizing and executing extra operation steps which are not provided in the SFC. Extensive case studies have been performed in this study to verify the feasibility and benefits of the proposed automata based approach.

The remainder of this article is organized as follows. To facilitate explanation of the proposed model-building approach, the general framework of automata and also the hierarchical structure of batch processes are first briefly described in the next two sections. A systematic procedure is then developed in Section 4 to construct the component models and also the live system model according to the P&ID and SFC of the given batch process. A number of qualitative and quantitative measures are adopted in this work to assess the corresponding diagnostic performance. This performance evaluation method is outlined in Section 5. In order to maximize diagnostic resolution, two practically feasible design options are discussed next. A simple liquid-storage system is adopted in this paper to illustrate the aforementioned model-building and performance assessment procedures. In order to further demonstrate the effectiveness of the proposed strategy, two more realistic examples are also presented in Section 7. Finally, conclusions and also some comments on future works are given at the end of this paper.

2. General framework of automata

As mentioned previously, automata are used in this study for the purpose of modeling the batch operations. To facilitate clear description of the proposed method, a brief summary of the automaton structure is first given here. Specifically, a deterministic automaton A can be regarded as a six-tuple (Cassandras and Lafortune, 1999):

$$A=(S, E, f, \Sigma, s_0, S_m) \qquad (1)$$

where S is the set of system states; E is the event set; $f: S\times E\rightarrow S$ represents the transition function; $\Sigma: S\rightarrow 2^E$ denotes the active event function; s_0 is the initial system state; $S_m\subseteq S$ is the set of marked states. The transition function $f(s, e)=t$ means that a transition from state $s\in S$ to state $t\in S$ is caused by the feasible event $e\in E$, while the active event function $\Sigma(s)$ can be regarded as the set of active events at state s.

Notice that every automaton can also be viewed as a *language-generating machine*. The events in set E should be regarded as the alphabets of this language and an event sequence allowed in automaton is regarded as a trace, string or word (*trace* is used in this work). The event set E can be further partitioned into subsets of observable and unobservable events, i.e., $E=E_o\dot{\cup}E_{uo}$. Another unobservable subset $E_f\subseteq E_{uo}\subseteq E$ can also be introduced to characterize the failure events which are to be diagnosed. In our applications, any trace that contains one or more failure event is regarded as a *fault propagation scenario*.

3. Hierarchical structure of batch processes

It has been well recognized that every batch process can be unambiguously described with a P&ID and a SFC. Basically every identifiable hardware item in the batch process is treated as a component in this work and they are classified into a 5-

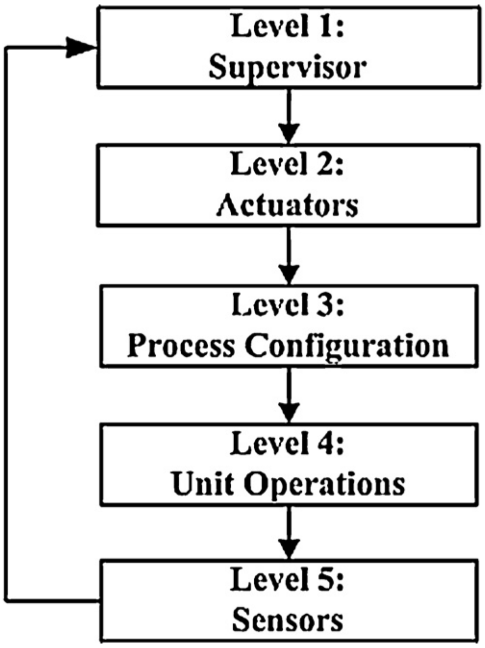

Fig. 1 – Hierarchical structure of a batch process.

level hierarchy according to Fig. 1. The top-level component, supervisor, is usually a programmable logic controller (PLC) used for executing the given SFC so as to alter the actuator states. More than one 2nd-level actuator may be present in the batch process, e.g., hand valves, control valves, switches, pump, and compressor, etc. These actuators are installed for the purposes of adjusting the process configuration, i.e., the material and/or energy flow patterns in the given system, which is viewed as the 3rd-level component in the hierarchy. Every major unit operation in P&ID, such as reaction, separation, heat exchange and storage, is considered as a level-4 component, while every on-line sensor is treated as a component in level 5.

To further illustrate this hierarchical structure, let us consider the liquid-storage system presented in Fig. 2. This problem was studied in Chen et al. (2010) and will later be referred to as Example 1 in this paper. The height of liquid level in this tank is monitored on-line. Two distinct sensor signals, i.e. (1) LH (level high) and (2) LL (level low), are sent to a PLC to actuate the control valves (V-1 and V-2) on the outlet and inlet pipelines (P-1 and P-2) respectively. Under the assumptions that the initial liquid level in tank is low and both valves are at the close positions initially, a sequential function chart can be produced to represent the needed cyclic operating procedure (see Fig. 3). Notice that S_i ($i = 0, 1, 2$) and T_j ($j = 1, 2, 3$) denote the operation steps and the activation conditions of these steps respectively. Notice also that the control actions taken in each step and the sensor signals used in each condition are also specified in this chart. It is clear that the components in this system can be classified into five hierarchical levels, i.e., the programmable logic controller (PLC), the solenoid valves (V-1 and V-2), the pipelines (P-1 and P-2), the storage tank (T-1), and the level sensor (S-1).

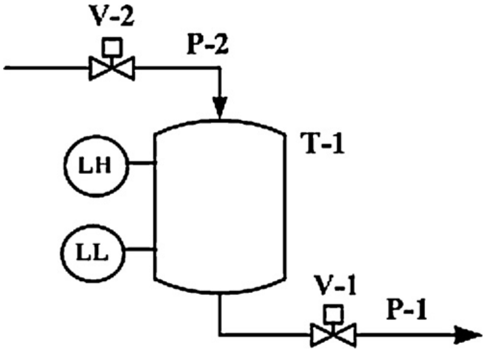

Fig. 2 – A simple liquid storage system (Example 1) (Chen et al., 2010).

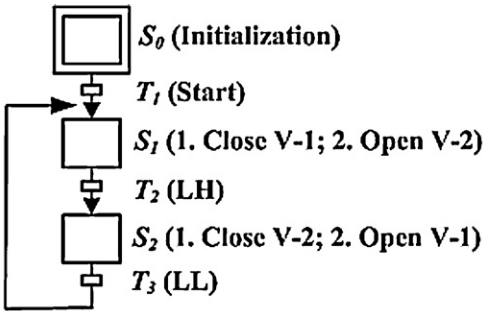
S_0 (Initialization)
T_1 (Start)
S_1 (1. Close V-1; 2. Open V-2)
T_2 (LH)
S_2 (1. Close V-2; 2. Open V-1)
T_3 (LL)

Fig. 3 – Sequential function chart of the operating procedure in Example 1.

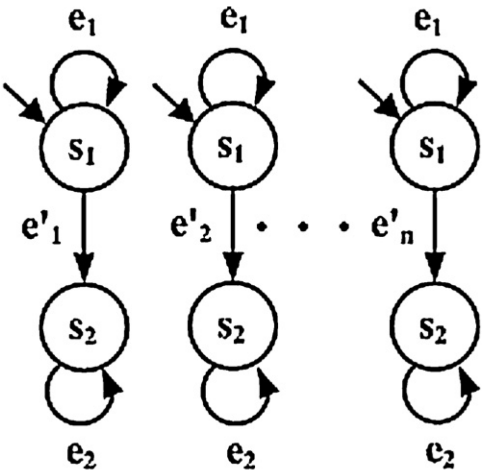

Fig. 4 – Common features of normal component models.

4. Systematic model-building procedure

The model-building procedure consists of three general steps: (1) dividing the batch system into distinct components and then building the corresponding automata according to aforementioned hierarchical structure, (2) combining all components to create a system model by applying the standard *parallel composition* operation (Cassandras and Lafortune, 1999), and (3) introducing the artificial self-looping events “STOP” into the system model to ensure liveliness. These steps are explained below in detail.

4.1. *Step 1: developing component models*

A component model is used to characterize a finite set of identifiable states of the hardware item under consideration and all possible state transition processes. The transition from one state (say s_1) to another (say s_2) may be realized by one or more event and the common features of a *normal* transition process can be represented with the state transition diagram presented in Fig. 4. In this model, the S_1-to-S_2 transition process is triggered by a collection of events (i.e., $e'_1, e'_2, \cdots, e'_n$),

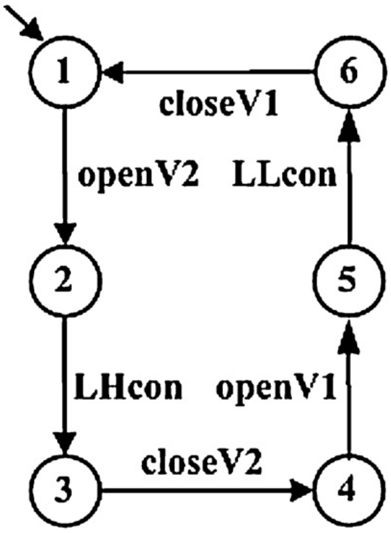

Fig. 5 – The controller model (Example 1).

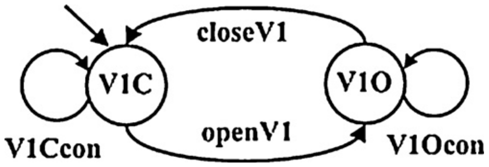

Fig. 6 – The discharge valve model (V-1) (Example 1).

while events e_1 and e_2 result in transitions to their originating states, i.e., the component states are maintained at s_1 and s_2 respectively. The former events are referred to as *the state-transition events* in this paper and the latter *the state-maintaining events*. Notice that every initial state in this model is marked by attaching an incoming arrow without origins. Since the initial state in this case is s_1, the state-maintaining event e_2 can only be enabled after all state-transition events (e_i') are triggered. Finally, it should be noted that the state-transition events in a component model should always be the state-maintaining events in the higher-level models.

The automaton representations of all components in the aforementioned liquid storage system under normal operating conditions are briefly outlined below:

- *Level 1*: The PLC model can be constructed in a straightforward fashion according to Fig. 3 (see Fig. 5). For simplicity, it is assumed that the operation steps in S_1 can always be executed initially and thus the event specified in S_0 is omitted. The level-5 events *LHcon* and *LLcon* is used to represent the situations when the sensor reading continues at the high and low levels for a long enough period respectively. The events *openV1* and *closeV1* are the control actions to open and close valve V-1, while *openV2* and *closeV2* denote the corresponding control actions to manipulate V-2.
- *Level 2*: The automaton model of valve V-1 is presented in Fig. 6. States *V1C* and *V1O* are used to represent the close and open positions respectively, while the events *openV1* and *closeV1* denote the corresponding close-to-open and open-to-close processes. From Fig. 3, it is clear that these two events are triggered by the control actions of PLC, which is a level-1 component. On the other hand, the level-2 events *V1Ccon* and *V1Ocon* represent V-1 continues at close and open positions respectively for a sufficiently long period of time. A similar model can be built for V-2 with the same approach.

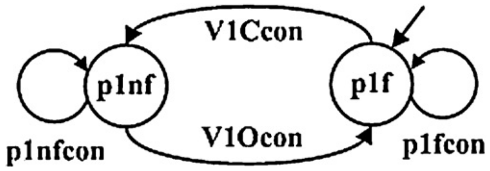

Fig. 7 – The outlet pipeline model (P-1) (Example 1).

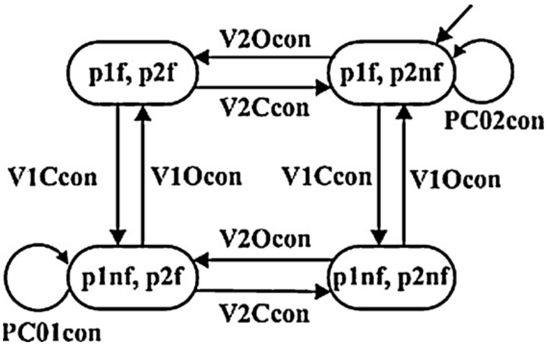

Fig. 8 – The component model representing process configuration (Example 1).

- *Level 3*: The automaton model of the outlet pipeline is presented in Fig. 7. There are two pipeline states, i.e., "flow" (*p1f*) and "no flow" (*p1nf*). It should be noted that, the level-2 events *V1Ocon* and *V1Ccon* should cause the "no-flow-to-flow" and "flow-to-no-flow" processes respectively. For illustration simplicity, it is assumed in this example that the flow in outlet pipeline can be produced by opening V-1 even when the liquid level in tank is low. This assumption is removed in the other two examples. A similar model for the inlet pipeline can be built with the same approach. The component model representing process configuration can then be obtained by performing *parallel composition* on the above two pipeline models (see Fig. 8). For completeness, a detailed description of this method and a simple illustrative example are provided in Supplementary Material (Part I.1). Notice that the state-maintaining events *PC01con* and *PC02con* respectively denote that the process configuration is maintained at *PC01* (in which the pipeline states are *p1nf* and *p2f*) and *PC02* (in which the pipeline states are *p1f* and *p2nf*) for a long enough period of time. Notice also that the state-maintaining events associated with the other two configurations, i.e., (*p1f*, *p2f*) and (*p1nf*, *p2nf*), are neglected on the ground that, in normal operation, these states may be unidentifiable or at best present for a very short period of time only.
- *Level 4*: The automaton model of storage tank is presented in Fig. 9. It can be observed that two tank states are used in this model, i.e., "level high" (state *LH*) and "level low" (state *LL*). Notice that, if the process configuration is kept at *PC01con* (or the pipeline states continue at *p1nfcon* and *p2fcon*), the

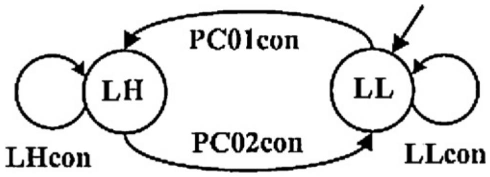

Fig. 9 – The tank model (Example 1).

Please cite this article in press as: Yeh, M.-L., Chang, C.-T., An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes. Chem Eng Res Des (2011), doi:10.1016/j.cherd.2011.05.007

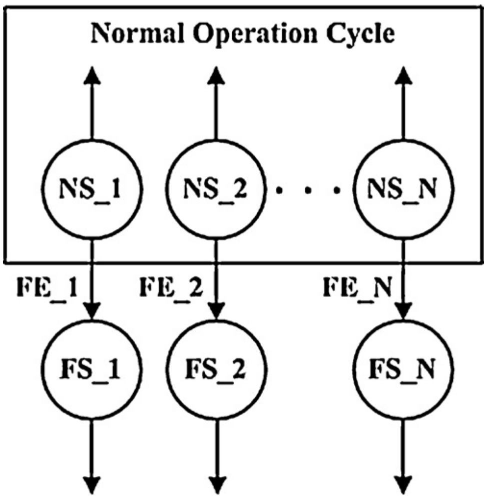

Fig. 10 – The general failure model.

LL-to-LH process should be realized. By the same token, the events *PC02con* should cause state transition in the opposite direction. Finally, the level-4 events *LHcon* and *LLcon* represent the liquid level continues at high and low positions respectively.

- *Level* 5: For the sake of brevity, it is assumed in the present example that only the liquid level is monitored online and the chance of sensor malfunctions is negligibly low. Consequently, the sensor model is omitted here and the measurement readings are considered to be identical to the tank states. Notice that the desired level of sensing-system reliability can almost always be achieved by introducing hardware redundancy in design and also by adopting a proper maintenance policy (Liang and Chang, 2008).

After building the automata to represent normal behaviors of all components, additional mechanisms should then be incorporated to describe failures. The general model structure used to represent possible failures is shown in Fig. 10. The top-layer states in this figure represent normal states, i.e., *NS*$_{-}$*i* (i = 1, 2, ..., *n*), and the boxed automaton represents the normal operation cycle in which only routine events are allowed. Any failure event (i.e., *FE*$_{-}$*i* and *i* = 1, 2, ..., *N*) could result in a change from a normal state within the box to a failure state outside, i.e., *FS*$_{-}$*i* and i = 1, 2, ..., *N*.

Let us first consider the automaton given in Fig. 6, i.e., the normal model of outlet valve, as an example to illustrate the proposed modeling practice (see Fig. 11). In this modified version, the abnormal valve states, i.e., "V-1 sticks at the close position" and "V-1 sticks at the open position", are represented respectively with V1SC and V1SO. Notice that the component state of V-1 is trapped at V1SC in the former scenario and state V1SO can be attached to the normal model in a similar fashion in the latter scenario. Notice also that, since the control actions *open*V1 and *close*V1 cannot cause any state change in either case, they are treated as the state-maintaining events at V1SC and V1SO. Finally, it should be noted that the same approach can be easily adopted to characterize the failures of V-2.

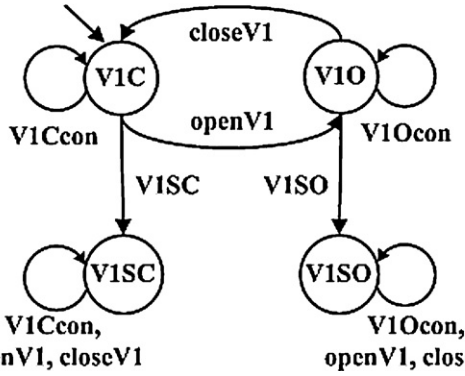

Fig. 11 – The outlet valve model that contains additional failure modules (Example 1).

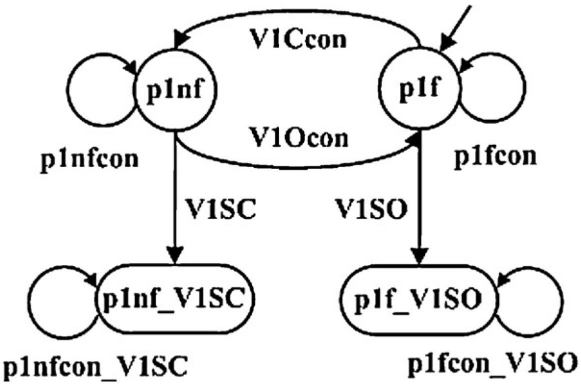

Fig. 12 – The simplified outlet pipeline model (P-1) that contains additional failure modules (Example 1).

Table 1 – Process configurations in Example 1.

Valve states	Process configuration	Symbol
V1C, V2O	p1nf, p2f	PC01
V1O, V2C	p1f, p2nf	PC02
V1C, V2SC	p1nf, p2nf	PC03
V1O, V2SO	p1f, p2f	PC04
V1SC, V2C	p1nf, p2nf	PC05
V1SC, V2O	p1nf, p2f	PC06
V1SO, V2O	p1f, p2f	PC07

Since additional failure states and events are introduced into the normal valve models, it becomes necessary to modify the directly affected component models in the third level. Specifically, the normal outlet pipeline models in Fig. 7 and should be replaced respectively with the ones given in Fig. 12. A similar model can also be built for P-2 with the same approach. In the component model of outlet pipeline (P-1), failure V1SC should cause the normal state *p1nf* to be trapped in a new abnormal state *p1nf*$_{-}$*V1SC* and failure *V1SO* should change the component state from *p1f* to the new state *p1f*$_{-}$*V1SO*. Notice that the failures of P-2 can be characterized with the same approach. By applying parallel composition with these two revised pipeline models, a complete representation of all possible process configurations can then be produced. As indicated previously in constructing the normal component model in Fig. 8, some of the configurations (states) can be judiciously ignored for simplicity. The same approach can be taken in this case and the kept states are listed in Table 1.

If an additional failure, i.e., the tank leakage, is to be considered in Example 1, then the automaton in Fig. 9 should be modified with that given in Fig. 13. The abnormal tank states, i.e., *LH*$_{-}$*leak* and *LL*$_{-}$*leak*, represent "leakage occurs while level high" and "leakage occurs while level low" respectively. It should be noted that the event *LHcon* is not allowed when tank leakage occurs. Notice also that the abnormal process configurations *PC04con* and *PC05con* may cause the

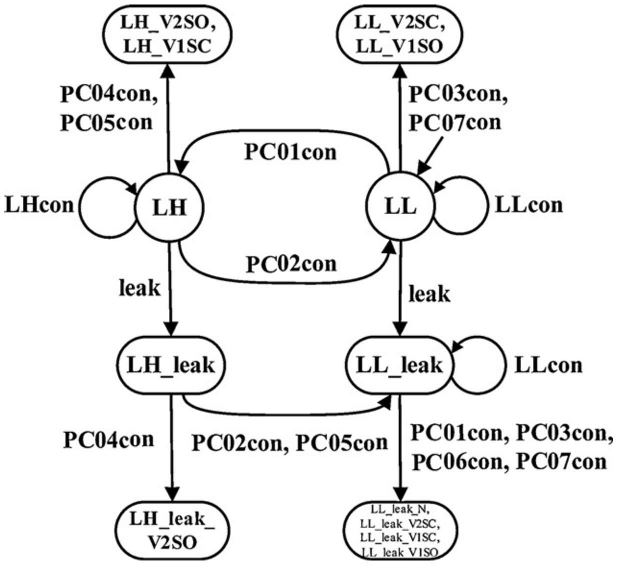

Fig. 13 – The tank model that contains additional failure modules (Example 1).

normal tank state *LH* to be trapped in the abnormal states *LH_V2SO* and *LH_V1SC* respectively. Similarly, the abnormal process configurations *PC03con* and *PC07con* may cause the normal tank state *LL* to be trapped in the abnormal states *LL_V2SC* and *LL_V1SO* respectively. In addition, the abnormal process configuration *PC04con* could also cause the abnormal tank state *LH_leak* to be trapped in another abnormal state *LH_leak_V2SO*, while *PC01con*, *PC03con*, *PC06con* and *PC07con* could cause the abnormal tank state *LH_leak* to be trapped in the abnormal states *LL_leak_N*, *LL_leak_V2SC*, *LL_leak_V1SC* and *LL_leak_V1SO* respectively.

4.2. *Step 2: assembling system model*

Although both normal behaviors and failure mechanisms can be incorporated in a component model, there is still a need to impose additional constraints in the controller model to limit the scope of event-sequence evolution. These modifications are introduced mainly for the purpose of avoiding state explosion and also producing a succinct diagnoser. Specifically, it is assumed in this study that:

(1) All state-maintaining events of the components in levels 2–4 (i.e., actuators, process configuration and unit operations) should occur before the sensor state reaches the resulting activation condition in SFC.
(2) The failure event of any component in levels 2–5 (i.e., actuators, process configuration, unit operations and sensors) can only occur just before the controller triggers a subsequent actuator event. Furthermore, the aforementioned failure and actuator events are mutually exclusive.

These constraints can be incorporated into the *controller model* with additional self-looping transitions according to the following rules:

(1) Every state-maintaining event in level 2 should be incorporated with a self-looping transition at the controller state that enables the corresponding activation condition(s) in SFC.

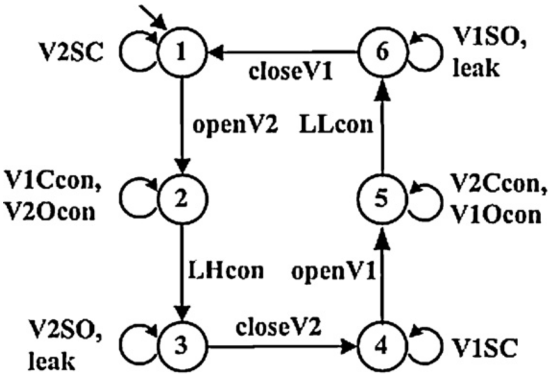

Fig. 14 – The tank model that contains additional failure modules (Example 1).

(2) Every failure event in levels 2–5 should be incorporated with a self-looping transition at the controller state that enables the subsequent normal control action(s) in SFC.

In the liquid storage system mentioned above, let us assume that

- the inlet valve may stick at close or open position,
- the outlet valve may stick at close or open position, and
- the tank may leak.

The modified controller model for this example is given in Fig. 14. Notice that

- The state-maintaining events *V1Ocon* and *V2Ccon* of actuators V-1 and V-2 are both constrained at state 5, which is the controller state that enables the activation condition *LLcon* in SFC. Similarly, the state-maintaining events *V1Ccon* and *V2Ocon* should both be constrained at state 2, which is the state that enables the activation condition *LHcon* in SFC.
- The failures of V-1, i.e., *V1SC* and *V1SO* in Fig. 11, should be constrained at state 4 (which enables the control action *openV1*) and state 6 (which enables the control action *closeV1*) respectively. Similarly, the failures of V-2, i.e., *V2SC* and *V2SO*, are constrained at states 1 and 3 respectively for the same reasons. On the other hand, the tank failure, i.e., *leak* should be constrained at the states 3 and 6 (which enable the control actions S_2 and S_1 respectively).

After introducing the aforementioned modifications, the parallel composition operation can be performed to integrate all component models into a system model.

4.3. *Step 3: ensuring liveliness*

The automaton $A^{non-live}$ obtained in step 2 is in fact non-live, i.e., there exists at least a *dead state* in the system and no feasible events are available to cause a transition to any other state. A precise definition of the dead state and also a simple example can be found in Supplementary Material (Part I.2). According to Sampath et al. (1995, 1996), a non-live automaton must be converted to a live one A^{live} before constructing the diagnoser. This task can be accomplished simply by adding a fictitious self-loop event “*STOP*” at every dead state of $A^{non-live}$. Clearly $STOP \subset E_o$, i.e., the permanently stationary system state should be observable in batch operation. For instance, let us consider the trace resulted from failure *V1SC* in the aforementioned storage system (see the partial automaton shown in

Please cite this article in press as: Yeh, M.-L., Chang, C.-T., An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes. Chem Eng Res Des (2011), doi:10.1016/j.cherd.2011.05.007

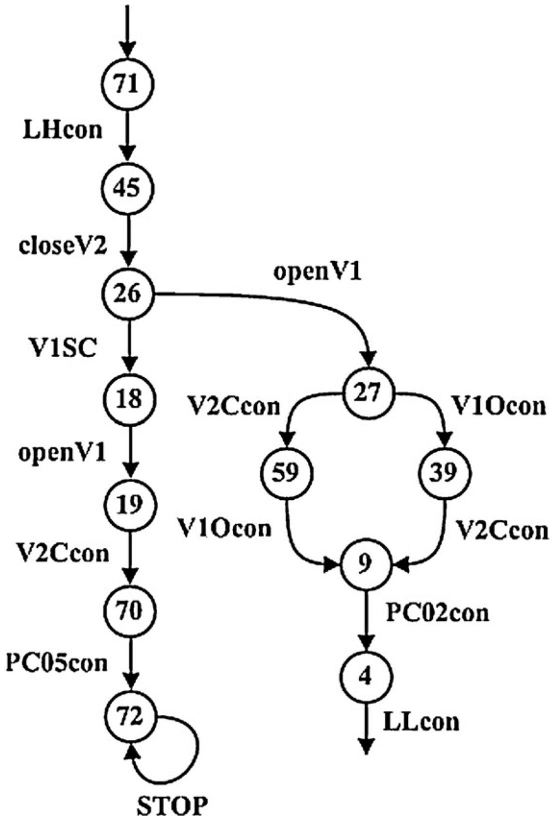

Fig. 15 – Partial automaton in Example 1.

Fig. 15). Note that the self-loop event *STOP* has been added at the end state 53 of the failed trace to ensure liveliness.

5. Assessment of diagnostic performance

On the basis of the live system model, a diagnoser can be constructed (Sampath et al., 1995, 1996) and the corresponding diagnostic performance can then be assessed properly. Following is a detailed description of the implementation method adopted in this work.

5.1. Qualitative measures

The automaton model can be used to generate all possible event sequences, i.e., traces, in the given system according to well-established algorithms (Lafortune and Teneketzis, 2000). For illustration convenience, let us consider a fictitious system in which the traces given in Fig. 16 can be identified. Let $E_{uo} = \left\{a, b, F_1, F_2, F_3, F_4\right\}$ and $E_f = \{F_1, F_2, F_3, F_4\}$. Thus, Tr_j (j = 1, 2, ..., 6) denotes a trace which consists of both observable and unobservable events, and the corresponding observable trace Tr_{oi} (i = 1, 2, 3, 4) is specified in the adjacent parenthesis. Notice that Tr_3, Tr_4 and Tr_5 all result in the same observable trace, i.e., Tr_{o3}, while Tr_1 and Tr_2 can be identified from two distinct observable traces Tr_{o1} and Tr_{o2} respectively. In addition, Tr_1–Tr_5 are finite failed traces ending at terminal states 1–5 respectively, while Tr_6 is a cyclic normal trace which is denoted by the symbol *. As mentioned previously, each abnormal trace should be viewed as the event sequence occurred in a possible fault propagation scenario. To facilitate precise description of the diagnoser, all failure events on a particular trace is collectively referred to as the *fault origin* of the corresponding scenario in this paper.

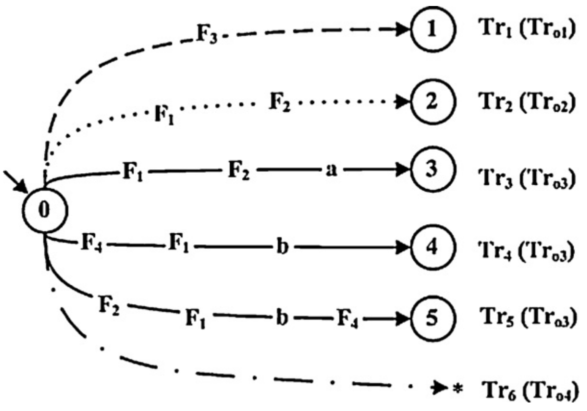

Fig. 16 – All possible traces in a fictitious system.

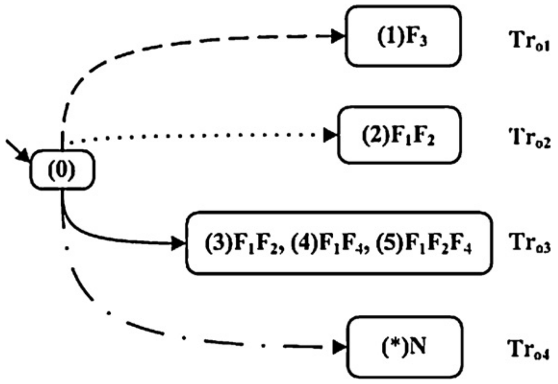

Fig. 17 – The diagnoser built from Fig. 16.

A fault propagation scenario is considered to be *detectable* if the corresponding *observable* event sequence is not the same as that on the normal trace. However, since the same observable sequence may result from several different candidate root causes, the fault origin of a detectable scenario can be considered to be *diagnosable* if and only if the corresponding observable trace is unique. Furthermore, notice that a fault origin may consist of more than one failure. A failure event in a detectable scenario can thus be regarded as *confirmable* if its presence (or absence) can be unambiguously determined according to the observable trace. If a failure is confirmable in every trace of the diagnoser, then it can be considered to be *diagnosable* also.

According to Cassandras and Lafortune (1999), the diagnoser of the fictitious system described in Fig. 16 can be built by lumping the unobservable events with the observable ones on the same trace and then merging the identical observable traces (see Fig. 17). For instance, since events *a*, F_1 and F_2 are unobservable in trace Tr_3, they should be hidden to form an observable trace in the diagnoser. In addition, since the terminal states 3–5 in Fig. 16 are propagated along the same observable trace Tr_{o3}, they should all be merged into one. The failure events in each fault propagation scenario are specified in the terminal node, while the normal trace is marked with the label *N*. The node numbers in the original automaton are given in the parentheses in the corresponding initial and terminal nodes of diagnoser.

On the basis of the diagnoser presented in Fig. 16, the diagnostic performance of the given system can be evaluated *qualitatively* first. In particular, on-line observation of Tr_{o3} indicates that (1) there are three possible scenarios (or fault origins), (2) the presence of failure F_1 and the absence of failure F_3 are both certain, and (3) at least one of the

Please cite this article in press as: Yeh, M.-L., Chang, C.-T., An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes. Chem Eng Res Des (2011), doi:10.1016/j.cherd.2011.05.007

Table 2 – Diagnostic performance measures of the liquid storage system in Example 1.

Sensors	Range of uncertainty index		Observable traces	Confirmable failures (an underline denotes absence, otherwise the failure is present)	Candidate fault origins
	$\overline{H}_{\min}$	$\overline{H}_{\max}$			
L	0.2594	0.5511	1	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$	F_3
			2	$\underline{F_2}$, $\underline{F_3}$	$F_1F_4F_5$, F_4F_5, F_1F_4, F_1, F_4
			3	$\underline{F_4}$	$F_1F_3F_5$, F_2, F_3F_5, F_1F_5, F_2F_3, F_2F_5, F_3, $F_2F_3F_5$, F_5
L, P-1	0.0602	0.3597	1	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$	F_3
			2	F_1, $\underline{F_2}$, $\underline{F_3}$	$F_1F_4F_5$, F_1F_4, F_1
			3	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, F_4	F_4F_5, F_4
			4	F_1, $\underline{F_2}$, $\underline{F_4}$, F_5	$F_1F_3F_5$, F_1F_5
			5	$\underline{F_1}$, F_2, $\underline{F_4}$	F_2, F_2F_5, F_2F_3, $F_2F_3F_5$
			6	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_4}$	F_3F_5, F_3, F_5
L, P-2	0.0602	0.3806	1	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$	F_3
			2	$\underline{F_2}$, $\underline{F_3}$, F_4	$F_1F_4F_5$, F_4F_5, F_4 F_1F_4
			3	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$	F_1
			4	F_3, $\underline{F_4}$	$F_1F_3F_5$, F_3F_5, F_2F_3, F_3, $F_2F_3F_5$
			5	$\underline{F_3}$, $\underline{F_4}$	F_2, F_1F_5, F_2F_5, F_5
L, P-1, P-2	0.0000	0.1505	1	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$	F_3
			2	F_1, $\underline{F_2}$, $\underline{F_3}$, F_4	F_1F_4, $F_1F_4F_5$
			3	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, F_4	F_4F_5, F_4
			4	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$	F_1
			5	F_1, $\underline{F_2}$, F_3, $\underline{F_4}$, F_5	$F_1F_3F_5$
			6	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, F_5	F_1F_5
			7	$\underline{F_1}$, F_2, $\underline{F_3}$, $\underline{F_4}$	F_2, F_2F_5
			8	$\underline{F_1}$, F_2, F_3, $\underline{F_4}$	F_2F_3, $F_2F_3F_5$
			9	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$	F_3, F_3F_5
			10	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, F_5	F_5

remaining two possible failures F_2 and F_4 can be confirmed. Similarly, Tr_{o1} confirms the occurrence of failure F_3 and the absence of all other failures, while Tr_{o2} confirms the occurrence of both F_1 and F_2 and the absence of F_3 and F_4. It can therefore be concluded that (1) the fault origins of Tr_{o1} and Tr_{o2} are diagnosable, and (2) the failures F_1 and F_3 are diagnosable.

5.2. *Quantitative measures*

The diagnostic performance of a batch monitoring system can also be characterized quantitatively in terms of Shannon entropy (1948). In particular, the degree of diagnostic uncertainty associated with the ith observable trace in diagnoser can be written as

$$H_i = -\sum_{j=1}^{N_i^{FO}} p_{i,j} \, \log p_{i,j} \tag{2}$$

where H_i is the uncertainty measure of observable trace Tr_{oi}; $p_{i,j}$ is the conditional probability of candidate fault origin j after observing trace Tr_{oi}; N_i^{FO} is the number of candidate fault origins confirmed by Tr_{oi}.

Notice that the aforementioned conditional probabilities cannot really be evaluated accurately since the failure-rate data are often unavailable or unreliable in practical applications. To properly assess the diagnostic performance, the upper and lower bounds of uncertainty index are *estimated* in this study. The upper uncertainty limit can be determined by assuming that, before observing trace i, the occurrence probabilities of all its fault origins are roughly the same. As a result, it can be deduced that their conditional probabilities are:

$$p_{i,1} = p_{i,2} = \ldots = \frac{1}{N_i^{FO}} \tag{3}$$

On the other hand, the approximate lower uncertainty limit can be calculated by assuming that the occurrence probabilities of all failures are equal and quite small. As a result, the following formulas can be used:

$$p_{i,1} = p_{i,2} = \cdots = \frac{1}{N_i^{FS}} \tag{4}$$

where N_i^{FS} is the number of single-failure origins confirmed by Tr_{oi}. Notice that, if $N_i^{FS} = 0$, then the corresponding trace should be neglected in calculating the lower limit.

Quantitative measures of the overall diagnostic performance can then be determined by taking the averages of the upper and lower limits of the *uncertainty indices* associated with all traces in diagnoser. As an example, the average upper and lower limits of uncertainty index of the diagnoser in Fig. 17 can be determined respectively as follows:

- The average upper limit of uncertainty index:

$$\overline{H}_{\max} = -\frac{1}{3}\left(\log 1 + \log 1 + \log \frac{1}{3}\right) = 0.1590$$

- The average lower limit of uncertainty index:

$$\overline{H}_{\min} = -\frac{1}{1}\log 1 = 0$$

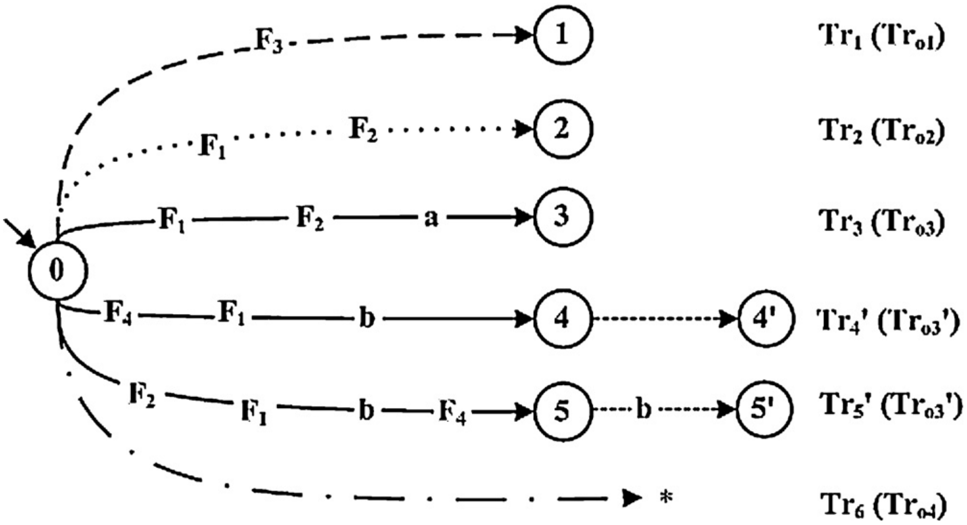

Fig. 18 – Traces extended from those in Fig. 16 after executing extra operation steps.

5.2.1. *Performance measures of a simple batch system*

The diagnoser for the liquid storage system in Figs. 2 and 3 can be characterized with the performance measures listed in Table 2. Notice that the failures F_1–F_5 represent *V1SO*, *V1SC*, *V2SO*, *V2SC* and *leak* respectively. If only the level sensor is installed in this system, it can be observed from the first row in the table that (1) there are three different observable traces, (2) the fault origin of the first trace is diagnosable, and (3) none of the failures are diagnosable. Notice also that the upper and lower bounds of average uncertainty index are both quite large. Thus, it would be desirable to improve the diagnostic performance by incorporating the design options described in the following section. Notice that a wide spectrum of failure mechanisms has been analyzed and a comprehensive evaluation of the overall diagnostic resolution has been carried out in this work, whereas the assessment in Chen et al. (2010) was limited to the single-failure scenarios only.

6. Performance enhancement options

Two performance enhancement options are considered in this work, i.e. (1) installing additional sensors which are not included in the P&ID and (2) executing extra operation steps which are not specified in the SFC. The effectiveness of these options can be evaluated on the basis of the proposed performance measures.

6.1. *Additional sensors*

The simplest way to enhance diagnostic resolution is to use more sensors. The obvious selection strategy is to measure hidden events in the diagnoser which can respond differently to the fault origins. Let us use Fig. 18 again as an example to illustrate this approach. Intuitively, trace Tr_3 could be distinguished from traces Tr_4 and Tr_5 if the originally unobservable event *a* or *b* is measured on-line. Notice that Tr_4 and Tr_5 are inseparable since they form the same observable traces even after measuring the hidden event *b*. Consequently, the failure event F_2 is always undiagnosable.

In Example 1, the upper and lower limits of average uncertainty index can be determined to be 0.5511 and 0.2594 respectively if only the level sensor is available on-line. After installing additional sensors on pipelines P-1 and P-2, these limits can be lowered to 0.1505 and 0 respectively. All failures except F_5 in this system become diagnosable, while the fault origins of traces 1, 4, 5, 6 and 10 are also diagnosable (see Table 2).

6.2. *Extra operation steps*

Another approach to improve diagnostic performance is to perform extra operation steps which are not included in the original SFC. As indicated earlier concerning the fictitious system in Fig. 16, Tr_4 and Tr_5 cannot be distinguished from one another by placing additional sensors. For illustration convenience, let us assume that these two traces could be extended to form traces Tr_4' and Tr_5' in Fig. 18 respectively by executing additional operation steps. Since the additional event *b* exists only in the latter trace, the corresponding fault origins become diagnosable if the sensor for detecting event *b* can be made available. In this work, a systematic procedure has been developed to construct the extended diagnoser if these extra operation steps can be specifically assigned by the designer. The detailed implementation processes are reported in the following examples.

7. Applications

In order to demonstrate the effectiveness of the proposed approach to evaluate and improve the diagnostic performance of realistic fault monitoring systems, two additional case studies have been carried out in this work. The first application is concerned with a three-tank storage system, while the second a beer filtration plant. In these case studies, the software tools DESUMA and UMDES (Lafortune and Teneketzis, 2000) were adopted to perform various standard automata-based operations, e.g., parallel composition, trace generation, liveliness assurance and diagnoser synthesis, etc. The application results are outlined below.

7.1. *Example 2*

Let us consider the three-tank storage system presented in Fig. 19. This problem was also studied in Chen et al. (2010). Pipeline P-1 is the inlet pipeline of tank T-1 and its flow is controlled with valve V-1. The outlet pipeline of T-1 is P-2, which is connected to a 3-way valve V-2, and a pump is installed on P-2. When V-2 is at the position "+", the fluid in P-2 will

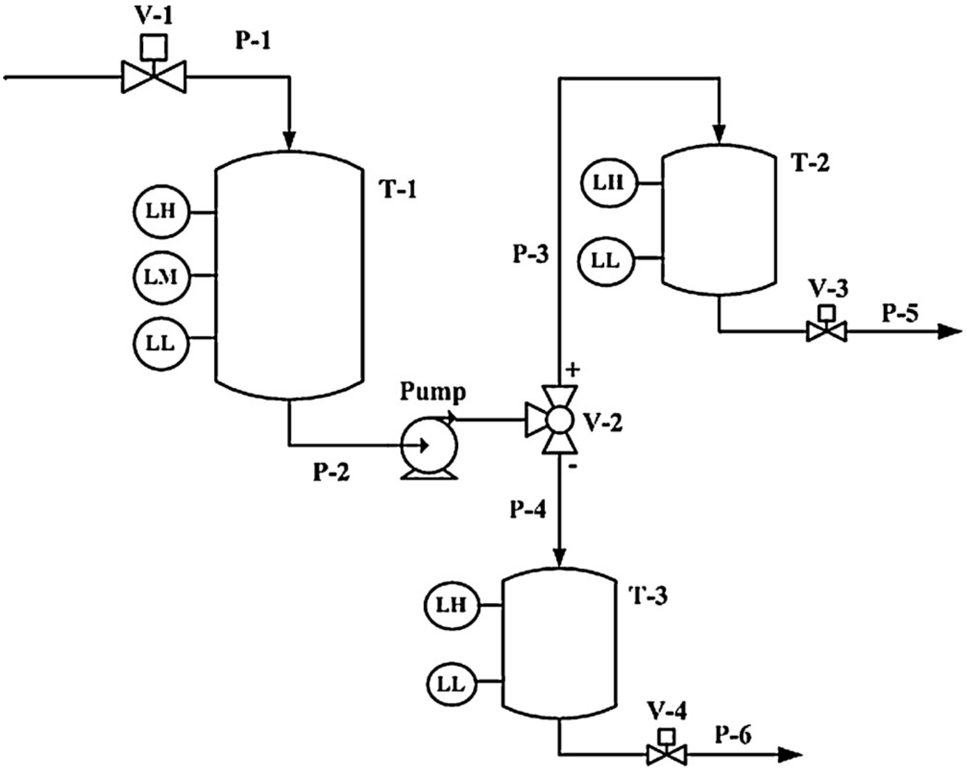

Fig. 19 – A three-tank storage system (Example 2) (Chen et al., 2010).

be transferred into pipeline P-3 and then tank T-2. If V-2 is at the position "−", the fluid in P-2 will flow into pipeline P-4 and enter tank T-3. Valve V-3 is used to discharge the material in tank T-2, while V-4 is for T-3. Tank T-1 has a level sensor installed, and the sensor reports three conditions: (1) level low (LL); (2) level middle (LM); (3) level high (LH). The level sensors on tanks T-2 and T-3 can be used to detect only two conditions, i.e. (1) level low (LL) and (2) level high (LH). The operation steps and activation conditions of SFC are listed in Tables 3 and 4, respectively.

In this case, the height of liquid level in every tank is assumed to be observable.

The possible failures considered here are:

(1) Valves V-1 and V-3 may experience sticking failures.
(2) Valve V-2 may be switched to a wrong position due to spurious controller signal(s).
(3) Tank T-2 may leak.

Table 3 – Operation steps in SFC for three-tank storage system in Example 2 (Chen et al., 2010).

Operation Step	Control actions
S_0	Initialization
S_1	(1) Close V-3 (2) Close V-4 (3) Open V-1
S_2	(1) Close V-1 (2) Switch V-2 to + (3) Switch on pump
S_3	(1) Switch off pump (2) Switch V-2 to − (3) Switch on pump
S_4	(1) Switch off pump (2) Open V-3 (3) Open V-4

Table 4 – Activation conditions in SFC for three-tank storage system in Example 2 (Chen et al., 2010).

Symbol	Conditions
T_1	Start
T_2	T1H
T_3	T1M & T2H
T_4	T1L & T3H
T_5	T2L & T3L

For the sake of brevity, all component models in this example are presented in Supplementary Material (Part II). The system model and the corresponding diagnoser can be assembled on the basis of these component models. Notice that, since spurious controller signal(s) may be generated in implementing the normal operation step S_2 or S_3 (see Table 3), the extra step selected here for enhancing diagnostic performance is to execute step S_2 again when controller stops at S_2. The component models for this diagnostic control plan can also be found Part II of the Supplementary Material.

The effects of implementing various combinations of the proposed performance enhancement measures are summarized in Tables 5 and 6. Notice that failures F_1–F_7 represent $V1SC$, $V1SO$, $V2M-$, $V2M+$, $V3SC$, $V3SO$, and $T2leak$ respectively. It can be observed from the first row in Table 5 that, if only the level sensors are installed in this system, there should be 11 different observable traces and the lowest upper and lower limits of uncertainty index can be determined to be 0.3001 and 0.0376 respectively. Notice that the diagnostic performance cannot be improved further by installing additional flow sensors except on pipeline P-5. If an additional sensor is installed on pipeline P-5, these limits can be lowered to 0.1338 and 0 respectively and all failures except F_6 and F_7 become diagnosable. On the other hand, if only the extra operation steps are applied without additional sensors these limits are 0.2825 and 0.0430 respectively. Furthermore, the best diagnostic perfor-

Table 5 – Diagnostic performance measures of the three-tank storage system in Example 2—results obtained without extra operation steps.

Sensors	Range of uncertainty index		Observable traces	Confirmable failures (an underline denotes absence, otherwise the failure is present)	Candidate fault origins
	$\overline{H}_{min}$	$\overline{H}_{max}$			
L	0.0376	0.3001	1	$F_1, \underline{F_2}, \underline{F_3}, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_1
			2	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_3
			3	$\underline{F_1}, F_2, F_3, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_2F_3
			4	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_2
			5	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_4, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_4
			6	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5, \underline{F_6}, \underline{F_7}$	F_5
			7	$F_1, \underline{F_2}, \underline{F_3}, \underline{F_4}$	$F_1F_6F_7, F_1F_7, F_1F_6, F_1F_5F_7, F_1$
			8	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}$	$F_3F_6F_7, F_3F_7, F_3F_6, F_3F_5F_7, F_3$
			9	$\underline{F_1}, F_2, F_3, \underline{F_4}, \underline{F_5}$	$F_2F_3F_7, F_2F_3F_6, F_2F_3F_5F_7, F_2F_3F_6F_7, F_2F_3$
			10	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}$	$F_2F_7, F_2F_6, F_2F_5F_7, F_2F_6F_7$
			11	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}$	F_6F_7, F_7, F_6, F_5F_7
L, P-5	0	0.1338	1	$F_1, \underline{F_2}, \underline{F_3}, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_1
			2	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_3
			3	$\underline{F_1}, F_2, F_3, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_2F_3
			4	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_2
			5	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_4, \underline{F_5}, \underline{F_6}, \underline{F_7}$	F_4
			6	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5, \underline{F_6}, \underline{F_7}$	F_5
			7	$F_1, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5, \underline{F_6}, F_7$	$F_1F_5F_7$
			8	$F_1, \underline{F_2}, \underline{F_3}, \underline{F_4}, \underline{F_5}$	$F_1F_6F_7, F_1F_7, F_1F_6, F_1$
			9	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, F_5, \underline{F_6}, F_7$	$F_3F_5F_7$
			10	$\underline{F_1}, F_2, F_3, \underline{F_4}, F_5, \underline{F_6}, F_7$	$F_2F_3F_5F_7$
			11	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5, \underline{F_6}, F_7$	F_5F_7
			12	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, F_5, \underline{F_6}, F_7$	$F_2F_5F_7$
			13	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, \underline{F_5}$	$F_3F_6F_7, F_3F_7, F_3F_6, F_3$
			14	$\underline{F_1}, F_2, F_3, \underline{F_4}, \underline{F_5}$	$F_2F_3F_7, F_2F_3F_6, F_2F_3F_6F_7, F_2F_3$
			15	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, \underline{F_5}, \underline{F_6}, F_7$	F_2F_7
			16	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, \underline{F_5}, \underline{F_6}, F_7$	F_7
			17	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, \underline{F_5}, F_6$	F_6F_7, F_6
			18	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, \underline{F_5}, F_6$	$F_2F_6, F_2F_6F_7$

mance can be achieved by implementing the aforementioned control plan with additional flow sensor on P-5. In this case, the range of uncertainty index can be improved to [0, 0.0903].

7.2. Example 3

Let us next consider a more realistic problem concerning the beer filtration plant presented in Fig. 20 (Lai et al., 2007; Chung and Lai, 2008). This system consists of two Multi-Micro System filters (MMS-1 and MMS-2), two buffer tanks (T-1 and T-2), a supply and collection system for the cleanser in pipe (CIP). Notice that the filtration process is operated with 16 double-disk piston valves (V-1 to V-16) and each valve can be switched to two alternative positions, i.e. OFF and ON. When a valve is at the OFF position, the fluids in vertical and horizontal pipelines flow separately. On the other hand, if this valve is turned to the

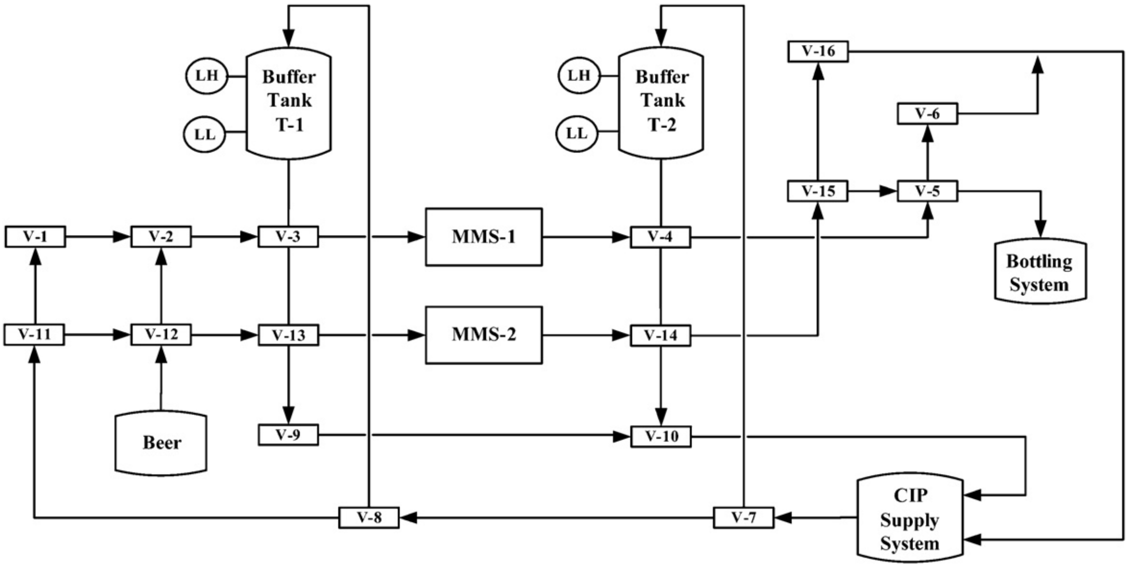

Fig. 20 – A beer filtration plant (Example 3) (Lai et al., 2007; Chung and Lai, 2008).

Please cite this article in press as: Yeh, M.-L., Chang, C.-T., An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes. Chem Eng Res Des (2011), doi:10.1016/j.cherd.2011.05.007

Table 6 – Diagnostic performance measures of the three-tank storage system in Example 2—results obtained with extra operation steps.

Sensors	Range of uncertainty index		Observable traces	Confirmable failures (an underline denotes absence, otherwise the failure is present)	Candidate fault origins
	$\overline{H}_{\min}$	$\overline{H}_{\max}$			
L	0.0430	0.2825	1	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_1
			2	$\underline{F_1}$, F_2, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_2
			3	$\underline{F_1}$, F_2, F_3, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_2F_3
			4	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_3
			5	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, F_4, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_4
			6	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, F_5, $\underline{F_6}$, $\underline{F_7}$	F_5
			7	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$	$F_1F_6F_7$, F_1, F_1F_7, $F_1F_5F_7$, F_1F_6
			8	$\underline{F_1}$, F_2, $\underline{F_3}$, $\underline{F_4}$	$F_2F_6F_7$, F_2F_7, $F_2F_5F_7$, F_2F_6
			9	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$	F_6, F_7, F_6F_7, F_5F_7
			10	$\underline{F_1}$, F_2, F_3, $\underline{F_4}$	$F_2F_3F_6F_7$, $F_2F_3F_6$, $F_2F_3F_7$, $F_2F_3F_5F_7$
			11	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, F_7	$F_3F_6F_7$, F_3F_6, F_3F_7, $F_3F_5F_7$
L, P-5	0	0.0903	1	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_1
			2	$\underline{F_1}$, F_2, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_2
			3	$\underline{F_1}$, F_2, F_3, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_2F_3
			4	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_3
			5	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, F_4, $\underline{F_5}$, $\underline{F_6}$, $\underline{F_7}$	F_4
			6	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, F_5, $\underline{F_6}$, $\underline{F_7}$	F_5
			7	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, F_5, $\underline{F_6}$, F_7	$F_1F_5F_7$
			8	F_1, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$	$F_1F_6F_7$, F_1, F_1F_7, F_1F_6
			9	$\underline{F_1}$, F_2, $\underline{F_3}$, $\underline{F_4}$, F_5, $\underline{F_6}$, F_7	$F_2F_5F_7$
			10	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, F_5, $\underline{F_6}$, F_7	F_5F_7
			11	$\underline{F_1}$, F_2, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, F_7	F_2F_7
			12	$\underline{F_1}$, F_2, F_3, $\underline{F_4}$, F_5, $\underline{F_6}$, F_7	$F_2F_3F_5F_7$
			13	$\underline{F_1}$, F_2, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, F_6	$F_2F_6F_7$, F_2F_6
			14	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, F_6	F_6, F_6F_7
			15	$\underline{F_1}$, $\underline{F_2}$, $\underline{F_3}$, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, F_7	F_7
			16	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, F_5, $\underline{F_6}$, F_7	$F_3F_5F_7$
			17	$\underline{F_1}$, F_2, F_3, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, F_7	$F_2F_3F_7$
			18	$\underline{F_1}$, F_2, F_3, $\underline{F_4}$, $\underline{F_5}$, F_6	$F_2F_3F_6F_7$, $F_2F_3F_6$
			19	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$, F_6	$F_3F_6F_7$, F_3F_6
			20	$\underline{F_1}$, $\underline{F_2}$, F_3, $\underline{F_4}$, $\underline{F_5}$, $\underline{F_6}$, F_7	F_3F_7

ON position, the fluids entering the valve from vertical and horizontal pipelines will be mixed and then together flow out of the valve via all exit pipelines (Lai et al., 2007; Chung and Lai, 2008).

There are four different operations in this beer filtration process, i.e., filling, filtration, bottling and cleaning. The purpose of filling operation is to transport fresh beer from a source tank to the buffer tank T-1. In the filtration operation, the beer is transferred from tank T-1 to T-2 via filter MMS1 or MMS2. Clearly, the filtered beer in T-2 should then be moved to the bottling station in another material-transfer operation. The last operation in the plant is concerned with cleaning the equipments and pipelines in which beer has been processed previously. It is assumed in this example that each item must be cleaned after it has been used for a designated number of times. The operation steps and activation conditions of SFC are listed in Tables 7 and 8, respectively.

Table 7 – Operation steps in SFC for the beer filtration plant (Example 3).

Operation step	Control actions
S_0	Initialization
S_1	(1) Close V-11; (2) Close V-16; (3) Close V-8; (4) Close V-9 (5) Close V-7; (6) Close V-10; (7) Open V-2; (8) Open V-3
S_2	(1) Close V-2; (2) Open V-4
S_3	(1) Close V-3; (2) Open V-5
S_4	(1) Close V-4; (2) Close V-5; (3) Open V-12; (4) Open V-13; (5) Open V-1; (6) Open V-6
S_5	(1) Close V-12; (2) Open V-14
S_6	(1) Close V-13; (2) Close V-1; (3) Close V-6; (4) Open V-15
S_7	(1) Close V-14; (2) Close V-15; (3) Open V-11; (4) Open V-16; (5) Open V-8; (6) Open V-9; (7) Open V-7; (8) Open V-10

Table 8 – Activation conditions in SFC for beer filtration plant (Example 3).

Symbol	Conditions
T_1	Start
T_2	T1H
T_3	T1L & T2H
T_4	T2L
T_5	T1H
T_6	T1L & T2H
T_7	T2L
T_8	t

In this case, the height of liquid level in every tank is assumed to be observable.

The possible failures considered here are:

Please cite this article in press as: Yeh, M.-L., Chang, C.-T., An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes. Chem Eng Res Des (2011), doi:10.1016/j.cherd.2011.05.007

Table 9 – Diagnostic performance measures of the beer filtration plant in Example 3.

Sensors	Range of uncertainty index		Observable traces	Confirmable failures (an underline denotes absence, otherwise the failure is present)	Candidate fault origins
	$\overline{H}_{min}$	$\overline{H}_{max}$			
L	0.1505	0.7341	1	$\underline{F_2}, \underline{F_3}, \underline{F_4}$	F_1F_5, F_1, F_5
			2	$\underline{F_1}, \underline{F_4}, F_5$	$F_2F_3F_5, F_5, F_3F_5, F_2F_5$
			3	None	$F_4F_5, F_2F_4F_5, F_2F_5, F_1F_4F_5, F_3F_5, F_2F_3F_5, F_1F_4, F_1F_3F_5, F_1F_3, F_1F_5, F_1, F_5$
			4	$\underline{F_1}, F_5$	$F_2F_3F_5, F_5, F_3F_5, F_2F_5, F_4F_5, F_2F_4F_5$
L, P-6C	0.2007	0.2306	1	$\underline{F_2}, \underline{F_3}, \underline{F_4}$	F_1F_5, F_1, F_5
			2	$\underline{F_1}, F_2, F_3, \underline{F_4}, F_5$	$F_2F_3F_5$
			3	$\underline{F_1}, \underline{F_4}, F_5$	$F_2F_3F_5, F_3F_5, F_2F_5$
			4	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5$	F_5
			5	$F_3, \underline{F_4}$	$F_3F_5, F_2F_3F_5, F_1F_3F_5, F_1F_3$
			6	$\underline{F_1}, F_2, \underline{F_3}, F_4, F_5$	$F_2F_4F_5$
			7	$\underline{F_2}, \underline{F_3}$	$F_1F_4F_5, F_1F_4, F_1F_5, F_1, F_5$
			8	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, F_5$	F_2F_5
			9	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_4, F_5$	F_4F_5
			10	$\underline{F_1}, F_3, \underline{F_4}, F_5$	$F_2F_3F_5, F_3F_5$
			11	$\underline{F_1}, F_2, \underline{F_3}, F_4, F_5$	$F_2F_4F_5$
			12	$\underline{F_1}, F_2, \underline{F_4}, F_5$	$F_2F_3F_5, F_2F_5$
			13	$\underline{F_1}, \underline{F_3}, F_4, F_5$	$F_4F_5, F_2F_4F_5$
			14	$\underline{F_1}, F_2, F_3, \underline{F_4}, F_5$	$F_2F_3F_5$
			15	$F_1, \underline{F_2}, \underline{F_3}, F_4$	$F_1F_4F_5, F_1F_4$
L, P-23	0	0.1436	1	$F_1, \underline{F_2}, \underline{F_3}, \underline{F_4}$	F_1F_5, F_1
			2	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5$	F_5
			3	$\underline{F_1}, F_2, \underline{F_4}, F_5$	$F_2F_3F_5, F_2F_5$
			4	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, F_5$	F_3F_5
			5	$\underline{F_1}, F_2, F_3, \underline{F_4}, F_5$	$F_2F_3F_5$
			6	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5$	F_5
			7	$\underline{F_1}, F_2, \underline{F_3}, F_4, F_5$	$F_2F_4F_5$
			8	$F_1, \underline{F_2}, F_3, \underline{F_4}$	$F_1F_3F_5, F_1F_3$
			9	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, F_5$	F_3F_5
			10	$\underline{F_1}, F_2, F_3, \underline{F_4}, F_5$	$F_2F_3F_5$
			11	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, F_5$	F_2F_5
			12	$F_1, \underline{F_2}, \underline{F_3}$	$F_1F_4F_5, F_1F_4, F_1F_5, F_1$
			13	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_5$	F_4F_5, F_5
			14	$\underline{F_1}, F_2, \underline{F_3}, F_4, F_5$	$F_2F_4F_5$
			15	$\underline{F_1}, F_2, F_5$	$F_2F_3F_5, F_2F_5, F_2F_4F_5$
			16	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_4, F_5$	F_4F_5
			17	$F_1, \underline{F_2}, \underline{F_3}, F_4$	$F_1F_4F_5, F_1F_4$
			18	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_4, F_5$	F_4F_5
L, P-23, P-6C	0	0.0602	1	$F_1, \underline{F_2}, \underline{F_3}, \underline{F_4}$	F_1F_5, F_1
			2	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5$	F_5
			3	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, F_5$	F_2F_5
			4	$\underline{F_1}, F_2, F_3, \underline{F_4}, F_5$	$F_2F_3F_5$
			5	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, F_5$	F_3F_5
			6	$\underline{F_1}, \underline{F_2}, \underline{F_3}, \underline{F_4}, F_5$	F_5
			7	$F_1, \underline{F_2}, F_3, \underline{F_4}$	$F_1F_3F_5, F_1F_3$
			8	$\underline{F_1}, F_2, \underline{F_3}, F_4, F_5$	$F_2F_4F_5$
			9	$\underline{F_1}, \underline{F_2}, F_3, \underline{F_4}, F_5$	F_3F_5
			10	$\underline{F_1}, F_2, F_3, \underline{F_4}, F_5$	$F_2F_3F_5$
			11	$F_1, \underline{F_2}, \underline{F_3}$	$F_1F_4F_5, F_1F_4, F_1F_5, F_1$
			12	$\underline{F_1}, F_2, \underline{F_3}, \underline{F_4}, F_5$	F_2F_5
			13	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_4, F_5$	F_4F_5
			14	$\underline{F_1}, F_2, \underline{F_3}, F_4, F_5$	$F_2F_4F_5$
			15	$\underline{F_1}, \underline{F_2}, \underline{F_3}, F_4, F_5$	F_4F_5
			16	$F_1, \underline{F_2}, \underline{F_3}, F_4$	$F_1F_4F_5, F_1F_4$

1. Valves V-2 and V-6 may experience sticking failures.
2. Tank T-1 may leak.

The component models used in this example can be found in Supplementary Material (Part III). The system model and the corresponding diagnoser have been constructed according to these models. The diagnostic performance measures of this system are summarized in Table 9. Notice that failures F_1–F_5 represent *V2SC*, *V2SO*, *V6SC*, *V6SO* and *T2leak* respectively. If only the level sensors are installed in this system, it can be observed that there should be 4 different observable traces and the minimum and maximum uncertainty indices $\overline{H}_{min}$ and $\overline{H}_{max}$ in this situation should be 0.1505 and 0.7341 respectively. After installing an additional flow sensor on pipeline

Please cite this article in press as: Yeh, M.-L., Chang, C.-T., An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes. Chem Eng Res Des (2011), doi:10.1016/j.cherd.2011.05.007

P-23 or P-6C, the ranges of uncertainty index can be changed to [0, 0.1436] and [0.2007, 0.2306] respectively. However, if the additional flow sensors are installed on both P-23 and P-6C, the range of uncertainty index could be further improved to [0, 0.0602].

8. Conclusions and future works

A systematic automata-based procedure is presented in this paper to evaluate and improve the performance of any on-line fault diagnosis scheme for multi-failure scenarios in a batch chemical process. This procedure consists of the following three steps: (1) building a system automaton based on the given process flow diagram and operating procedure; (2) constructing the corresponding diagnoser to determine several qualitative and quantitative performance measures; (3) implementing the proposed design options to enhance the diagnostic performance. The feasibility and effectiveness of this proposed methodology have been positively confirmed with extensive case studies.

As mentioned previously, two specific measures can be adopted to improve diagnostic performance, i.e. (1) identifying and installing additional sensors which are not included in the P&ID and (2) synthesizing and executing extra operation steps which are not provided in the SFC, and they are carried out essentially with brute force in the present study. Future effort should therefore be devoted to the development of a more concrete methodology to efficiently generate such design options.

Acknowledgement

This work is supported by the National Science Council of Taiwan under Grant NSC 98-2221-E-006-045-MY2.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.cherd.2011.05.007.

References

Caccavale, F., Pierri, F., Iamarino, M., Tufano, V., 2009. An integrated approach to fault diagnosis for a class of chemical batch processes. Journal of Process Control 19 (5), 827–841.

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

Cerutti, S., Lamperti, G., Scaroni, M., Zanella, M., Zanni, D., 2007. A diagnostic environment for automaton networks. Software: Practice and Experience 37 (4), 365–415.

Chang, C.T., Chen, J.Y., 2007. Systematic enumeration of fuzzy diagnosis rules for identifying multiple faults in chemical processes. Industrial & Engineering Chemistry Research 46 (11), 3635–3655.

Chang, C.T., Chen, J.Y., 2011. Fault diagnosis with automata generated languages. Computers and Chemical Engineering 35 (2), 329–341.

Chen, J., Jiang, Y.C., 2011. Development of hidden semi-Markov models for diagnosis of multiphase batch operation. Chemical Engineering Science 66 (15), 1087–1099.

Chen, J.Y., Chang, C.T., 2009. Development of fault diagnosis strategies based on qualitative predictions of symptom evolution behaviors. Journal of Process Control 19 (5), 842–858.

Chen, Y.C., Yeh, M.L., Hong, C.L., Chang, C.T., 2010. Petri-net based approach to configure online fault diagnosis systems for batch processes. Industrial & Engineering Chemistry Research 49 (9), 4249–4268.

Chung, S.L., Lai, Y.H., 2008. Process control of brewery plants. Journal of the Chinese Institute of Engineers 31 (1), 127–140.

Fleming, D.W., Pillai, V.A., Pillai, J.A., 1998. S88 Implementation Guide. McGraw-Hill Inc., New York.

Hashizume, S., Yajima, T., Kuwashita, Y., Onogi, K., 2008. Integration of fault analysis and interlock controller synthesis for batch processes. Chinese Journal of Chemical Engineering 16 (1), 57–61.

Kourti, T., Macgregor, J.F., 1995. Process analysis, monitoring and diagnosis, using multivariate projection methods. Chemometrics and Intelligent Laboratory Systems 28 (1), 3–21.

Kourti, T., Nomikos, P., Macgregor, J.F., 1995. Analysis monitoring and fault-diagnosis of batch processes using multiblock and multiway PLS. Journal of Process Control 5 (4), 277–284.

Lafortune, S., Teneketzis, D., 2000. UMDES-LIB, Library of Commands for Discrete Event Systems Modeled by Finite State Machines.

Lai, J.W., Chang, C.T., Hwang, S.H., 2007. Petri-net based binary integer programs for automatic synthesis of batch operating procedures. Industrial & Engineering Chemistry Research 46 (9), 2797–2813.

Lee, J.M., Yoo, C.K., Lee, I.B., 2004. Fault detection of batch processes using multiway kernel principal component analysis. Computers & Chemical Engineering 28 (9), 1837–1847.

Liang, K.H., Chang, C.T., 2008. A simultaneous optimization approach to generate design specifications and maintenance policies for the multi-layer protective systems in chemical processes. Industrial & Engineering Chemistry Research 47 (15), 5543–5555.

Liu, F., Qiu, D., Xing, H., Fan, Z., 2008. Decentralized diagnosis of stochastic discrete event systems. IEEE Transactions on Automatic Control 53 (2), 535–546.

Maurya, M., Rengaswamy, R., Venkatasubramanian, V., 2004. Application of signed digraph-based analysis for fault diagnosis chemical process flowsheets. Engineering Applications of Artificial Intelligence 17 (5), 501–518.

Nomikos, P., MacGregor, J.F., 1994. Monitoring batch processes using multiway principal component analysis. AIChE Journal 40 (8), 1361–1375.

Nomikos, P., MacGregor, J.F., 1995. Multivariate SPC charts for monitoring batch processes. Technometrics 37 (1), 41–59.

Pierri, F., Paviglianiti, G., Caccavale, F., Mattei, M., 2008. Observer-based sensor fault detection and isolation for chemical batch reactors. Engineering Applications of Artificial Intelligence 21 (8), 1204–1206.

Qiu, W.B., Kumar, R., 2006. Decentralized failure diagnosis of discrete event systems. IEEE Transactions on Systems, Man and Cybernetics: Part A. Systems and Humans 36 (2), 384–395.

Qiu, W.B., Kumar, R., 2008. Distributed diagnosis under bounded-delay communication of immediately forwarded local observations. IEEE Transactions on Systems, Man and Cybernetics: Part A. Systems and Humans 38 (3), 628–643.

Rigatos, G.G., 2009. Fault detection and isolation based on fuzzy automata. Information Sciences 179 (12), 1893–1902.

Ruiz, D., Canton, J., Nougues, J.M., Espuna, A., Puigjaner, L., 2001a. On-line fault diagnosis system support for reactive scheduling in multipurpose batch chemical plants. Computers & Chemical Engineering 25 (4–6), 829–837.

Ruiz, D., Nougues, J.M., Calderon, Z., Espuna, A., Puigjaner, L., 2001b. Neural network based framework for fault diagnosis in batch chemical plants. Computers & Chemical Engineering 24 (2–7), 777–784.

Sampath, M., Lafortune, S., Sinnamohideen, K., Teneketzis, D., 1995. Diagnosability of discrete-event systems. IEEE Transactions on Automatic Control 40 (9), 1555–1575.

Sampath, M., Sengupta, R., Lafortune, S., Sinamohideen, K., Teneketzis, D., 1996. Failure diagnosis using discrete-event models. IEEE Transactions on Control Systems Technology 4 (2), 104–105.

Shannon, C.E., 1948. A mathematical theory of communication. The Bell System Technical Journal 27 (379–423), 623–656.

Undey, C., Ertunc, S., Cinar, A., 2003. Online batch fed-batch process performance monitoring, quality prediction, and variable contribution analysis for diagnosis. Industrial & Engineering Chemistry Research 42 (20), 4645–4658.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003a. A review of process fault detection and diagnosis: Part I. Quantitative model based methods. Computers & Chemical Engineering 27 (3), 293–311.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., 2003b. A review of process fault detection and diagnosis: Part II. Qualitative model and search strategies. Computers & Chemical Engineering 27 (3), 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003c. A review of process fault detection and diagnosis: Part III. Process history based methods. Computers & Chemical Engineering 27 (3), 327–346.

Viswanathan, S., Johnsson, C., Venkatasubramanian, V., A¨rzen, K.E., 1998a. Automating operating procedure synthesis for batch processes: Part I. Knowledge representation and planning framework. Computers and Chemical Engineering 22 (11), 1673–1685.

Viswanathan, S., Johnsson, C., Venkatasubramanian, V., A¨rzen, K.E., 1998b. Automating operating procedure synthesis for batch processes: Part II. Implementation and application. Computers & Chemical Engineering 22 (11), 1687–1698.

Viswanathan, S., Shah, N., Venkatasubramanianian, V., 2002. Hybrid framework for hazard identification and assessment in batch processes. AIChE Journal 48 (8), 1765–1774.

Wang, W., Lafortune, S., Girard, A.R., Lin, F., 2010. Optimal sensor activation for diagnosing discrete event systems. Automatica 46 (7), 1165–1175.

Zad, S.H., Kwong, R.H., Wonham, W.M., 2003. Fault diagnosis in discrete-event systems: framework and model reduction. IEEE Transactions on Automatic Control 48 (7), 1199–1204.

Zhang, Z., Wu, C., Zhang, B., Xia, Y., Li, A., 2005. Sdg multiple fault diagnosis by real-time inverse inference. Reliability Engineering and System Safety 87 (2), 173–189.

Zineb, S.A., Maria, D.M., Michal, K., 2010. Fault diagnosis for discrete event systems: modelling and verification. Reliability Engineering and System Safety 95 (4), 369–378.

Please cite this article in press as: Yeh, M.-L., Chang, C.-T., An automaton-based approach to evaluate and improve online diagnosis schemes for multi-failure scenarios in batch chemical processes. Chem Eng Res Des (2011), doi:10.1016/j.cherd.2011.05.007