Computers and Chemical Engineering 97 (2017) 220–241

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Automata-based operating procedure for abnormal situation management in batch processes

Chun-Jung Wang, Yi-Chung Chen, Shih-Ting Feng, Chuei-Tin Chang*

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

ARTICLE INFO

Article history:
Received 29 March 2016
Received in revised form 18 October 2016
Accepted 29 November 2016
Available online 2 December 2016

Keywords:
Automata
Abnormal situation management
Batch processes
Emergency response procedure
Dynamic simulation

ABSTRACT

"Abnormal situation management" (ASM) in general refers to the various tasks required for online fault diagnosis and also hazard mitigation. Although quite a few ASM-related studies have already been carried out in the past, none of them addressed the wide range of issues consistently and rigorously with the same modeling tool. An automata-based strategy is therefore proposed in this work to synthesize all operating procedures needed for diagnostic tests and also other emergency response operations in the batch processes. The proposed model building techniques are suitable not only for characterizing all components in any given process, but also for representing the operation targets of all ASM tasks. Finally, notice that every resulting procedure can be readily expressed with an implementable sequential function chart (SFC).

1. Introduction

Generally speaking, the term "abnormal situation management" (ASM) refers to a collection of distinct tasks that must be performed online in a chemical plant for timely identification and mitigation of any significant departure of the system state from acceptable normal conditions (Bullemer and Nimmo, 1994; Nimmo, 1995). The scope of ASM primarily encompasses fault diagnosis and the subsequent emergency response operations. Yeílamos et al. (2009) tried in a pioneering work to dynamically integrate the conventional techniques for offline hazard analysis into ASM in continuous processes, while fault diagnosis and hazard mitigation in the *batch* plants obviously cannot be handled with the same approach. Although there have been a few related studies discussing various aspects of ASM for the batch processes (Chen et al., 2010; Yeh and Chang, 2011; Li et al., 2014), none of them addressed the wide range of issues consistently and thoroughly. In particular, notice that the diagnostic resolution may be further enhanced via test actions and, also, more flexible emergency response procedures could be synthesized either to maintain a lower but acceptable production rate after confirmation of the abnormal situation(s), or simply to bring the system to a shutdown state safely. A brief literature review of the related works is first presented in the sequel:

Online fault diagnosis of the batch processes has always been a popular research issue. Nomikos and MacGregor (1994, 1995) utilized the multi-way principal component analysis for batch process monitoring, which has later been extended for online diagnosis applications (Lee et al., 2004; Ruiz et al., 2001a,b; Undey et al., 2003; Hashizume et al., 2004; Pierri et al., 2008; Caccavale et al., 2009; Chen and Jiang. 2011). Other AI techniques, such as the artificial immune systems, artificial neural networks and knowledge-based expert systems (Dai and Zhao, 2011; Ghosh and Srinivasan, 2011; Tan et al., 2012; Zhao, 2014), have also utilized for identify fault origins in the batch plants. However, these approaches are mostly effective in systems with relatively few interconnected units and, moreover, the diagnostic resolution in systems with coexisting failures may not always be satisfactory.

To circumvent the above drawbacks, Chen et al. (2010) developed several Petri-net based algorithms to configure fault identification systems for more complex plants. Since the event sequences in multi-failure scenarios cannot be conveniently generated with the Petri-net models, their approach was limited to the single-failure scenarios. On the other hand, the automata were widely adopted as more appropriate models to circumvent this drawback (Debouk et al., 2000; Benveniste et al., 2003; Zad et al., 2003; Qiu and Kumar, 2006; Sköldstam et al., 2007; Malik et al., 2011). Gascard and Simeu-Abazi (2013) utilized the software UPPAAL to build diagnosers with timed automata, while Gomes Cabral et al. (2015) built diagnosers for discrete-event systems modelled with the finite-state automata.

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

http://dx.doi.org/10.1016/j.compchemeng.2016.11.042
0098-1354/

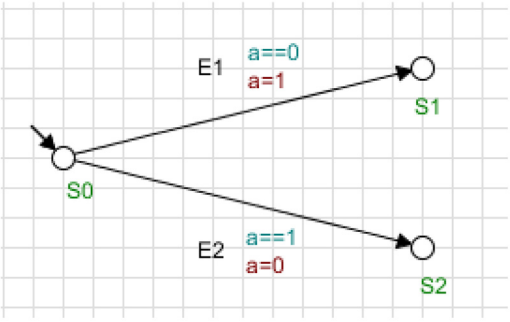


Fig. 1. Example of an EFA model.

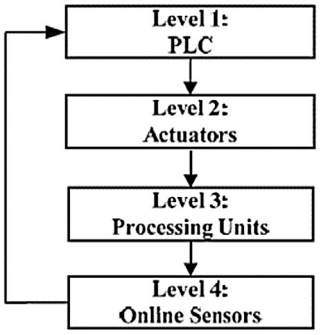


Fig. 2. Hierarchical structure of a batch process.

It should be noted that the aforementioned automata-based approaches preclude the diagnostic tests. Yeh and Chang (2011) developed a trial-and-error design procedure to introduce extra sensors and additional operating procedures into a batch process so as to improve its diagnostic performance, while Kang and Chang (2014) developed a systematic method with DESUMA (Ricker et al., 2006) to search for the optimal diagnostic test plans. It was found in the latter case that, for large systems, the required automata may be quite cumbersome. Finally, note that there have been relatively few published studies on the automata-based synthesis strategies for generating the emergency response procedures, e.g., see Yeh and Chang (2012) and Li et al. (2014).

To develop a consistent and comprehensive approach to ASM, a unified automata-based modeling strategy has been developed in this work to synthesize credible operating procedures needed for diagnostic tests and emergency responses. Specifically, the extended finite automata (EFA) have been adopted to facilitate model building and procedure synthesis with the free software SUPREMICA (Åkesson et al., 2006). The proposed ASM enabling methods can be divided into three groups, i.e., (1) automata building methods, (2) synthesis methods for stipulating the diagnostic test plans, and (3) synthesis methods for generating the emergency response procedures. These methods are presented sequentially in detail as follows.

2. The model building methods

2.1. Extended finite automata

To facilitate a clear description of the proposed model construction method, a brief review of the so-called extended finite automata (EFA) is given here. Let us first consider the standard structure of a deterministic automaton, which can be viewed as a six-tuple as follows:

$$A = (X, E, f, \Sigma, x_0, X_m) \tag{1}$$

where X is the set of system states; E is the event set; $f : X \times E \to X$ represents the state transition function; $\Sigma : X \to 2^E$ denotes the active event function and 2^E is the power set of E; x_0 is the initial system state; $X_m \subset X$ is the set of marked states. The function f can be viewed as a transition process (which is triggered by the feasible event $e \in E$) from state $x \in X$ to another state $x' \in X$, while the active event function Σ of state x is a set of corresponding active events.

The EFA is an improved version of the aforementioned standard structure. It is adopted in this study primarily for the purpose of managing large automata with existing software, e.g., SUPREMICA (Åkesson et al., 2006). To this end, each event in EFA is equipped with two extra auxiliary elements, i.e., *variable* and *guard*. The more specific explanations can be found in the sequel:

- An integer *variable* (with user-specified upper and lower bounds) can be used to update the equipment state after completing an event-driven transition. An example is shown in Fig. 1, in which variable "a" is updated to 1 (a = 1) via event *E1*. In the present study, the variables can be utilized to represent the states of processing units, e.g., the level, temperature or concentration of liquid in a tank.
- A *guard* is the sufficient condition of the corresponding state transition. Let us again consider Fig. 1 as an example and assume that the initial value of variable a is 0. Therefore, only event *E1* is permissible at the initial state S0 due to its guard "a = = 0" and, when S1 is reached after state transition, variable a should be updated to 1.

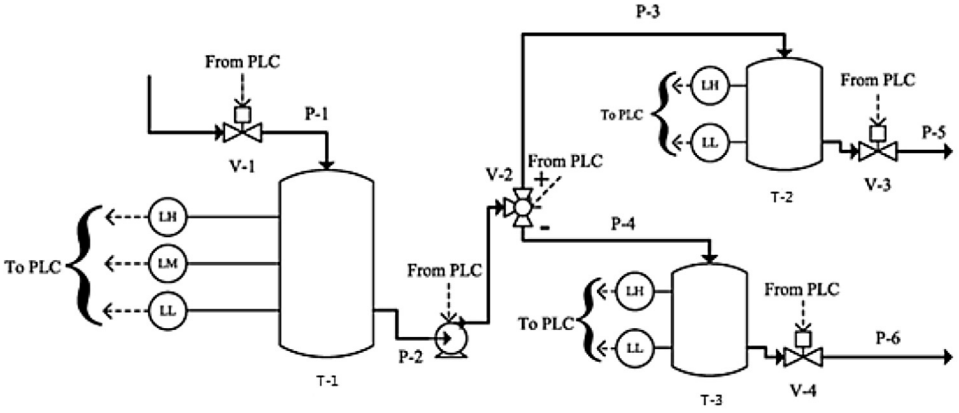


Fig. 3. Three-tank system.

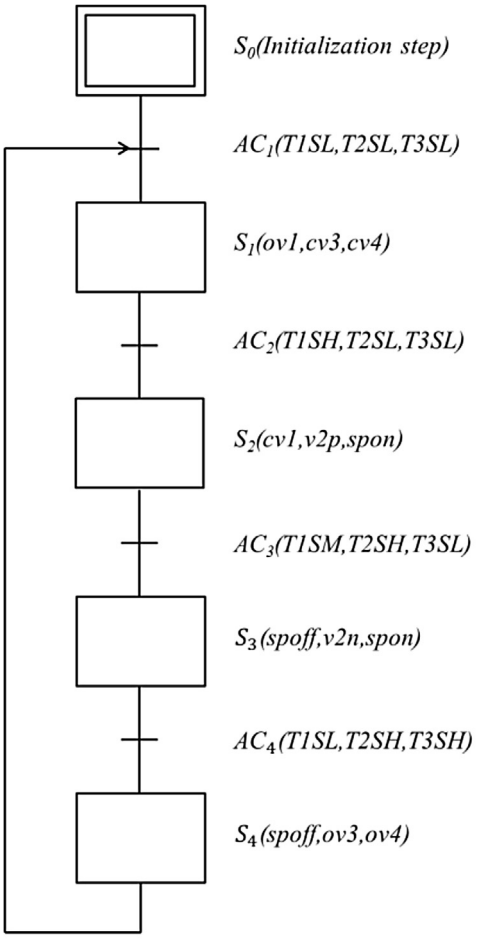


Fig. 4. Sequential function chart of three-tank system.

2.2. Hierarchy structure of batch processes

The components in any given batch process can always be classified into a hierarchy of four different levels (see Fig. 2), i.e., the programmable logic controller (PLC), the actuators, the processing units and the online sensors. To execute the operation procedure specified in a sequential function chart (SFC), the PLC issues control commands that manipulate the states of actuators which, in turn, alter the material and/or energy flows in the batch process. This material-and-energy flow configuration further dictates the operation modes of the embedded units and subsequently changes the corresponding sensor measurements.

Let us consider the three-tank system presented in Figs. 3 and 4 to further illustrate this hierarchical structure. It is assumed that all tanks (T-1, T-2, and T-3) are equipped with level sensors, V-1, V-3, and V-4 are regular gate valves, and V-2 is a three-way valve. The liquid in T-1 is transferred to T-2 when V-2 is placed at position "+", while it is directed to T-3 if V-2 is switched to position "-". The liquid flows in pipelines P-2, P-3, and P-4 are driven by the pump installed on P-2. The normal operation procedure is specified with the SFC given in Fig. 4. On the basis of the above specifications, all components in this example can be grouped in four hierarchical levels, i.e., (1) PLC, (2) V-1, V-2, V-3, V-4, and pump, (3) T-1, T-2, and T-3, and (4) the level sensors on tank T-1, T-2, and T-3. Finally, to facilitate later discussions, the initial conditions of all components in this example are chosen as follows:

- V-1, V-3, and V-3 are closed;
- V-2 is at position "−";
- the liquid levels in all three tanks are low (T1SL, T2SL, and T3SL);
- pump is off.

Table 1
Possible failures in three-tank system.

Symbol	Failure
F_1	scv1
F_2	sov1
F_3	V2nM
F_4	V2pM
F_5	scv3
F_6	sov3
F_7	leakt2

Table 2
Component states and their variable values.

	−1	0	1	2
v1	v1sc	v1c	v1o	v1so
v2		v2c	v2o	
v3	v3sc	v3c	v3o	v3so
v4		v4c	v4o	
t1		t1L	t1M	t1H
t2	t2leak	t2L	t2H	
t3		t3L	t3H	
p		pump on	pump off	

2.3. Diagnoser synthesis procedure

To facilitate fault diagnosis in any given batch process, a so-called "diagnoser" can be synthesized according to the following three steps:

1. Build the component models based on the piping and instrumentation diagram (P&ID).
2. Build the controller model based on the sequential function chart (SFC).
3. Assemble the system model and then synthesize its diagnoser accordingly.

Again, the three-tank system described in Figs. 3 and 4 is utilized here to illustrate the above procedure. For simplicity, let us consider only the failure events listed in Table 1:

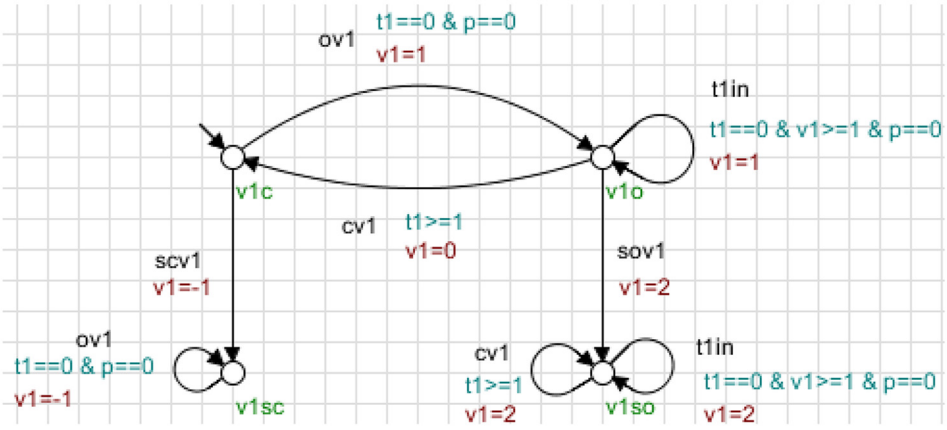
- V-1 and V-3 may stick at the close (scv1 and scv3) or open (sov1 and sov3) position;
- V-2 may be switched to the wrong position due to controller error, i.e., V-2 is mistakenly placed at "−" instead of the correct position "+" (v2 nM), or the other way around (V2pM).
- A leak develops on T-2 (leakt2).

Finally, a variable is adopted to characterize the states of each component in the P&ID. These states and the corresponding variable values are presented in Table 2.

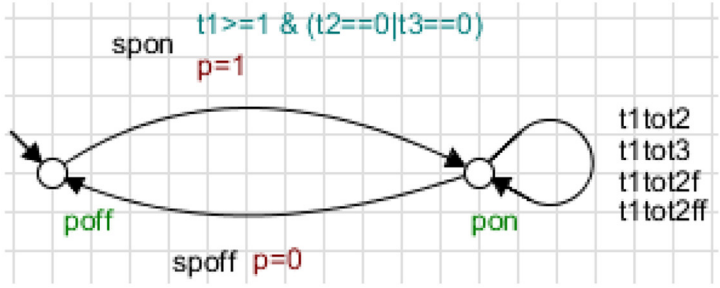
2.3.1. Component models

Every component in the P&ID (which is associated with levels 2–4 of the hierarchical structure) must be modelled with an automaton. The model building principles can be summarized as follows:

All normal and failed states of the component should be first enumerated and represented with places in the automaton. The initial state should be selected and the corresponding place marked with a directed arc without inputs. All normal and failure events that facilitate state transitions should then be identified and each described with a directed arc between two places. Finally, the guard



(a) V-1.



(b) Pump.

Fig. 5. Actuator models of three-tank system.

of every event and the resulting variable value should be added on the corresponding arc.

The components in the aforementioned three-tank system can be modelled with these principles. The resulting automata are presented in the sequel:

(1) Level 2:

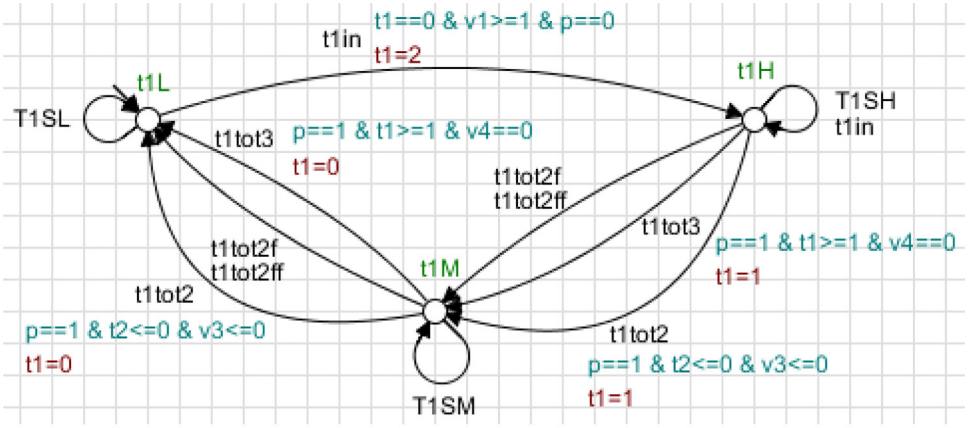
Due to space limitation, only the component models of valve V-1 and pump are shown in Fig. 5,

- Let us first consider valve V-1. There are two normal states in this model, i.e., v1c and v1o, representing the close and open positions respectively, and also two events that activate the transitions between those states, i.e., ov1 and cv1. Variable v1 denotes the state of valve V-1. Its value is updated to either 0 or 1 after event cv1 or ov1 takes place. Notice that ov1 is equipped with a guard ($t1 == 0 \& p == 0$) to impose the constraint that V-1 can only open when level of tank T-1 is low and pump is off. The guard ($t1 >= 1$) of event cv1 represent valve V-1 can only be closed when the level in T-1 is not low. On the other hand, the event t1in (filling tank T-1) can be activated while T-1 is not at low, V-1 is open (v1o), and pump is closed, i.e., the guard "t1 == 0& v1 >= 1& p == 0". The failures scv1 and sov1 update the value of v1 to −1 and 2, respectively. The self-looping event on the failed state scv1 and scv2 (i.e., ov1 and cv1) is adopted to delineate the fact that V-1 cannot be manipulated in this condition. Please notice that filling tank T-1 fill-in can also take place when v1so.
- Let us next consider the pump model. There are two normal states, i.e., on (pon) and off (poff). Note that the guard of spon (switching on the pump) is expressed as "t1 >= 1&(t1 == 0|t3 == 0)", which implies that the level in T-1 must be at the middle or high and also that the level in either T-2 or T-3 must be low. When the pump stays on (pon), the self-looping events t1tot2 and t1tot3 represent the scenarios that the liquid in T-1 can be transferred to T-2 and T-3, respectively, during normal operation. On the other hand, the liquid in T-1 may also be driven to T-2 by operating this pump without raising the liquid level in T-2 when V-3 sticks at the open position (sov3) or when a leak develops in T-2 (leakt2). These two additional failure-induced scenarios can only be fully described by including the corresponding self-looping events, i.e., t1tot2f and t1tot2ff, on the place pon.

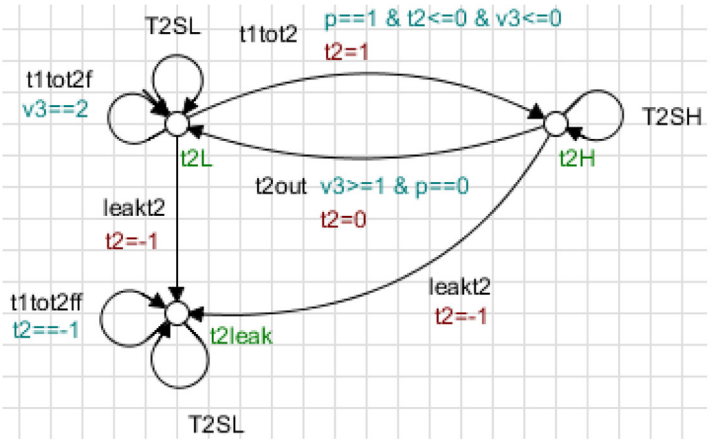
(2) Level 3:

The component models for all processing units in the three-tank system are shown in Fig. 6. These models are briefly described below:

- The state of T-1 is characterized in its component model according to the liquid level only. Three places are included in the corresponding automaton, i.e., t1H, t1M and t1L, to represent the high, intermediate and low levels respectively under normal conditions. The low liquid level in T-1 (t1L) can be raised to high (t1H) by feeding T-1 (t1in) and this high level can be lowered to t1M and then to T1L by transferring liquid from T-1 to either T-2 (t1tot2, t1tot2f, and t1tot2ff) or T-3 (t1tot3). Finally, the self-



(a) T-1.



(b) T-2.

Fig. 6. Process models of three-tank system.

looping events on t1H, t1M and t1L, i.e., T1SL, T1SM and T1SH, denote the corresponding measurement-taking actions respectively.

- Both T-2 and T-3 are modelled with two places representing distinct normal levels, i.e., high (t2H) and low (t2L). The connecting transitions are adopted to describe basically the same events as those used for T-1. Since the leakage of T-2 is considered in the present example, an addition failed state (t2leak) is introduced into the corresponding automaton in Fig. 6(b). On the other hand, since it is still possible to feed liquid into a leaking T-2, the self-looping event on t2leak, i.e., t1tot2ff with guard $t2 == -1$, indicates that the liquid level in T-2 always stays at low value in this scenario. Similarly, the self-looping event t1tot2f (with guard $v3 == 2$) describes the same effect on level in T-2 if V-3 sticks at the open position (sov3). The normal model of tank T-3 is essentially the same as that of T-2 and, thus, not shown because of space limitation.

(3) Level 4:

Without loss of generality, the sensor failure is excluded in this example for the sake of simplicity. In other words, it is assumed that the level measurements are always accurate. The sensor failures will be discussed in other examples later in this paper.

2.3.2. Controller model

The normal controller is modelled to mimic the given SFC closely. Specifically, its operation actions (denoted by rectangles) and activation conditions (denoted with horizontal bars) are all treated as events and their precedence order in automaton should be exactly the same as that in SFC. If a periodic operation is under consideration, this event sequence naturally forms a loop. Note that a cyclic automaton can indeed be constructed (see Fig. 7) according to the SFC shown in Fig. 4.

It should also be noted that, in addition to the cyclic event sequence mentioned above, the extra branches and self-looping events in Fig. 7 are used to model the effects of various failures. To be more specific, the component failures in four levels of the hierarchical structure may exert distinct influences on the controller actions and thus the corresponding fault propagation mechanisms must be modelled differently as follows:

1. Since a controller malfunction is viewed as a mistakenly performed actuator action in this work, this failure can be modelled with a branch attached to the normal event sequence at the place where the correct action is active. In the present example, since v2nM and v2pM represent erroneous operations of the 3-way valve V-2, the corresponding branches are introduced respectively before the normal actions v2p (S4) and v2n (S9). It is also assumed that the subsequent controller actions should still fol-

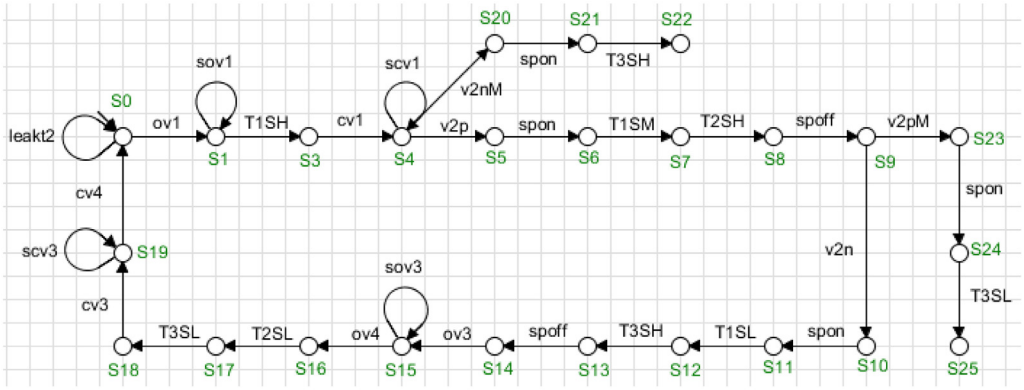


Fig. 7. Controller model of three-tank system.

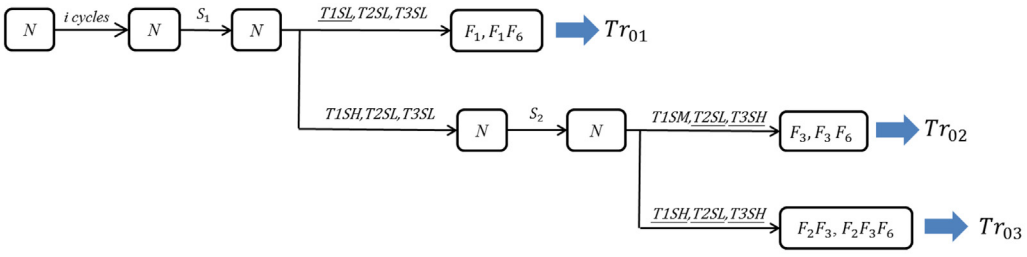


Fig. 8. Undiagnosable OETs in three-tank system.

low the given SFC. Therefore, in each of the above two scenarios, the pump is switched on immediately after controller malfunction but the resulting sensor measurement fails to satisfy the required activation condition.

2. Since an actuator failure is assumed to be unrecoverable in this work, the corresponding failed state is always trapped in a deadlock place in the component model, i.e., see Fig. 5(a) and (c). To describe the impact of such a failure on controller, a self-looping event can be added at the place immediately after the corresponding actuator action in normal operation. For example, the self-looping event sov1 is attached to S1 in Fig. 7 since this failure (V-1 sticks at the open position) can only occur after ov1 (opening V-1). As a second example, notice that sov3 is located at S15 since it is a realizable failure only after ov3 (opening V-3).
3. The failure of any processing unit can be represented with a self-looping event at the initial place in the controller model. Since such failures can happen anytime, they are introduced in the beginning of each operation cycle to simplify the resulting diagnoser. Notice that, in Fig. 7, a self-looping event leakt2 (i.e., a leak develops in T-2) is located at S0.
4. The sensor failure is also represented with a self-looping event located after the measurement taking event. For example, if a new failure sLt2s (i.e., the level sensor on T-2 is stuck at the low value) is considered in the three-tank system, the corresponding event should be introduced at S17.

2.3.3. Diagnoser

Although any number of hardware items may fail while operating a chemical plant, these failures are usually unobservable. Fault diagnosis is a common ASM task that can be performed to identity the root causes of abnormal conditions based on the available online information. It is assumed in this work that the observable events are limited to those associated with actuator actions and sensor measurements. Based on this assumption, the standard operation of parallel composition can be applied to integrate all automata mentioned previously so as to synthesize a diagnoser (Cassandras and Lafortune, 1999). Let us again consider the three-tank system for illustration convenience. The aforementioned component and controller models can be synchronized in SUPREMICA and the fault origins of all observable event traces (OETs) in the resulting diagnoser are listed in Table 3. Notice first that the definitions of all failures (i.e., F_1–F_7) in this table have already been given in Table 1. If a collection of failures results in an OET, they are referred to as a "fault origin" in this work. Notice also that an OET may be caused by more than one fault origins and, in Table 3, they are separated by comma(s) in each row.

Table 3
Fault origins of observable event traces in diagnoser.

Traces	Fault origins
1	F_1,F_1F_6
2	F_3,F_3F_6
3	F_2F_3,$F_2F_3F_6$
4	F_1
5	F_3
6	F_2F_3
7	F_2
8	F_5
9	F_4
10	F_2F_6
11	F_7
12	F_6

Clearly a diagnoser is useful for online fault diagnosis. Observing any of the single-origin scenarios, i.e., traces 4–12 in Table 3, essentially implies that the corresponding failure(s) is present. However, traces 1–3 are still undiagnosable. To facilitate further analysis, these undiagnosable traces are shown in Fig. 8 and labelled as Tr_{01}, Tr_{02}, and Tr_{03}, respectively. A rectangle on any of these traces denotes a system state, i.e., it is either normal (N) or failed due to the implied fault origins, while every transition a set of observable events. The initial transition *i cycles* ($i \geq 0$) represents the normal batch operation may have been performed periodically for i times. S_1 and S_2 denote the controller actions defined in Fig. 4. The remain-

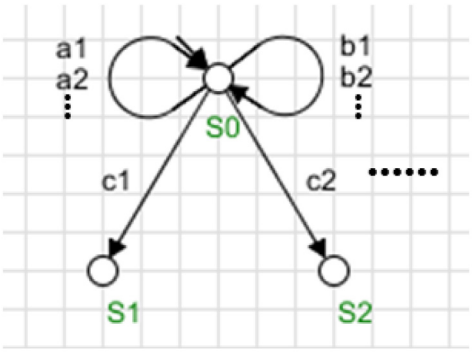


Fig. 9. Single-layer structure of auxiliary automaton *A*.

ing observable events in these traces are sensor measurements which are explained below:

- Tr_{01}: After completing *i* cycles, the steps S_1 (*ov1*, *cv3*, and *cv4*) in the original SFC is then performed. The subsequent normal sensor readings should be T1SH, T2SL and T3SL, but instead T1SL, T2SL and T3SL are obtained. There are two possible fault origins in this scenario: (1) F_1 and (2) F_1F_6.
- Tr_{02}: The initial path (*i cycles* $\rightarrow S_1$) is identical to that of Tr_{01}, while the sensor readings after S_1 are normal, i.e., the same as those given in SFC. After subsequently performing the operation steps in S_2 (*cv1*, *v2p*, and *spon*), the resulting sensor measurements are T1SM, T2SL and T3SH. This is abnormal since, according to the given SFC in Fig. 4, the expected sensor responses to S_2 should be T1SM, T2SH and T3SL. The two possible fault origins in this scenario are (1) F_3 and (2) F_3F_6.
- Tr_{03}: The path before S_2 on this trace is the same as that on Tr_{02}(i.e., *i cycles* $\rightarrow S_1 \rightarrow$ T1SH,T2SL,T3SL $\rightarrow S_2$), while a different set of sensor readings, i.e., T1SH, T2SL and T3SH, appear afterwards. The two possible fault origins in this case are (1) F_2F_3 and (2) $F_2F_3F_6$.

3. Diagnostic tests

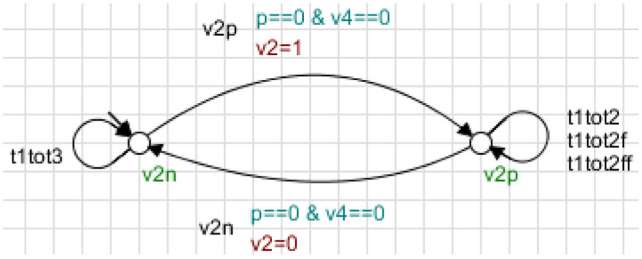
In addition to fault diagnosis, two types of ASM-facilitating operations can also be considered in advance. Firstly, if an OET of the diagnoser is undiagnosable, a test plan can be devised to enhance diagnostic resolution by implementing extra actuator actions. Secondly, with a given OET and its fault origin(s), the emergency response procedure(s) can be synthesized to steer the batch system away from hazardous conditions while still maintain an acceptable production rate or simply to bring it to the shutdown condition safely.

Detailed descriptions of the required procedure synthesis strategies for the former operation are presented below:

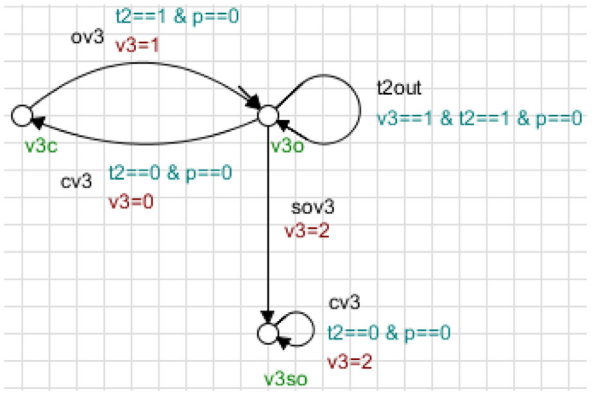
3.1. Test target

In order to enhance the diagnosibility of the above observable event traces, an auxiliary automaton *A* (see Fig. 9) should be built to facilitate the search for the test plan of each trace to generate different system states (Kang and Chang, 2014). In this automaton, the self-looping event "*a1, a2...*" and "*b1, b2...*", and the transition event "*c1, c2...*" are defined as follows

- a1, a2...: all events concerning executing actuator actions;
- b1, b2...: all events concerning taking sensor readings;
- c1, c2...: all events concerning reaching possible combinations of system states (except that of the initial conditions).



(a) V-2.



(b) V-3.

Fig. 10. Modified actuator models for generating the test plan of Tr_{02}.

3.2. Test plans

After marking all terminal states in automaton *A*, the test plan of trace Tr_i can be synthesized as follows:

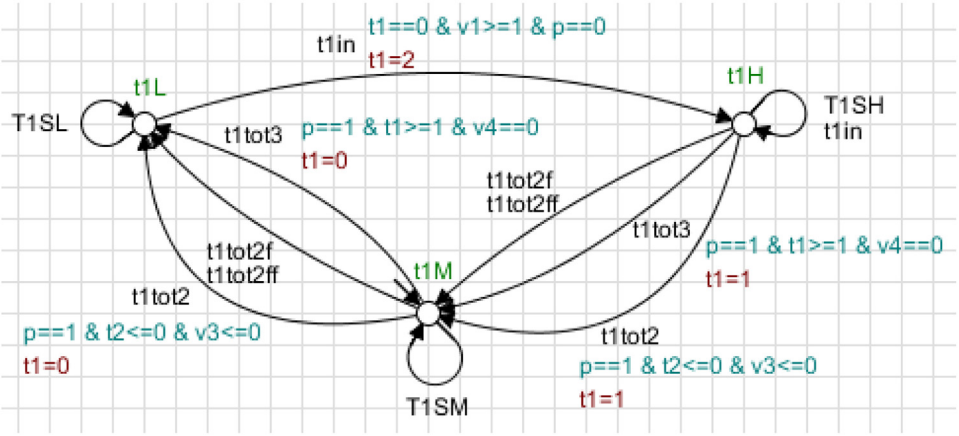
(1) Remove the normal model of SFC.
(2) Set the initial state of each component as the last state on the observable event trace Tr_i with the following steps:
   a. If a failure can be confirmed, the abnormal state of the corresponding component is set as the initial state.
   b. If a component is normal in trace Tr_i, the initial state is set as the last normal state in trace Tr_i.
   c. If the component state is uncertain, the initial state should be set at the state prior to the failure.
(3) Modify the model and initial state of each component. Specifically, the initial state of each component needs to be reset and remove the failure state that does not exist in trace Tr_i.
(4) Generate the auxiliary automaton *A*.
(5) Perform the "Synchronize" function in SUPREMICA and search for paths leading to the marked states.

3.3. An illustrative example of test-plan synthesis

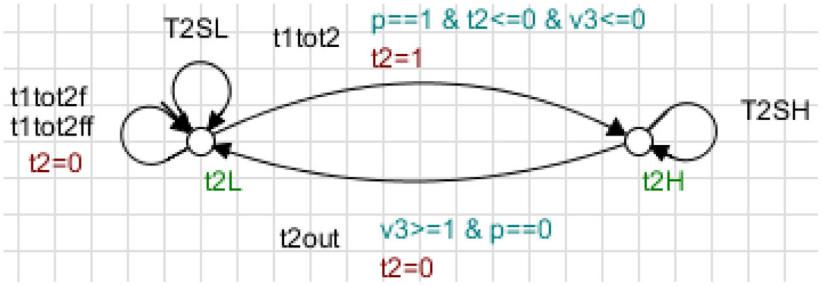
For illustration purpose, the above test-plan synthesis procedure can be applied to all OETs in Fig. 8:

3.3.1. Test plan of Tr_{01}

No feasible test plans can be generated. Since fault origins of this OET are F_1 and F_1F_6, failure F_1 (scv1) should be present in either scenario. From the facts that both T-1 and T-2 are empty after Tr_{01}



(a) T-1.



(b) T-2.

Fig. 11. Modified tank models for generating the test plan of Tr_{02}.

is fully developed and also T-1 cannot be refilled again by opening V-1, one can deduce that it is not possible to adjust the liquid level in T-2 (due to lacking water source) for the purpose of proving or disproving the existence of F_6 (sov3).

3.3.2. Test plan of Tr_{02}

The fault origins of this trace are F_3 (v2nM) and F_3F_6 (v2nM and sov3). The test plan can be obtained with the following procedure:

(1) Remove the controller model in Fig. 7.
(2) Determine the initial state of each component according to Tr_{02}:
- V-1 (v1c): Since none of the failures concerning V-1 appear in Tr_{02}, this valve should be closed after completing S2.
- V-2 (v2n): Since F_3 (v2nM) can be confirmed, V-2 should be at position "–".
- V-3 (v3o): From the given fault origins, one cannot be certain if V-3 is closed normally (v3c) or stuck at the open position (v3so). Based on step (2)cin the previous subsection, the initial state of V-3 should be set at the state prior to the failure F6(sov3), i.e., v3o.
- V-4 (v4c): Since the failures of V-4 are not considered in this example (see Table 1), this valve naturally should be at the close position after completing S2.
- Pump (pon): Since pump failures are not considered either (see Table 1), the pump should be on after completing S2.
- Tanks (t1M, t2L, t3H): These are the final states observed on trace Tr_{02}.

(3) Modify the component models according to step 3 in Section 3.2. Since the component models of V-1, V-4, pump, and T-3 are unchanged, let us use valves V-2, V-3, tank T-1 and tank T-2 as examples to illustrate the model modification step (see Figs. 10 and 11):
- V-2: Since v2nM (F_3) and v2pM (F_4) are controller failures, they are not included in this component model.
- V-3: Because failure F_6 (v3so) may or may not be present, the initial state is placed at v3o according to step (2) above. Consequently, both scenarios can be considered in synthesizing the test plan. Note that, from this initial state, V-3 may be either transferred to v3c via actuator action cv3 or stuck at v3so via failure sov3.
- T-1: According to step (2), set the initial state to t1M.
- T-2: Set the initial state to t2L and remove failure F_7 (leakt2).

(4) Construct auxiliary automaton *A*.

Since the number of fault origins is 2 (F_3 and F_3F_6) in this case, the upper and lower bounds of the layer number should be both 1 and, thus, automaton *A* has only one layer. The resulting single-layer auxiliary automaton is presented in Fig. 12. The self-looping events on S0 are all possible actuator-moving and measurement-taking actions, while events r1–r4 and r6–r12 represent the transitions to new sensor readings (other than their initial values of t1M, t2L and t3H).

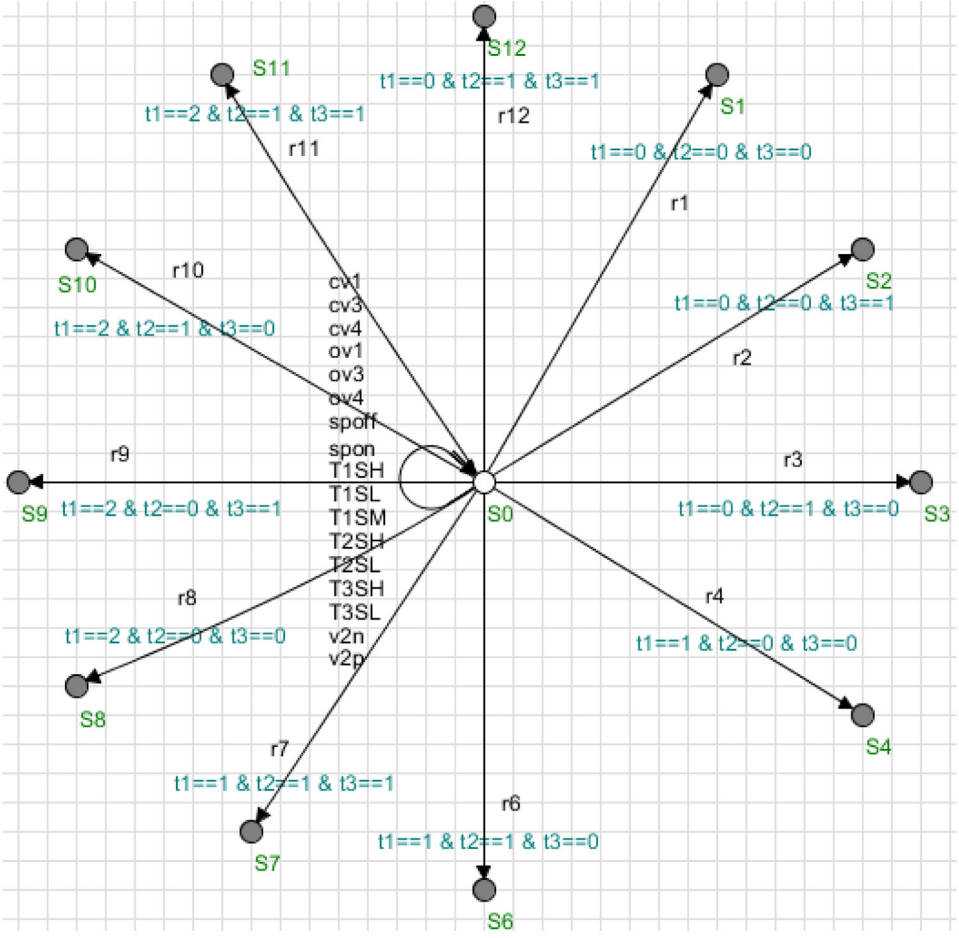
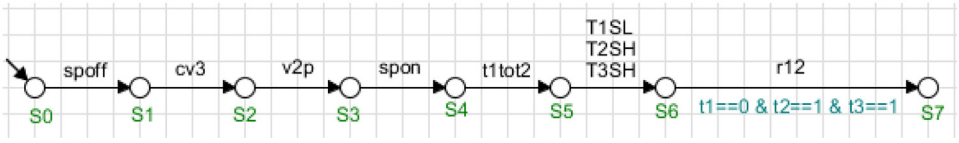
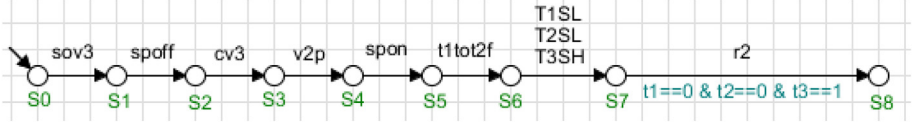


Fig. 12. Auxiliary automaton A for Tr_{02}.



(a) F_3.



(a) F_3F_6.

Fig. 13. Event traces for Tr_{02} after executing diagnostic test.

(5) Search for event sequences leading to the marked places in automaton A.

After synchronizing all aforementioned models, two event traces (including unobservable events) can be found and they are both presented in Fig. 13. Notice that, after executing the common actuator actions, i.e., spoff, cv3, v2p, and spon, these two scenarios end up with different sensor readings. The trace with fault origins F_3 in Fig. 13(a) reaches a new state via transition r12, while the one with F_3F_6 in Fig. 13(b) reaches another state via transition r2. A sequential functional chart of the diagnostic test required for Tr_{02} can then be obtained by combining those two traces (see Fig. 14). The complete test plan can be summarized as follows:

i When the trace Tr_{02} is observed online, it can be confirmed that V-2 has been switched mistakenly to the "−" position (F_3). However, the condition of V-3 is uncertain (V-3 may or may not be stuck at the open position). Thus, a diagnostic test is performed (by executing poff, cv3, v2p, and pon) to check if V-3 is stuck at the open position.

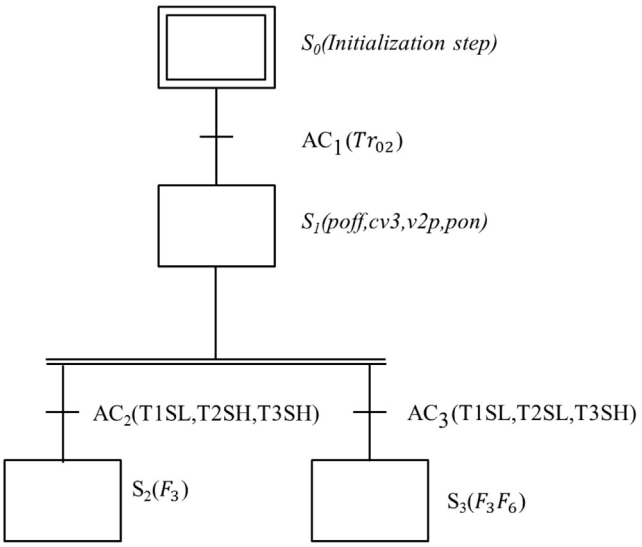


Fig. 14. SFC of diagnostic test for Tr_{02}.

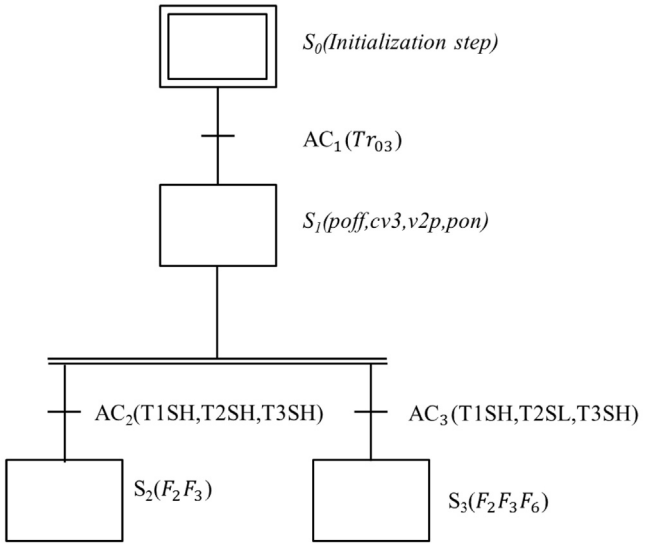


Fig. 16. SFC of diagnostic test for Tr_{03}.

ii If the resulting sensor readings are T1SL, T2SH and T3SH, fault origin should consist only the controller error F_3 (v2nM) and valve V-3 should be working.

iii If the resulting sensor readings are T1SL, T2SL and T3SH, then both failures, i.e., F_3 (v2nM) and F_6 (sov3), should be present.

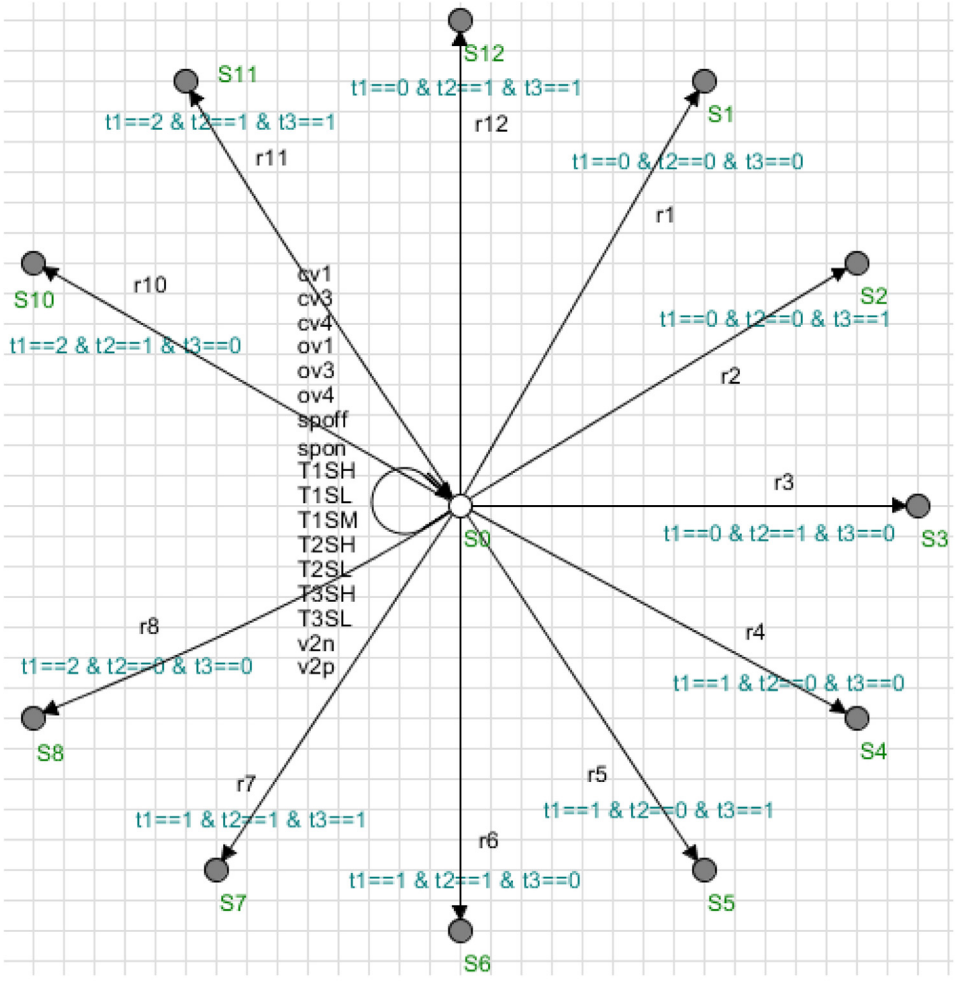


Fig. 15. Auxiliary automaton A for Tr_{03}.

3.3.3. Test plan for Tr_{03}

As mentioned before, the two fault origins in this scenario can be expressed as F_2F_3 (sov1 and v2nM) and $F_2F_3F_6$ (sov1, v2nM, and v3so). Since the number of fault origins is 2 (F_2F_3 and $F_2F_3F_6$) in this case, the auxiliary automaton A has only one layer. The resulting single-layer auxiliary automaton is presented in Fig. 15. After synchronizing the required automata, two event traces can be found and a sequential functional chart of the corresponding diagnostic tests can then be obtained accordingly (see Fig. 16).

4. Emergency response operations

As mentioned previously, a set of proper operation steps can be synthesized and then implemented on the basis of an OET in the diagnoser (obtained with or without the diagnostic test) to steer the batch system away from hazardous conditions while still maintain an acceptable production rate or simply to bring it to the shutdown condition safely.

The required procedure synthesis steps are outlined below:

4.1. Operational goals

After the diagnostic tests, the operational goals of emergency responses may be expressed with the following auxiliary automata:

- In order to maintain an acceptable production rate, auxiliary automaton B (see Fig. 17) can be constructed to facilitate search for the steps to drive the system to satisfy one of the activation conditions in the original SFC.
- If the above search is successful, another auxiliary automaton C (see Fig. 18) should be built to identify a temporary SFC so as to continue production at a lower rate. If this search fails, then a third auxiliary automaton D (see Fig. 19) should be adopted to stipulate the controller actions (which are usually concerned with emptying all processing vessels and switching off all actuators) so as to bring the system to a shutdown state safely.

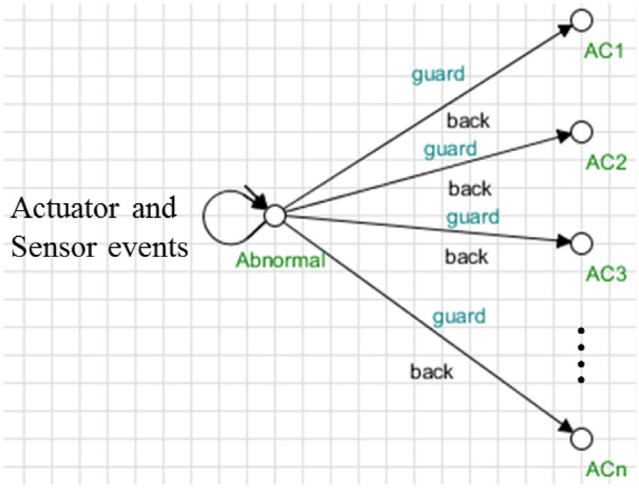


Fig. 17. Auxiliary automaton *B*.

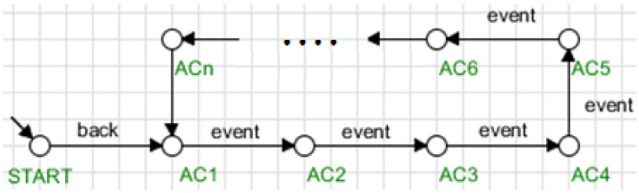


Fig. 18. Auxiliary automaton *C*.

4.2. Response procedures

Seven steps are required to generate the emergency response procedures. The first three steps are identical to those used for test-plan synthesis (Section 3.2), except that all potential failures in the given OET should be treated as events with 100% certainty. For example, if fault origins F_1, F_1F_2, and F_1F_3 are implied by an OET, all

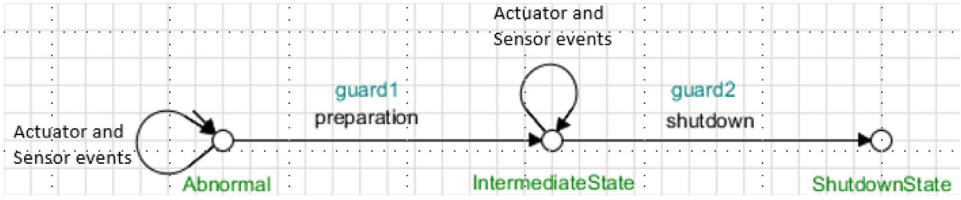


Fig. 19. Auxiliary automaton *D*.

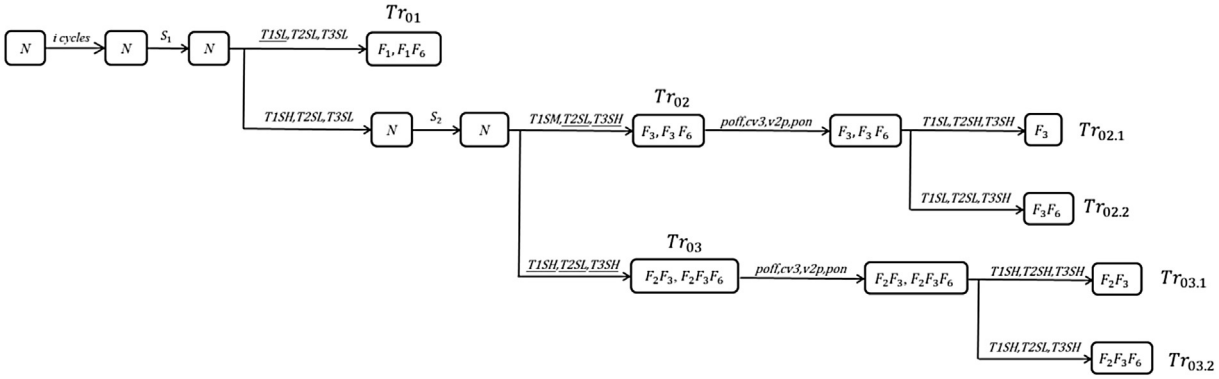
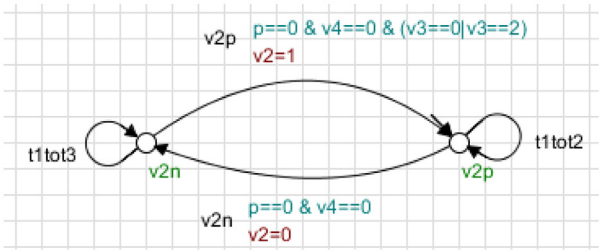
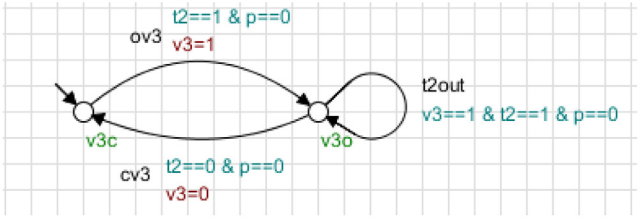


Fig. 20. OETs of three-tank system after diagnostic tests.



(a) V-2.



(b) V-3.

Fig. 21. Modified actuator models for generating response operations of $Tr_{02.1}$.

failures, i.e., F_1, F_2 and F_3, should be included in the corresponding component models.

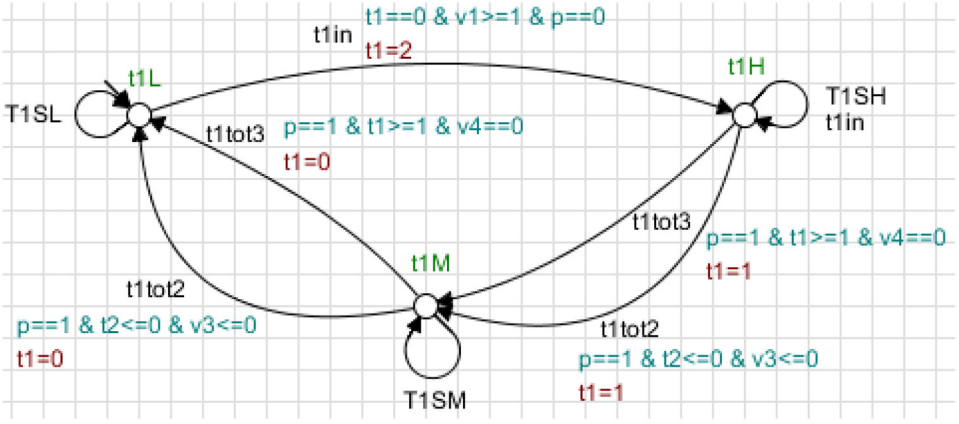
The remaining four steps are described in the sequel:

(4) Construct the auxiliary automaton B according to Fig. 17. The self-looping events on the *abnormal* state should be all possible actuator-moving and measurement-taking actions. Note that there are n transitions in this automaton from the abnormal state to the marked normal states AC_i^N $(i = 1, 2, \cdots, n)$, and each is associated with the same event *back*. The different activation conditions in the original SFC are adopted as the alternative *guards* of such an event.

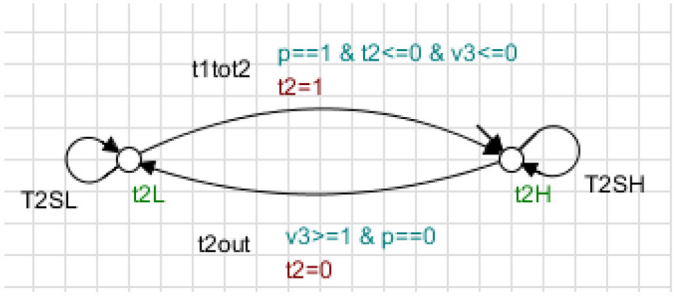
(5) Perform the "Synchronize" function in SUPREMICA to search for the paths leading to marked states. The system can be driven back to the normal condition if any of the marked states can be reached.

(6) If it is possible to identify one or more path leading to a normal condition, then search for the operating procedure to maintain production as follows:
   a. Modify automaton B by removing the infeasible marked states, and then add all possible actuator-moving and measurement-taking actions as the self-looping events on the identified state also.
   b. Build automaton C to incorporate all or at least a maximum portion of the activation conditions in the original SFC. The initial state START in this automaton connects to the normal activation condition found in step (5). For example, if condition AC_1^N is identified, then the automaton in Fig. 18 should be used for procedure synthesis.
   c. Synchronize all modified component models, the modified automaton B and the just-constructed automaton C. The emergency response procedure to maintain production can be obtained if there is a path leading to the marked state.

(7) If no paths can be found in the previous step, auxiliary automaton D (see Fig. 19) should be built to generate the shutdown procedure. The operational goal(s) can be achieved as long as there is a path leading to state *ShutdownState*, while the state *IntermediateState* is optional in this automaton. The



(a) T-1.



(b) T-2.

Fig. 22. Modified tank model for generating response operations of $Tr_{02.1}$.

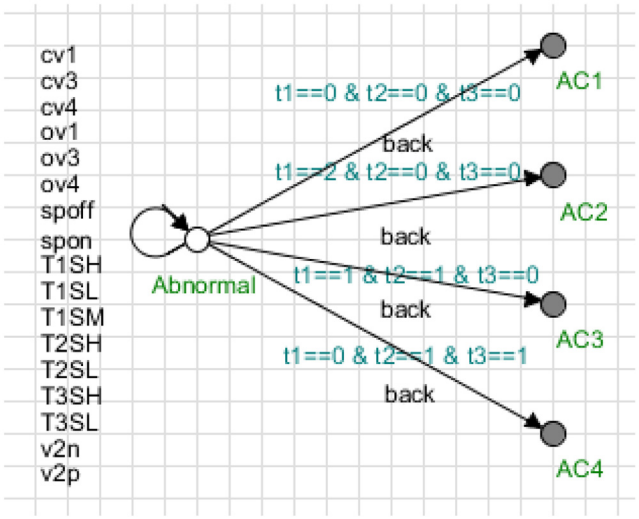


Fig. 23. Auxiliary automaton *B* for $Tr_{02.1}$.

guards on the corresponding transitions, i.e., *guard1* and *guard2*, are the prerequisite conditions of preparation and shutdown, respectively. The shutdown procedure can then be obtain by synchronizing automaton *D* and modified component models.

4.3. An illustrative example of response-procedure synthesis

After completing all diagnostic tests in the three-tank system, the resulting OETs can be expressed as Fig. 20. Trace Tr_{01} is undiagnosable and therefore cannot be developed further, while Tr_{02} and Tr_{03} can both be extended and each evolved into two sub-traces by performing the controller actions specified in its test plan. The response-procedure synthesis strategy described in Section 4.2 can be applied on each of these sub-traces.

4.3.1. Response procedure for $Tr_{02.1}$

(1) Remove the controller model in Fig. 7.
(2) Determine the initial state of each component according to $Tr_{02.1}$. After observing trace Tr_{02}, a diagnostic test (poff, cv3, v2p, and pon) was performed and the sensor reading (T1SL, T2SH, and T3SH) indicates the fault origin is F_3 (v2nM). Thus, the final component states implied by $Tr_{02.1}$, i.e., v1c, v2p, v3c, v4c, pon, t1L, t2H, and t3H, should be used as the initial states for response-procedure synthesis.
(3) Modify the actuator models in Fig. 10 by removing failure sov3 and changing the initial states of V-2 and V-3 to v2p and v3c, respectively (see Fig. 21). The component models of V-1 and V-4 should remain unchanged. Modify the tank models in Fig. 11 by changing the initial states of T-1 and T-2 to t1L and t2H respectively (see Fig. 22).
(4) Build auxiliary automaton *B* (see Fig. 23) based on the activation conditions in original SFC (see Fig. 4).
(5) Perform the "Synchronize" function in SUPREMICA and search for paths leading to any of the marked states. A direct transition from the abnormal state to the marked state AC_4^N can be found since the sensor readings at the end of $Tr_{02.1}$ already satisfy the corresponding activation conditions, i.e., T1SL, T2SH and T3SH.
(6) Search for the emergency response procedure(s):
   a. Modify automaton *B* by removing states AC_1^N–AC_3^N from Fig. 23 and also adding all possible actuator-moving and measurement-taking actions on the self-looping events of stateAC_4^N. The result is shown in Fig. 24.
   b. Build auxiliary automaton *C* (see Fig. 25) based on the original SFC (see Fig. 4), and mark the last state *S6* of the longest path in the cycle.
   c. Synchronize the modified component models, the modified automaton *B* and automaton *C*, and then search for path(s) leading to the marked state. The controller model and the SFC for executing the response procedure that maintains production can be obtained on the basis of the only feasible path identified in this case (see Fig. 26).

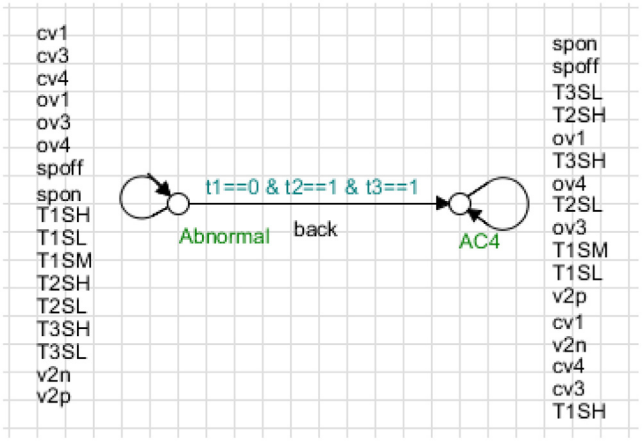


Fig. 24. Modified auxiliary automaton *B* for $Tr_{02.1}$.

In summary, since the final sensor readings on trace $Tr_{02.1}$ already satisfy the activation conditions specified in AC_4^N, there are no needs for additional controller actions to bring system back to normal. The normal production rate can then be maintained according to following steps: (1) switch off the pump and then open V-3 and V-4 to empty T-2 and T-3; (2) close V-3 and V-4 and then open V-1 to fill T-1 after the liquids in both T-2 and T-3 reach low levels (i.e., T2SL and T3SL); (3) resume the normal production cycle.

4.3.2. Response procedure for $Tr_{02.2}$

(1) Remove the controller model in Fig. 7.
(2) Determine the initial state of each component according to $Tr_{02.2}$. After observing trace Tr_{02}, a diagnostic test (poff, cv3, v2p, and pon) can be performed and the sensor readings (T1SL, T2SL, and T3SH) indicate that F_3 (v2nM) and F_6 (sov3) are both present. Thus, the final component states implied by $Tr_{02.2}$, i.e., v1c, v2p, v3so, v4c, pon, t1L, t2L, and t3H, should be used as the initial states for response-procedure synthesis.

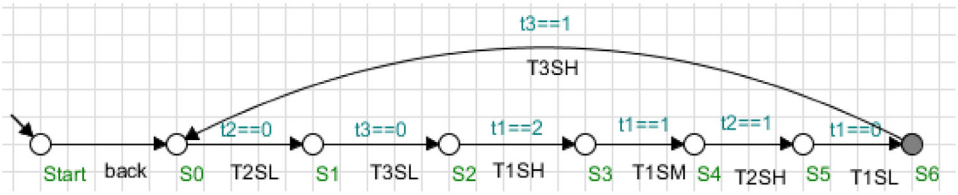
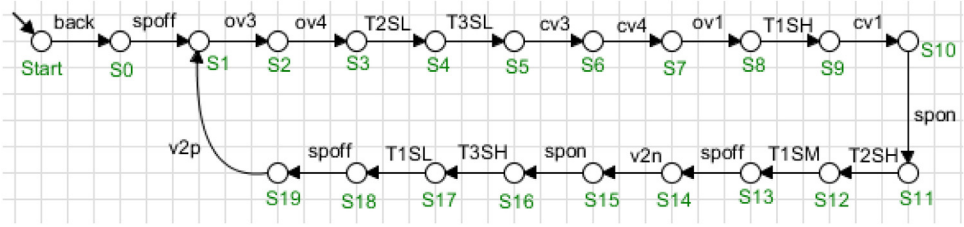
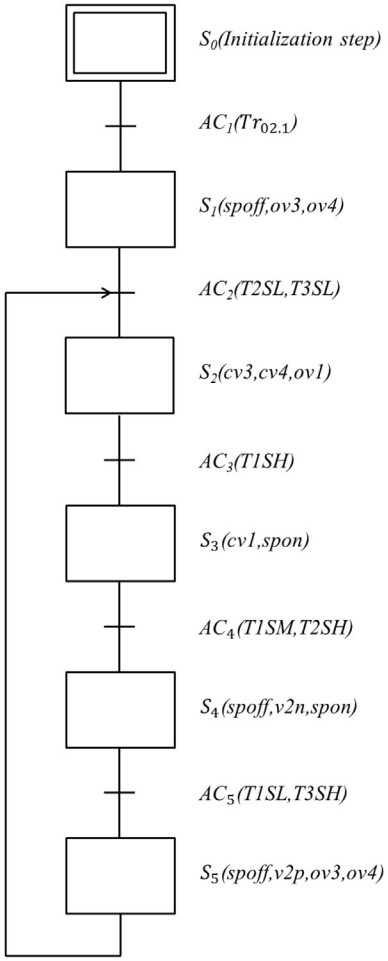


Fig. 25. Auxiliary automaton *C* for $Tr_{02.1}$.



(a) The controller model for $Tr_{02.1}$.



(b) SFC for $Tr_{02.1}$.

Fig. 26. Emergency response procedure to maintain a lower production rate after $Tr_{02.1}$.

(3) Modify the actuator model of V-2 in Fig. 10 by changing its initial state to v2p (see Fig. 27). The component models of V-1 and V-4 are the same as before, while that of V-3 can be discarded since it should always stay at the failed state v3so. Next, modify the tank models in Fig. 11 by changing the initial states of T-1 and T-2 to t1L and t2L respectively (see Fig. 28).
(4) Build automaton *B* (see Fig. 23, identical to $Tr_{02.1}$) based on the activation conditions in original SFC (see Fig. 4).
(5) Perform the "Synchronize" function in SUPREMICA and search for paths leading to any of the marked states. The path leading to AC_1^N can be found (see Fig. 29).
   a. Modify automaton *B* by removing states AC_2^N–AC_4^N from Fig. 23 and also adding all possible actuator-moving and measurement-taking actions on the self-looping events of stateAC_1^N. The result is shown in Fig. 30.
   b. Build automaton *C* (see Fig. 31) based on the original SFC (see Fig. 4), and mark the last state *S7* of the longest path in the cycle.
   c. Synchronize the modified component models, the modified automaton *B* and auxiliary automaton *C*, and then search for path(s) leading to the marked state. No marked states can be reached in this situation due to fact that V-3 is stuck at open position. Since V-3 is located at the outlet of T-2, automaton *C* should be modified to neglect the corresponding activation conditions in the original SFC, i.e., only the sensor readings of T-1 and/or T-3 can be considered (see Fig. 32). The resulting

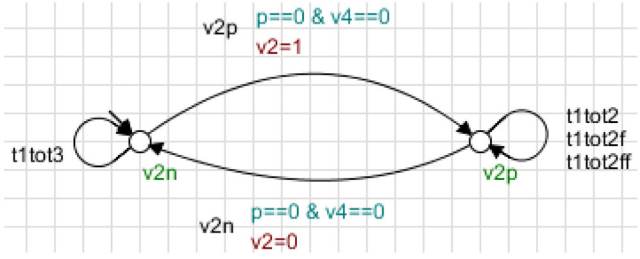
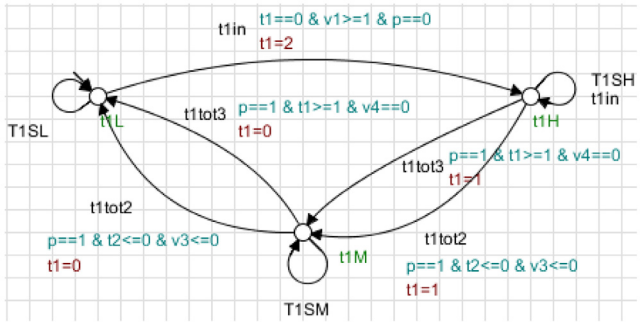
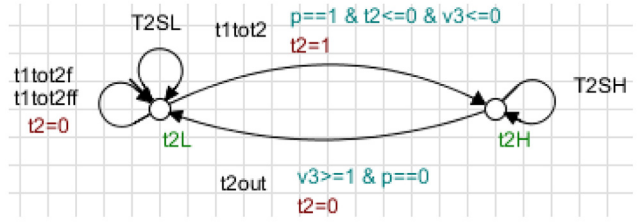


Fig. 27. Modified the actuator model of valve V-2 for generating response operations of $Tr_{02.2}$.



(a) T-1.



(b) T-2.

Fig. 28. Modified tank models for generating response operations of $Tr_{02.2}$.

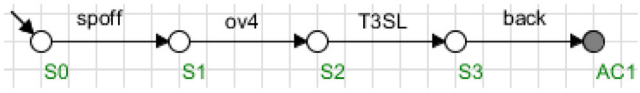


Fig. 29. The path leading to normal operation of $Tr_{02.2}$.

search was successful, and the controller model and the SFC for executing the response procedure that maintains production can be obtained accordingly (see Fig. 33).

(6) Search for the emergency response procedure(s):

In summary, after observing trace $Tr_{02.2}$, the system can be driven to normal activation condition AC_1^N by turning off pump,

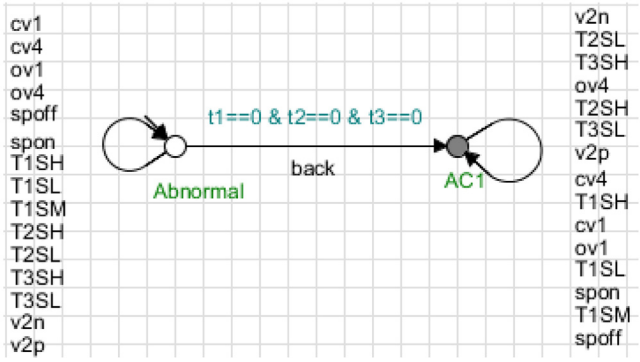


Fig. 30. Modified auxiliary automaton B for $Tr_{02.2}$.

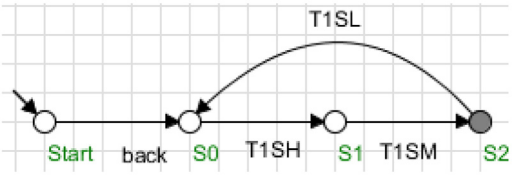


Fig. 32. Modified auxiliary automaton C for $Tr_{02.2}$.

opening V-4, and then waiting for the liquid level in T-3 drop to a low value (T3SL). Then a lower production rate can be achieved by operating T-1 and T-3 only.

4.3.3. Response procedure for $Tr_{03.1}$

Since this trace is utilized primarily to demonstrate the synthesis steps needed for generating the shutdown procedure, i.e., step (7) in subsection 4.2, the details of first six steps are omitted. Since both F_2 (v1so) and F_3 (v2nM) are confirmed by trace $Tr_{03.1}$, a response procedure can be synthesized to keep the production rate at a lower level (see Fig. 34). However, from the fact that the liquid in T-1 always stays at the high level (because V-1 sticks at the open position), one may conclude that it is hazardous to operate under such conditions. To address this concern, a shutdown procedure can be generated as follows:

In this example, let us assume that the intermediate state in Fig. 19 is not needed, and all actuators must be switched off and all tanks emptied at the shutdown state. Consequently, the automaton D in Fig. 35 can be adopted to generate the shutdown procedure. Note that, due to failure F_2 (v1so), the guard of transition *shutdown* does not include the conditions for closing V-1 and emptying T-1. The resulting controller model and shutdown procedure are shown in Fig. 36.

4.3.4. Response procedure for $Tr_{03.2}$

Since the corresponding procedure synthesis steps are very similar to those used for $Tr_{02.2}$, only the required SFC is presented here in Fig. 37. Notice also that both F_2 (sov1) and F_6 (sov3) can be confirmed in this scenario and, consequently, the liquid levels in T-1 and T-2 should always be high (T1SH) and low (T2SL) respectively

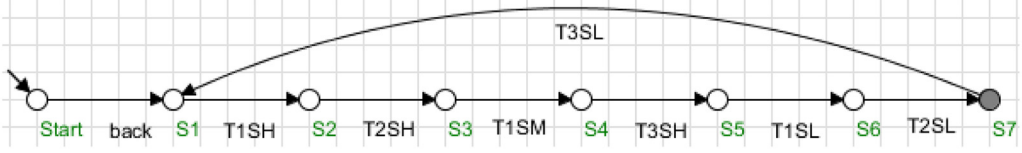
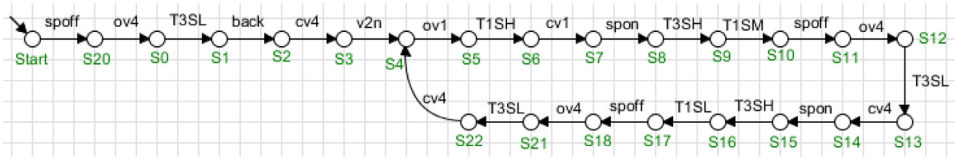
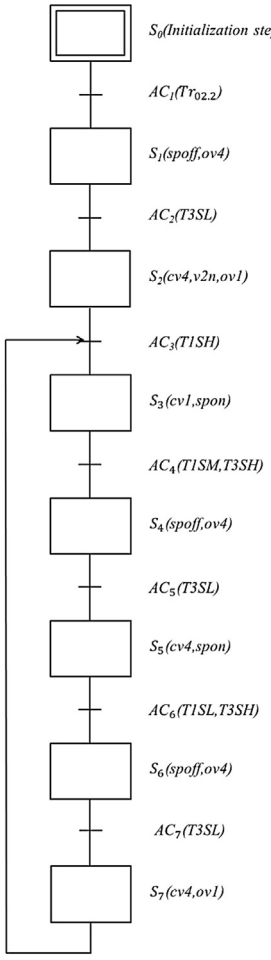


Fig. 31. Auxiliary automaton C for $Tr_{02.2}$.



(a) Controller model.



(b) SFC.

Fig. 33. Emergency response procedure to maintain a lower production rate after $Tr_{02.2}$.

after the above failures take place. As a result, only the level reading on T-3 is usable in the emergency response procedure. Specifically, after observing trace $Tr_{03.2}$, the system can be driven to normal activation condition AC_2^N by turning off pump, opening V-4, and then waiting for the low-level reading in T-3 (T3SL). A lower production rate can then be realized by operating T-3 only.

5. Case studies

The above test-plan and response-procedure synthesis strategies have been applied to a realistic process, i.e., batch evaporation and the results of extensive case studies are briefly summarized below:

5.1. Process description

Let us consider the batch evaporation system presented in Fig. 38 (Bauer et al., 2004). Evaporator T1 in this system is operated with a heater (H1) and a condenser (C1), while the buffer vessel T2 another heater (H2). The liquid flows in the inlet pipeline of T1, in the pipeline between T1 and T2, in the cooling-water line of C1 and in the outlet pipeline of T2 are facilitated with the gate valves V1, V2, V4 and pump P2, respectively, and V3 is only a normally-closed spare of V2. Various online sensors are also installed to monitor the operating conditions of T1, T2 and C1, i.e.,

a. Evaporator (T1) is equipped with the level, temperature and concentration sensors, which can be used respectively for detecting:
   i. low level (T1LSL), intermediate level (T1LSM), and high level (T1LSH),

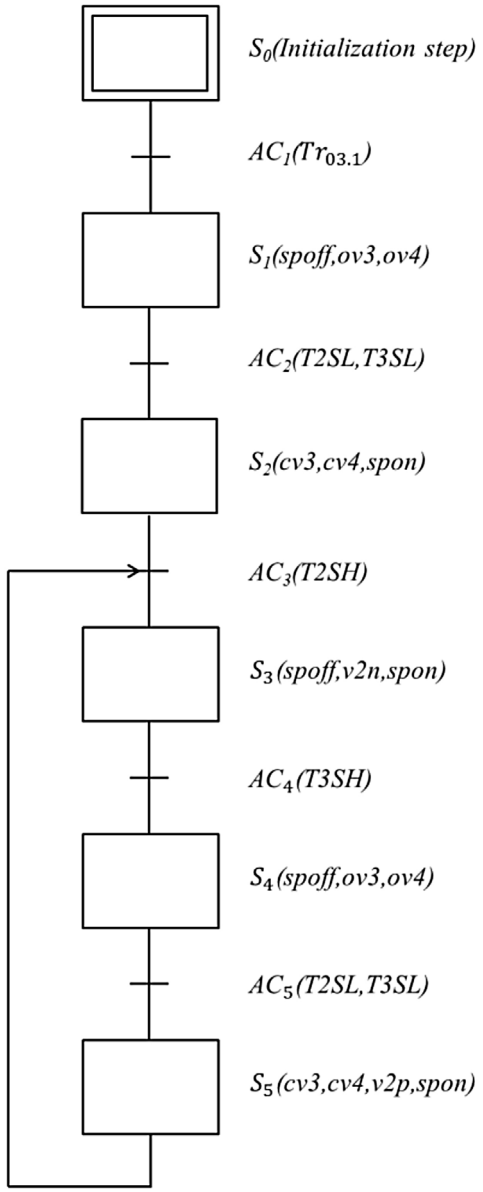


Fig. 34. Emergency response operations to maintain production of $Tr_{03.1}$.

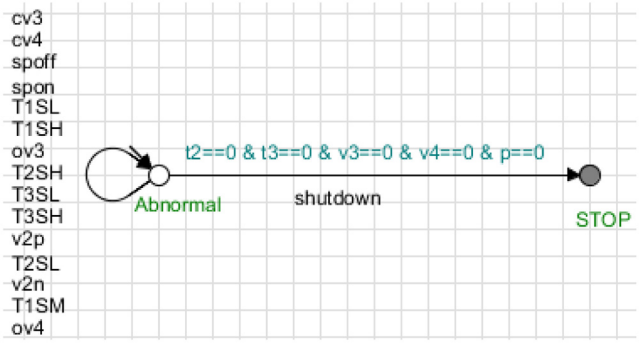


Fig. 35. Auxiliary automaton D of $Tr_{03.1}$.

ii. low temperature (T1TSL) and high temperature (T1TSH),
iii. low concentration (T1QSL) and high concentration (T1QSH);

b. Buffer vessel (T2) is equipped with the level and temperature sensors, which are used respectively to detect:
i. low level (T2LSL) and high level (T2LSH),
ii. low temperature (T2TSL) and high temperature (T2TSH);

c. Condenser (C1) is equipped with a flow sensor to monitor the cooling-medium flow, i.e., on or off.

The normal operating procedure of this system is specified with a SFC, which can be found in Fig. 39. Let us assume that, initially, all actuators are closed (or switched off) and all vessels are empty. To facilitate illustration clarity, each step of this procedure is further described as follows:

(S1) Fill T1 by opening V1 (ov1) when all sensor readings are low.

(S2) When the sensor reading of T1 reaches T1LSH, close V1 (cv1), open V4 (ov4), and turn on the heater H1 (H1on) to start evaporating the liquid in T1.

(S3) After the online sensors on T1 show T1LSM, T1TSH, and T1QSH, switch off H1 (H1off), open V2 (ov2), and close V4 (cv4) to stop evaporation and then start transportation of liquid from T1 to T2.

(S4) After the online sensors on T1 and T2 show T1LSL, T2LSH, and T2TSH, close V2 (cv2) to isolate T2, turn on H2 (H2on) to maintain the temperature of T2, and open V1 (ov1) to refill T1.

(S5) After observing T1LSH and T2TSH from the online sensors on T1 and T2, close V1 (cv1) and turn on H1 (H1on) to restart the evaporation process and, then, turn off H2 (H2off), open V4 (ov4) and switch on P1 (pon) to discharge T2.

(S6) After the level sensor on T2 shows T2LSL, switch off P1 and return to step S3 and repeat the operation cycle from S3 to S6.

Finally, let us further assume that there are five possible failure events, i.e., F_1 (T1Qfailure): the concentration sensor on T1 is broken, F_2 (scv4): V4 sticks at the close position, F_3 (H1failure): H1 is out of order, F_4 (T2leak): a leak develops in T2 causing a constantly-low liquid level, and F_5 (scv2): V2 sticks at the close position.

5.2. Diagnoser

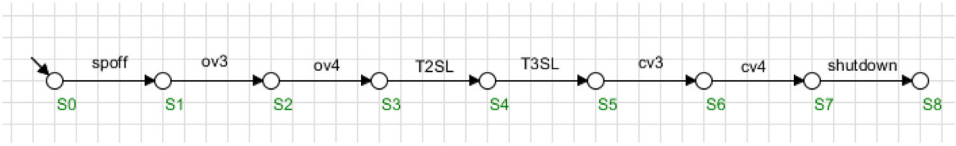
The component and controller models in this example are omitted for the sake of brevity, while all OETs in diagnoser can be found in Fig. 40. Since only trace Tr_{05} is diagnosable, i.e., it confirms the presence of fault origin F_4, additional diagnostic tests should be applied to Tr_{01}–Tr_{04}.

5.3. Test plans

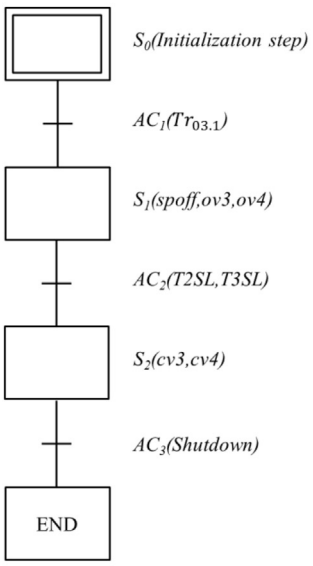
Figs. 41–44 respectively show the OETs established by applying diagnostic tests to Tr_{01}–Tr_{04}. Note that these traces are essentially self-explanatory and, thus, their test plans can be extracted accordingly in a straightforward fashion. Specifically, the rectangles are used to represent system states and each is characterized with the potential fault origins. Any transition between two consecutive states corresponds to either a collection of controller actions or a set of sensor readings. Note also that these two types of events always appear alternately in any OET and they can be used in the corresponding SFC as the operation steps and activation conditions respectively.

5.4. Response procedures

Since it is required to evaporate a large portion of solvent from the feed so as to obtain a high-concentration product, the operational goals of response procedures in almost all scenarios are set to be safe shutdown. For example, note that failures F_3 (H1failure) and F_5 (scv2) can be confirmed in trace $Tr_{01.3}$ (see Fig. 41) and, thus, it is clear that the evaporation operation cannot be performed any more without the essential heating capability provided by H1. As a result,



(a) The controller model.



(b) SFC

Fig. 36. Emergency shutdown procedure of $Tr_{03.1}$.

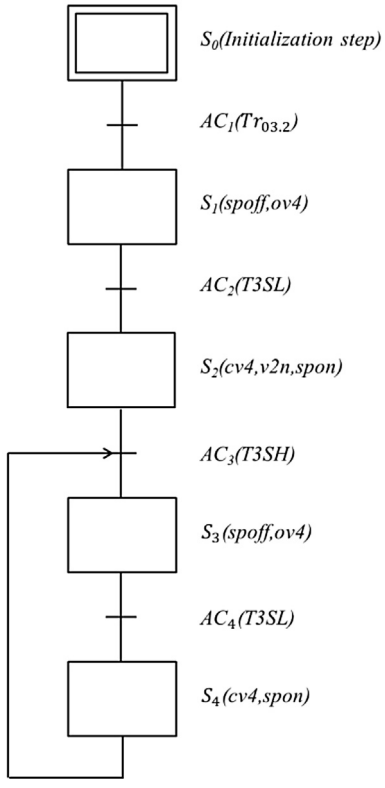


Fig. 37. Emergency response procedure to maintain a lower production rate after $Tr_{03.2}$.

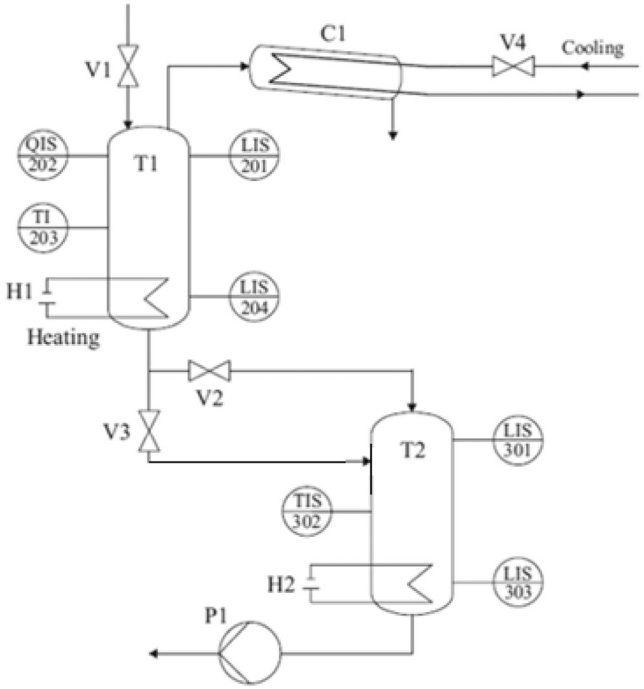


Fig. 38. P&ID of a batch evaporation system.

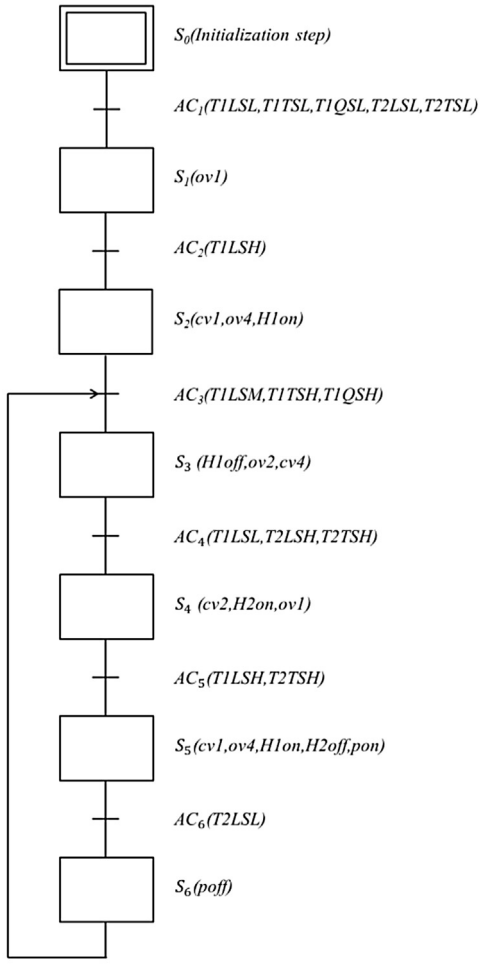


Fig. 39. The normal operating procedure of batch evaporation process.

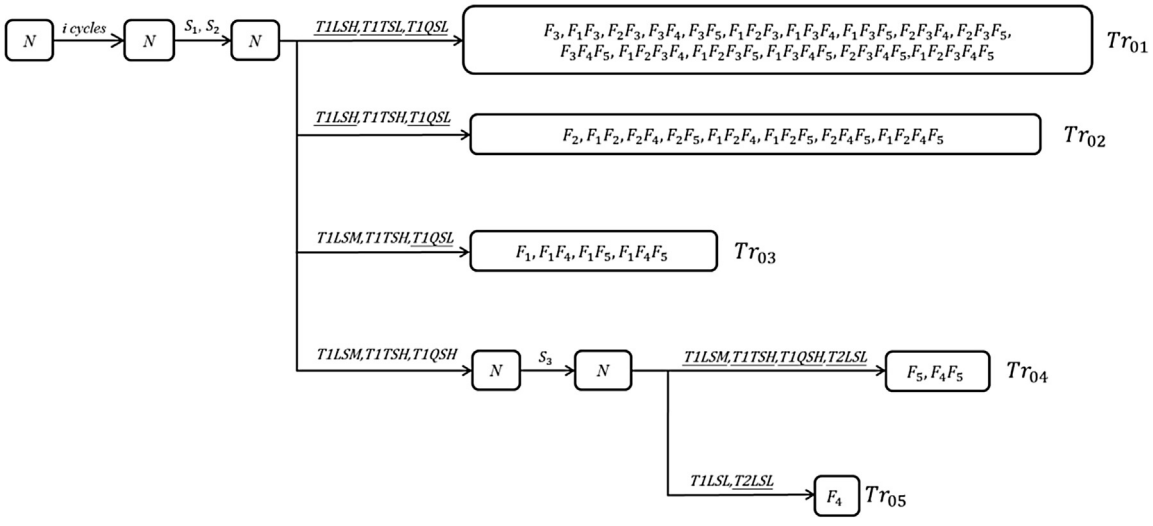


Fig. 40. All OETs in diagnoser of batch evaporation process.

the shutdown procedure in Fig. 45 should be carried out. Specifically, this procedure calls for closing V3 and switching on pump immediately after observing trace $Tr_{01.3}$ to discharge the liquid in T2. As soon as T2 is empty (T2LSL), the shutdown operation can be terminated by turning off the pump. On the other hand, notice that the response procedure after observing $Tr_{04.1}$ is used to maintain a lower production rate (see Fig. 46). Since F_5 (scv2) is confirmed in this case, the liquid in T1 can be transferred T2 via an alternative route by operating V3 instead.

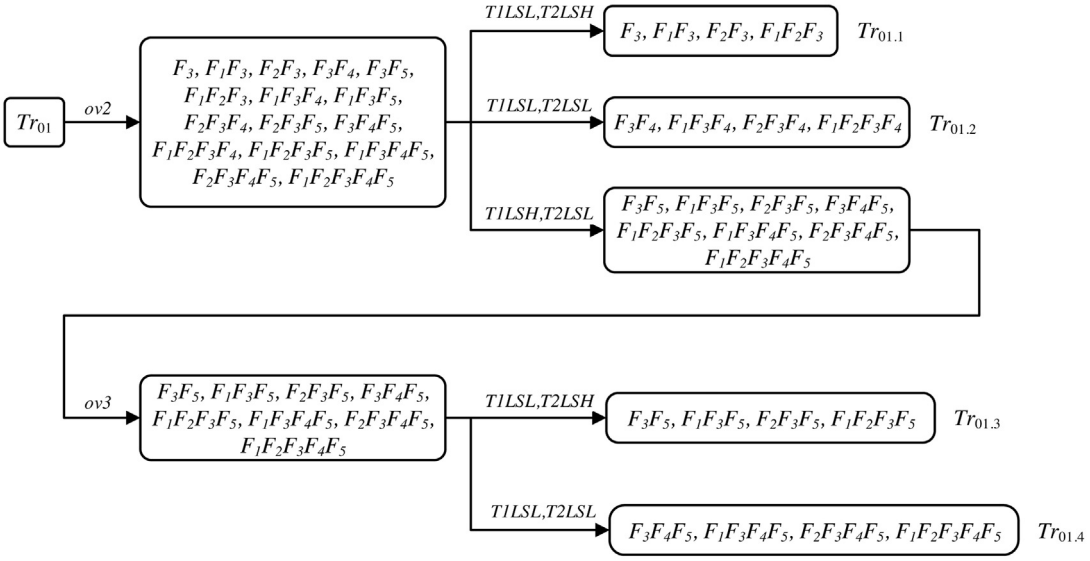


Fig. 41. OETs of Tr_{01} obtained after diagnostic tests.

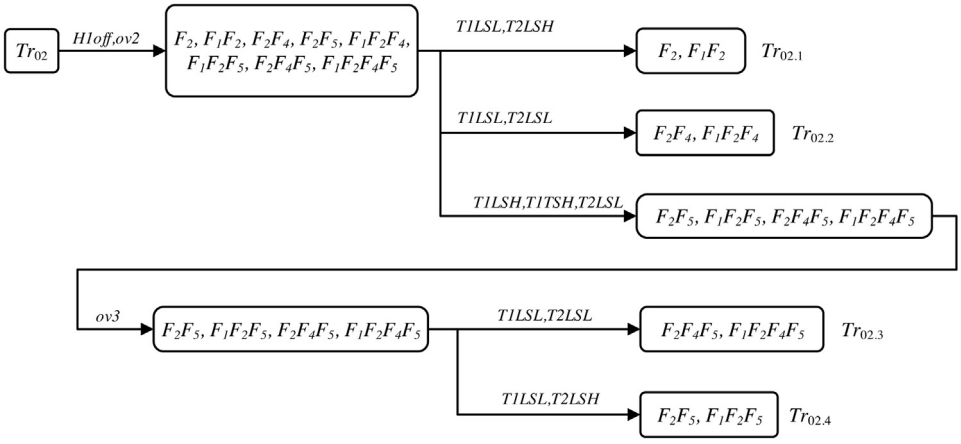


Fig. 42. OETs of Tr_{02} obtained after diagnostic tests.

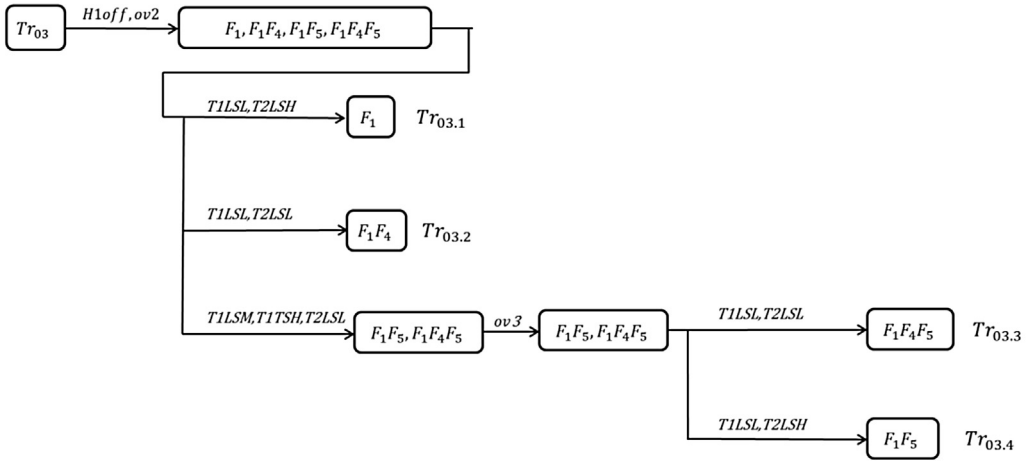


Fig. 43. OETs of Tr_{03} obtained after diagnostic tests.

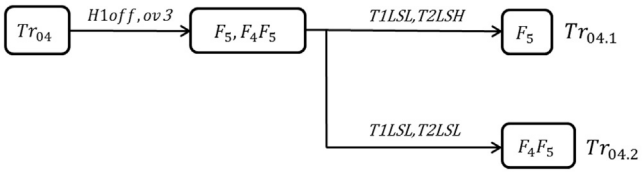


Fig. 44. OETs of Tr_{04} after diagnostic tests.

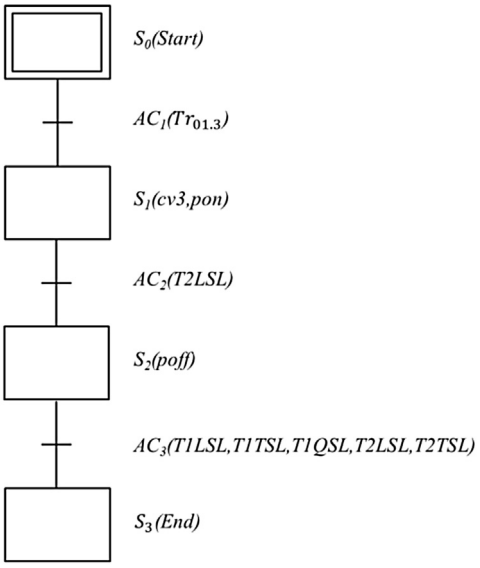


Fig. 45. Shutdown procedure after observing $Tr_{01.3}$.

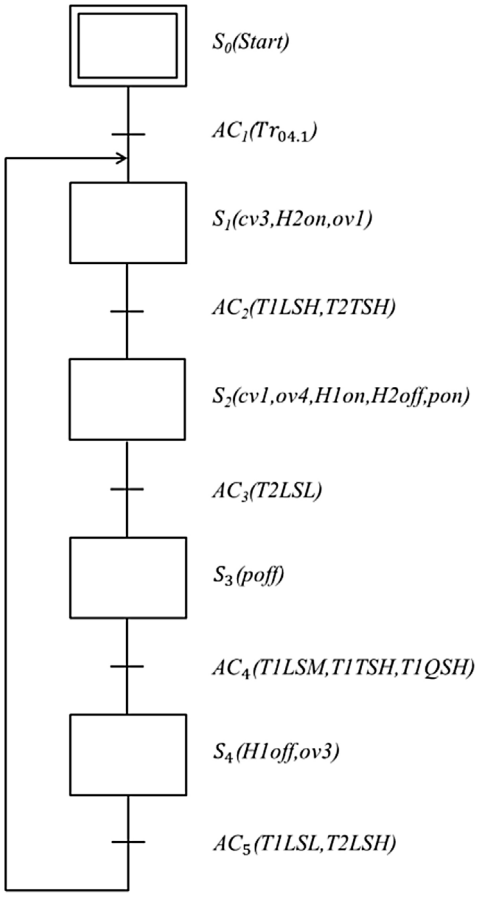


Fig. 46. Emergency response procedure after observing $Tr_{04.1}$.

6. Conclusions

In this research, the standardized automata-building methods have been developed to perform various tasks for abnormal situation management in the batch processes. These methods can be used

- to construct component, controller and system models based on the P&ID and SFC of a given batch process,
- to produce the diagnoser to facilitate fault diagnosis,
- to synthesize the test plans to further enhance diagnostic resolution, and
- to generate the emergency response procedures for maintaining a lower-than-normal production rate or for safe shutdown.

References

Åkesson, K., Fabian, M., Malik, R., 2006. Supremica – an integrated environment for verification, synthesis and simulation of discrete event systems. Proceedings of the 8th IEEE International Workshop on Discrete Event Systems, 384–385.
Bauer, N., Engell, S., Huuck, R., Lohmann, S., Lukoschus, B., Remelhe, M., Stursberg, O., 2004. Verification of PLC programs given as sequential function charts. In: Integration of Software Specification Techniques for Applications in Engineering. Springer, Berlin, Heidelberg, pp. 517–540.
Benveniste, A., Fabre, E., Haar, S., Jard, C., 2003. Diagnosis of asynchronous discrete-event systems: a new unfolding approach. IEEE Trans. Autom. Control 48 (5), 714–727.
Bullemer, P.T., Nimmo, I., 1994. Understanding and supporting abnormal situation management in industrial process control environments: a new approach to training. In: Proc. of the 1994 IEEE International Conference on Systems, Man, and Cybernetics, Piscataway, pp. 391–396.
Caccavale, F., Pierri, F., Iamarino, M., Tufano, V., 2009. An integrated approach to fault diagnosis for a class of chemical batch processes. J. Process Control 19 (5), 827–841.
Cassandras, C.G., Lafortune, S., 1999. Introduction to Discrete Event Systems. Kluwer Academic Publisher, Boston.
Chen, J., Jiang, Y.C., 2011. Development of hidden semi-Markov models for diagnosis of multiphase batch operation. Chem. Eng. Sci. 66 (15), 1087–1099.
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. Ind. Eng. Chem. Res. 49, 4249–4268.
Dai, Y., Zhao, J., 2011. Fault diagnosis of batch chemical processes using a dynamic time warping (DTW)-based artificial immune system. Ind. Eng. Chem. Res. 50, 4534–4544.
Debouk, R., Lafortune, S., Teneketzis, D., 2000. Coordinated decentralized protocols for failure diagnosis of discrete event systems. Discr. Event Dyn. Syst. 10 (1–2), 33–86.
Gascard, E., Simeu-Abazi, Z., 2013. Modular modeling for the diagnostic of complex discrete-event systems. IEEE Trans. Autom. Sci. Eng. 10 (4), 1101–1123.
Ghosh, K., Srinivasan, R., 2011. Immune-system-inspired approach to process monitoring and fault diagnosis. Ind. Eng. Chem. Res. 50, 4249–4268.
Gomes Cabral, F., Moreira, M.V., Diene, O., Basilio, J.C., 2015. A Petri net diagnoser for discrete event systems modeled by finite state automata. IEEE Trans. Autom. Control 60 (1), 59–71.
Hashizume, S., Yajima, T., Ito, T., Onogi, K., 2004. Synthesis of operating procedures and procedural controllers for batch processes based on Petri nets. J. Chin. Inst. Chem. Eng. 35, 363–369.
Kang, A., Chang, C.T., 2014. Automata generated test plans for fault diagnosis in sequential material- and energy-transfer operations. Chem. Eng. Sci. 113, 101–115.
Lee, J.M., Yoo, C.K., Lee, I.B., 2004. Fault detection of batch processes using multiway kernel principal component analysis. Comput. Chem Eng. 28 (9), 1837–1847.
Li, J.H., Chang, C.T., Jiang, D., 2014. Systematic generation of cyclic operating procedures based on timed automata. Chem. Eng. Res. Des. 92, 139–155.
Malik, R., Fabian, M., Akesson, K., 2011. Modelling large-scale discrete-event systems using modules, aliases, and extended finite-state automata. Proceedings of 18th IFAC World Congress 18, 7000–7005.
Nimmo, I., 1995. Adequately address abnormal situation operations. Chem. Eng. Prog. 91, 36–45.
Nomikos, P., MacGregor, J.F., 1994. Monitoring batch processes using multiway principal component analysis. AIChE J. 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. Eng. Appl. Arti. Intell. 21 (8), 1204–1206.
Qiu, W.B., Kumar, R., 2006. Decentralized failure diagnosis of discrete event system. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. 36 (3), 384–395.
Ricker, L., Lafortune, S., Genc, S., 2006. Desuma: a tool integrating GIDDES and UMDES. software tools. 8th International Workshop on Discrete-Event Systems.

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. Comput. Chem. Eng. 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. Comput. Chem. Eng. 24 (2–7), 777–784.
Sköldstam, M., Åkesson, K., Fabian, M., 2007. Modeling of discrete event systems using finite automata with variables. 46th IEEE Conference: Decision and Control.
Tan, W.L., Nor, N.M., Abu Bakar, M.Z., Ahmad, Z., Sata, S.A., 2012. Optimum parameters for fault detection and diagnosis system of batch reaction using multiple neural networks. J. Loss Prevent. Proc. Ind. 25, 138–141.
Undey, C., Ertunc, S., Cinar, A., 2003. Online batch fed-batch process performance monitoring, quality prediction, and variable contribution analysis for diagnosis. Ind. Eng. Chem. Res. 42 (20), 4645–4658.
Yeílamos, I., Bojarski, A., Joglekar, G., Venkatasubramanian, V., Puigjaner, L., 2009. Enhancing abnormal events management by the use of quantitative process hazards analysis results. Ind. Eng. Chem. Res. 48, 3921–3933.
Yeh, M.L., Chang, C.T., 2011. An automaton-based approach to evaluate and improve online diagnostic schemes for multi-failure scenarios in batch processes. Chem. Eng. Res. Des. 89, 2652–2666.
Yeh, M.L., Chang, C.T., 2012. An automata-based approach to synthesize untimed operating procedures in batch chemical processes. Korean J. Chem. Eng. 29, 583–594.
Zad, S.H., Kwong, R.H., Wonham, W.M., 2003. Fault diagnosis in discrete-event systems: framework and model reduction. IEEE Trans. Autom. Control 48 (7), 1199–1204.
Zhao, C., 2014. Quality-relevant fault diagnosis with concurrent phase partition and analysis of relative changes for multiphase batch processes. AIChE J. 60 (6), 2048–2062.