Computers and Chemical Engineering 84 (2016) 12–27

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Timed-automata based method for synthesizing diagnostic tests in batch processes

Wei-Chun Hsieh, Chuei-Tin Chang*

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

ARTICLE INFO

Article history:
Received 15 March 2015
Received in revised form 15 June 2015
Accepted 12 August 2015
Available online 20 August 2015

Keywords:
Timed automata
Diagnostic test plans
Model-checking tools
Batch processes

ABSTRACT

Hardware failures are inevitable but random events in the useful life of any batch chemical plant. If these incidents are not efficiently diagnosed, the consequences can be very serious. In general, two design measures may be implemented *offline* to enhance the overall diagnostic performance, i.e., installing sensors and/or stipulating test plans for *online* implementations. Since the former has already been studied extensively, the present study focuses only upon the latter. In a recent work, Kang and Chang (2014) proposed an effective method to conjecture diagnostic tests using the untimed automata. However, due to a lack of time-tracking mechanisms, the failure-induced behaviours cannot always be characterized adequately with such models. A systematic procedure-synthesis strategy is therefore developed in the present study by making use of the *timed* automata and the model-checking capabilities of existing software, e.g., UPPAAL (Behrmann et al., 2006). All component models are first constructed, and all possible fault propagation scenarios and their observable event traces (OETs) are next enumerated exhaustively. The optimal test plan for every OET can then be established by generating the supervisory controller to improve diagnostic resolution. Extensive case studies have also been carried out in this work to confirm the validity and effectiveness of the proposed approach.

1. Introduction

A large number of high value-added chemical products, such as the specialty chemicals, foods, semiconductors, pharmaceuticals, etc., are often manufactured in complex but flexible batch processes. Hardware failures are random but inevitable events over the lifespan of any such plant. If the root causes of a failure-induced event sequence cannot be correctly identified in time, the final consequences may be catastrophic. Generally speaking, the overall performance of a diagnostic system can be improved by capturing more online data. To this end, the obvious design strategy is to install additional sensors. However, since new hardware inevitably requires extra spending and, also, the related issues have already been discussed extensively in the literature, there are incentives to develop an alternative means for enhancing diagnostic resolution without capital investment. Yeh and Chang (2011) proposed to implement online test procedures for such a purpose, while Kang and Chang (2014) later developed an effective procedure-synthesis method to conjecture the diagnostic tests according to untimed automata.

It should be noted that several studies have already been performed to address various issues concerning fault diagnosis in batch processes. Nomikos and MacGregor (1994, 1995) utilized the multiway principal component analysis for batch process monitoring, which has later been extended for online diagnosis applications (Kourti and Macgregor, 1995; Kourti et al., 1995; Undey et al., 2003; Lee et al., 2004). Other fault identification tools, 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 been used for diagnosing the batch plants. Although satisfactory results were reported, the above methods are mostly effective for fault diagnosis in systems with relatively few interconnected units and, also, the diagnostic resolution in cases of coexisting failures may not always be acceptable.

In order to expand the scope of diagnosis in realistic applications, Chen et al. (2010) developed several Petri-net based algorithms to configure fault identification systems for plants with many more units. Since the event sequences (or traces) in multi-failure scenarios cannot be conveniently generated with the Petri-net models, their approach was limited to the single-failure incidents. On the other hand, it was found that this shortcoming can in general be avoided with the untimed automata (Sampath et al., 1995, 1996, 1998; Baroni et al., 1999, 2000; Debouk et al., 2000; Benveniste et al., 2003; Zad et al., 2003; Qiu and Kumar,

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

http://dx.doi.org/10.1016/j.compchemeng.2015.08.007
0098-1354/

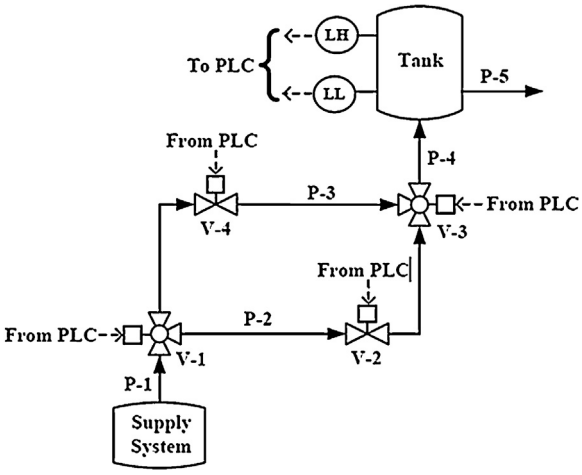

Fig. 1. P&ID of the liquid transfer system in Example 1.

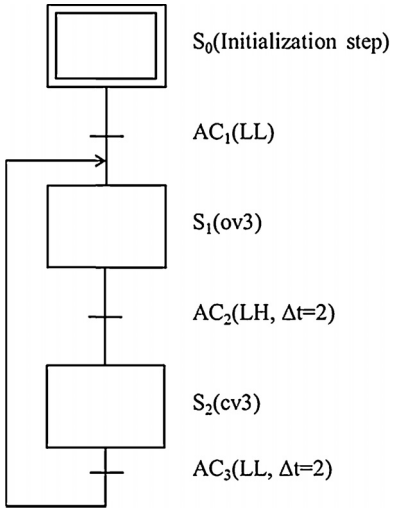

Fig. 2. Normal SFC of liquid transfer operation in Example 1.

2006; Yeh and Chang, 2011). A so-called "diagnoser" can be constructed accordingly to predict all observable fault-propagation event sequences (or "traces") and to pinpoint the corresponding fault origins. In a later study, Gascard and Simeu-Abazi (2013) improved this approach by using the timed automata to build diagnosers for the dynamic discrete-event systems.

Since the root cause(s) of a trace in the diagnoser may or may not be unique, it is desirable to further enhance the diagnostic resolution with other nonconventional means. As mentioned previously, Kang and Chang (2014) have developed a systematic method to generate the test plans for upgrading a given diagnoser without capital investment. However, due to the lack of time-tracking mechanisms in their *untimed* models, the failure-induced behaviours cannot always be characterized adequately. To overcome this drawback, it is obviously reasonable to make use of the *timed* automata for the purpose of generating more comprehensive plans. Notice that such models have already been utilized to address other closely related issues. For examples, they were used to verify if any given procedure conforms to the design specifications (Lohmann et al., 2006; Kim and Moon, 2009, 2011; Lahtinen et al., 2012), and Li et al. (2014) also proposed a systematic approach to synthesize controller actions for periodic operations.

Finally, to facilitate clear illustration of the proposed approach, the general procedure for test-plan synthesis is summarized in the sequel:

1. All embedded components in the given process are first modelled with the timed automata.
2. All possible fault propagation scenarios and their observable event traces (OETs) are next enumerated exhaustively.
3. The optimal test plan for every OET is then established by generating the supervisory controller to achieve a higher degree of diagnostic resolution.

The resulting test plans can then be implemented online after observing any of the OETs in diagnoser during actual operation.

2. General approach to build plant model

Since the modelling principles proposed by Kang and Chang (2014) are generic enough, their basic approach is adopted to build time automata in the present study. For the sake of clarity, this model construction method is illustrated here with a simple example. Specifically, let us consider a fictitious liquid transfer system represented by the piping and instrumentation diagram (P&ID) in Fig. 1 and also the sequential function chart (SFC) in Fig. 2. Notice that the components in this and any other batch process can be classified into a hierarchy of 4 different levels: (1) the programmable logic controller (PLC); (2) the actuators, i.e., the three-way valves (V-1 and V-3) and the two-way valves (V-2 and V-4); (3) the processing units, i.e., the buffer tank' and (4) the online sensor(s). If a three-way valve is closed, the port connecting to the horizontal pipeline in Fig. 1, i.e., pipe P-2 in the case of V-1 or pipe P-3 in the case of V-3, is assumed to be blocked. Otherwise, its inlet flow(s)

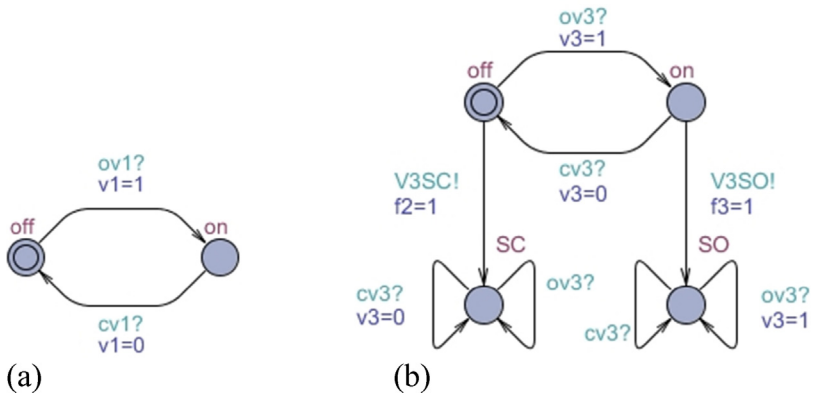

Fig. 3. Valve models for (a) V-1 and (b) V-3 in Example 1.

should be directed to every outlet pipeline. It is assumed that all valves except V-4 are closed initially. Thus, it clear from the SFC in Fig. 2 that, during the normal operation, the buffer tank is filled in two time units ($\Delta t = 2$) with liquid via pipelines P-1, P-3 and P-4 after opening V-3 and then drained via P-5 by gravity in another two time units ($\Delta t = 2$). For the sake of brevity, only 4 fault origins are considered in this example and they are denoted respectively by:

- f_{1A} (or *Tank_LEAK*), i.e., a large leak develops in tank,
- f_{1B} (or *Tank_leak*), i.e., a minor leak develops in tank,
- f_2 (or *V3SC*), i.e., V-3 fails at the "close" position, and
- f_3 (or *V3SO*), i.e., V-3 fails at the "open" position.

The plant model can in general be obtained by first building automata to model all components in the given process and then integrating them via the parallel decomposition operation (Cassandras and Lafortune, 1999). It should be noted that the free software UPPAAL (Behrmann et al., 2006) was used for model building and verification in the present work. Note also that the controller and the remaining components are characterized differently. Let us first outline the construction principles for the latter components, i.e., the valves, the tank and the sensors. Specifically, a timed automaton should be used to represent a finite set of all identifiable normal and abnormal states of the hardware item under consideration and also the specific events facilitating the state transitions. The prerequisite conditions of an event can be imposed with the so-called "guard" of the corresponding transition in UPPAAL, while the updated integer and clock variables after transition may also be specified as attributes (Behrmann et al., 2006). Finally, it is assumed that the failures and the resulting abnormal states included in these component models can be identified in advance with an available hazard assessment method. The corresponding models in the present example are briefly described below:

- Valve models: Since only the failures of V-3, i.e., *V3SC!* and *V3SO!*, are considered in the present example, V-1, V-2 and V-4 can be modelled with automata of the same structure and, therefore, only the component models of V-1 and V-3 are presented in Fig. 3(a) and (b) respectively. The place *off* in the former case is used to represent the closed position of V-1, while *on* the opposite state. The "receiver" events in this model are labelled with question marks, i.e., *ov1?* and *cv1?*, indicating that these events must occur in other components at some prior instances. The former triggers the *off*-to-*on* state transition and then resets the binary variable *v1* to 1, while the latter activates the *on*-to-*off* transition and resets *v1* to 0. The normal behaviour of V-3 described in Fig. 3(b) is essentially the same as that in Fig. 3(a), while the sender events *V3SO!* and *V3SC!* activate the transitions to the failed states *SO* and *SC* respectively. Note that the exclamation mark (*!*) is used here to specify an initiator or "sender" event that takes place in the present component as long as all prerequisite conditions can be satisfied. Notice also that a system deadlock, i.e., a state that no further event can be executed, is usually formed after reaching any of the failed states.
- Tank model: Only the elapsed times of state transitions in the tank model are assumed to be nonzero and the corresponding automaton can be found in Fig. 4. Note that the places *LL* and *LH* are used to respectively represent the low and high liquid levels under normal conditions, while *LL_leak* and *LH_leak* denote the corresponding liquid levels after a minor leak develops and *LL_LEAK* is the low level eventually reached after a large leak. For the sake of illustration brevity, let us consider only the attributes associated with the *LL*-to-*LH* transition. Three conditions (guards), formulated with four binary variables (*v1*–*v4*) and a clock variable *x*, must be satisfied before triggering this transition. The binary variables *v1*–*v4* are used to represent the corresponding valve states, i.e., 1 denotes open and 0 otherwise, and the clock variable *x* records time needed to complete the corresponding state transition. The second attribute *Tank_LH!* denotes the transitional event itself and, as mentioned before, the exclamation mark (*!*) is used to specify that it is an initiator or "sender" event. The last attribute is the reset condition, i.e., the liquid level is reset to high ($L = 1$) after completing the transition.
- Sensor model: Since sensor failures are not considered in this example, it is assumed that the online measurements always accurately reflect the tank states and, thus, the sensor model is omitted for the sake of brevity. It should be emphasized that this practice does not result in a loss in generality since the sensor models can always be built with the same principles described previously.

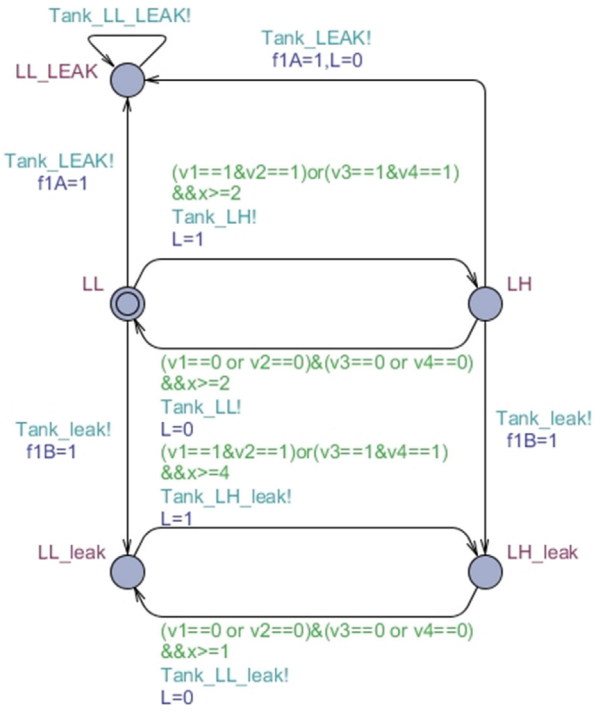

Fig. 4. Tank model in Example 1.

The controller model can be constructed on the basis of the given SFC, the failures and also the failure-induced events (see Fig. 5). The controller actions in this model, i.e., open V-3 (*ov3!*) and close V-3 (*cv3!*), should naturally be viewed as senders, while the observable state-transition events (i.e., *Tank_LH?* and *Tank_LL?*), the unobservable failures (i.e., *V3SC?*, *V3SO?*, *Tank_leak?* and *Tank_LEAK?*) and the failure-induced events (i.e., *Tank_LL_leak?*, *Tank_LH_leak?* and *Tank_LL_LEAK?*) are receivers.

3. Exhaustive identification of fault propagation scenarios

According to Clarke et al. (1986), "model checking" is essentially an algorithmic procedure for verifying whether a given system is compliant with the target specifications. A well-tested software verifier can often be applied to determine if a set of timed automata conform to the desired system properties. In cases when there is any specification violation, the verifier can provide a counter scenario, from which the user should be able to find error(s) and then modify the models accordingly. In this study, the model checking tool provided in UPPAAL (Behrmann et al., 2006) was utilized

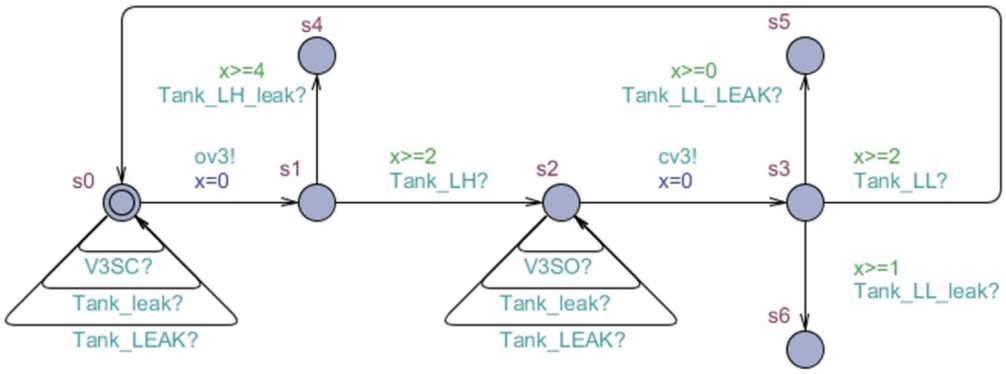

Fig. 5. Controller model in Example 1.

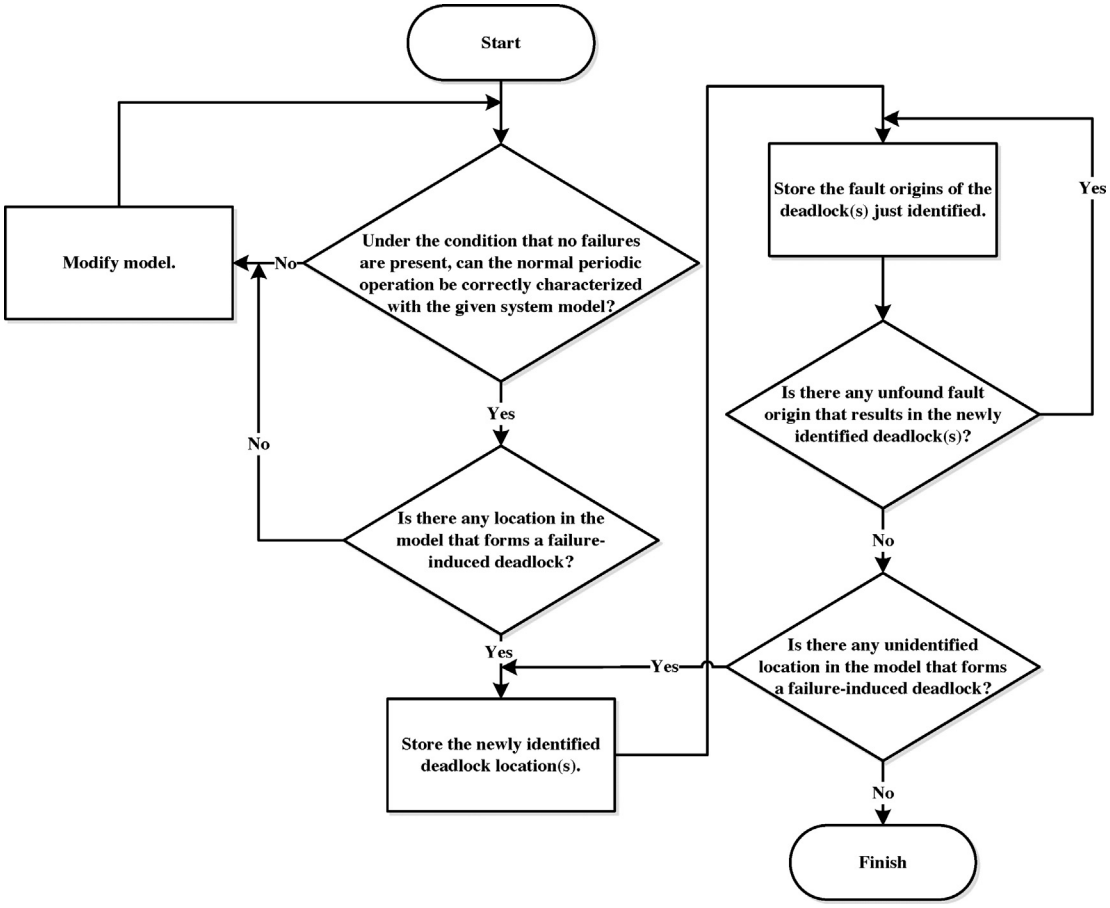

Fig. 6. Deductive reasoning procedure for identifying fault propagation paths.

for enumerating all fault propagation paths. Fig. 6 summaries the required reasoning procedure for logic deduction.

As mentioned before, failures may occur randomly at any time over the lifespan of a batch plant. Since these events are usually not directly perceptible with sensors, the fault origins can only be diagnosed with other available information. Under the assumption that the sensor measurements, the actuator signals and the clock readings can be obtained online, all observable event traces (OETs) in the present example can be identified according to the proposed reasoning procedure and summarized in Fig. 7. The rectangles in this figure are used to specify the implied system states, which may be either normal (*N*) or under the influence of one or more failure, while the arrows are transitions triggered by the corresponding observable events. Note also that every abnormal event is marked with a double quote. Let us consider these traces one-by-one:

- **Trace 1**: The first transition on this trace is used to represent the event sequence that may be experienced in i (where, $i = 0, 1, 2, \ldots$) completed normal cycles. The subsequent action is the first step of SFC, i.e., *ov3*, which should normally result in a high liquid level (*LH*). However, if the abnormal state *LL* is detected instead, one can deduce that there are four possibilities: (1) f_{1A}; (2) $f_{1A}f_2$; (3) $f_{1B}f_2$; (4) f_2. Note that, in either 2nd or 3rd scenario, the notation denotes that there are two coexistent failures.
- **Trace 2**: As shown in Fig. 7, the event sequence on this trace is almost identical to that on Trace 1. The only difference is the liquid

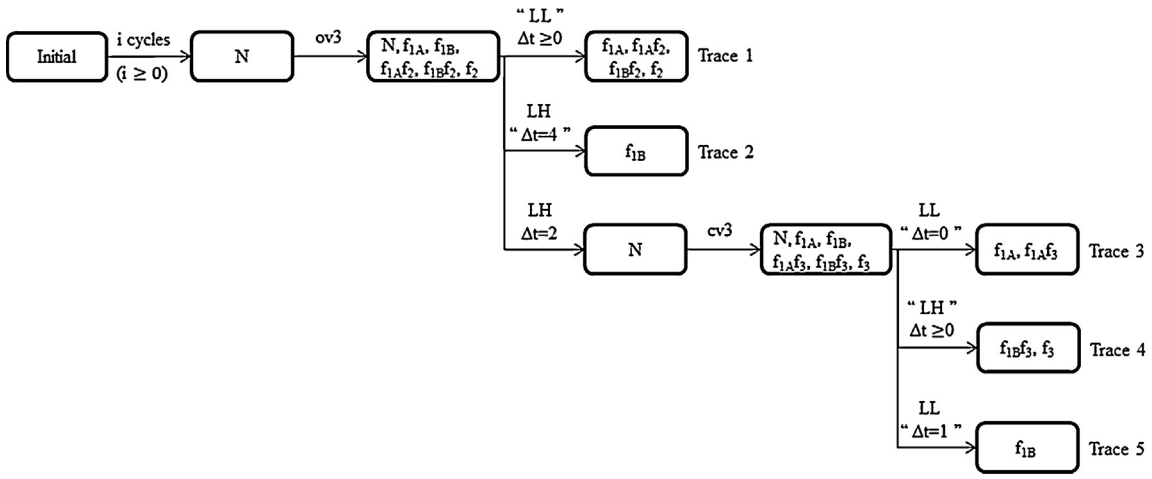

Fig. 7. Observable event traces in Example 1.

level reached after controller action *ov3*, i.e., the resulting target state (*LH*) is achieved in this case in a longer-than-normal period ($\Delta t = 4$). The only implied fault origin in this case is f_{1B}.

- **Trace 3**: As shown in Fig. 7, this trace and first two traces overlap in the initial stage. After completing *ov3*, the target state *LH* is reached in the allotted time period ($\Delta t = 2$) and, thus, the subsequent step in SFC, i.e., *cv3*, must be applied next. If the tank is emptied (*LL*) almost immediately afterwards, then it can be deduce that there are two possible root causes, i.e., (1) f_{1A} and (2) $f_{1A}f_3$.
- **Trace 4**: This trace is essentially the same as Trace 3 except the final symptom. The anticipated state *LL* can never be reached since the liquid level is maintained at the original level *LH* indefinitely in this scenario. The implied fault origins are (1) $f_{1B}f_3$ and (2) f_3.
- **Trace 5**: This trace is also the same as the previous two traces except for the final symptom. After executing the control action *cv3* in this case, if the target state *LL* can be reached in a shorter-than-expected time period ($\Delta t = 1$), this abnormally quick response can only be attributed to f_{1B}.

4. Systematic construction of test plans

If two or more different fault origins are implicated after observing a fully developed OET during operation, additional tests may be performed to further enhance diagnostic resolution. The test plan of an OET can be produced with the aforementioned model checking tool and the synthesis steps summarized in Fig. 8. In order to implement this procedure, all standard component models are needed except that of the controller. As shown in Fig. 9, the controller model in Example 1 must be modified by introducing an extra transition which points away from the deadlock location (at which there are no active events) reached in every scenario implied by Trace 1, i.e., f_{1A} (*Tank_LEAK?*), $f_{1A}f_2$ (*Tank_LEAK?* and *V3SC?*), $f_{1B}f_2$ (*Tank_leak?* and *V3SC?*), and f_2 (*V3SC?*). This added transition is activated by a fictitious receiver event (*test?*) and terminated at an artificial place without outputs. To guide test-plan synthesis, an additional automaton should also be constructed and subsequently modified repeatedly according to the following two procedures:

- Procedure A:
  - If $n_{stage} = 0$, connect artificial places *s0*, *s1* and *s2* in series and, then, connect *s2* to *s3a*, *s3b*, *s3c*, etc., in parallel. The guards on transition $s0 \rightarrow s1$ should be all fault origins implied by the given OET, while its triggering event is *test1!*. To facilitate evaluation of all feasible steps in the test, multiple loops are then assembled with *s1* and *s2* and each is associated with an allowed test action. Note also that, although these two places can in fact be merged into one to form self-recycle loops, the current configuration is adopted simply for the sake of legibility. Finally, a distinct loop is also constructed between *s2* and every downstream place (i.e., *s3a*, *s3b*, *s3c*, etc.) to represent a unique state-transition event observed online with a sensor and/or a clock. It should be noted that all aforementioned loops are used mainly for creating multi-step procedures. Fig. 10 shows the test model built for Trace 1 in Example 1.
  - If $n_{stage} > 0$, remove the guards for the confirmed fault origins in the original test model and then insert additional places between *s0* and *s1* to incorporate the confirmed test steps. Fig. 11 shows the test model built for Trace 1 after implementing the test steps *ov1* and *ov2* in Example 1. Note that, since these actions can be applied to produce unique responses for the 3rd and 4th implied fault origins, i.e., $f_{1B}f_2$ and f_2, respectively, only the first two are imposed as guards on the transition $s0 \rightarrow s1a$.
- Procedure B:
  - Remove all loops between *s1* and *s2* in the test model established according to Procedure A and a specific n_{stage}, and then insert additional places between them to incorporate the confirmed test steps. Fig. 12 shows the modified model built for Trace 1 in Example 1 after identifying the test steps *ov1* and *ov2* in the initial stage ($n_{stage} = 1$).

By appropriately checking these models according to Fig. 7, the SFCs in Figs. 13 and 14 can be generated for fault diagnosis after observing Traces 1 and 4 respectively. In the former case, the control actions *ov1* and *ov2* in the step S1 are called for as soon as the first activation condition, i.e., AC_1 (Trace 1), is confirmed. There may be three resulting scenarios depending on the sensor and clock readings obtained after the above test actions, i.e., (1) AC_2 (*LL*, $\Delta t \geq 0$), (2) AC_3 (*LH*, $\Delta t = 2$), and (3) AC_4 (*LH*, $\Delta t = 4$), and the corresponding fault origins should be: (1) f_{1A} or $f_{1A}f_2$; (2) f_2; (3) $f_{1B}f_2$. On the other hand, the first activation condition in the latter trace, i.e., AC_1 (Trace 4), prompts the test action in S_1 (*cv4*). There may be two possibilities, i.e., (1) AC_2 (*LL*, $\Delta t = 1$), which implies that the fault origin is $f_{1B}f_3$, and (2) AC_3 (*LL*, $\Delta t = 2$), which implies f_3.

Note that the test plans for the other three OETs are not presented here. There are obviously no needs to perform tests for Trace 2 or 5 since there is only one possible cause in either case, while none can be identified with the proposed procedure for Trace 3. Finally, if the above results are compared with those reported in Kang and Chang (2014), it is clear that the present test plans are

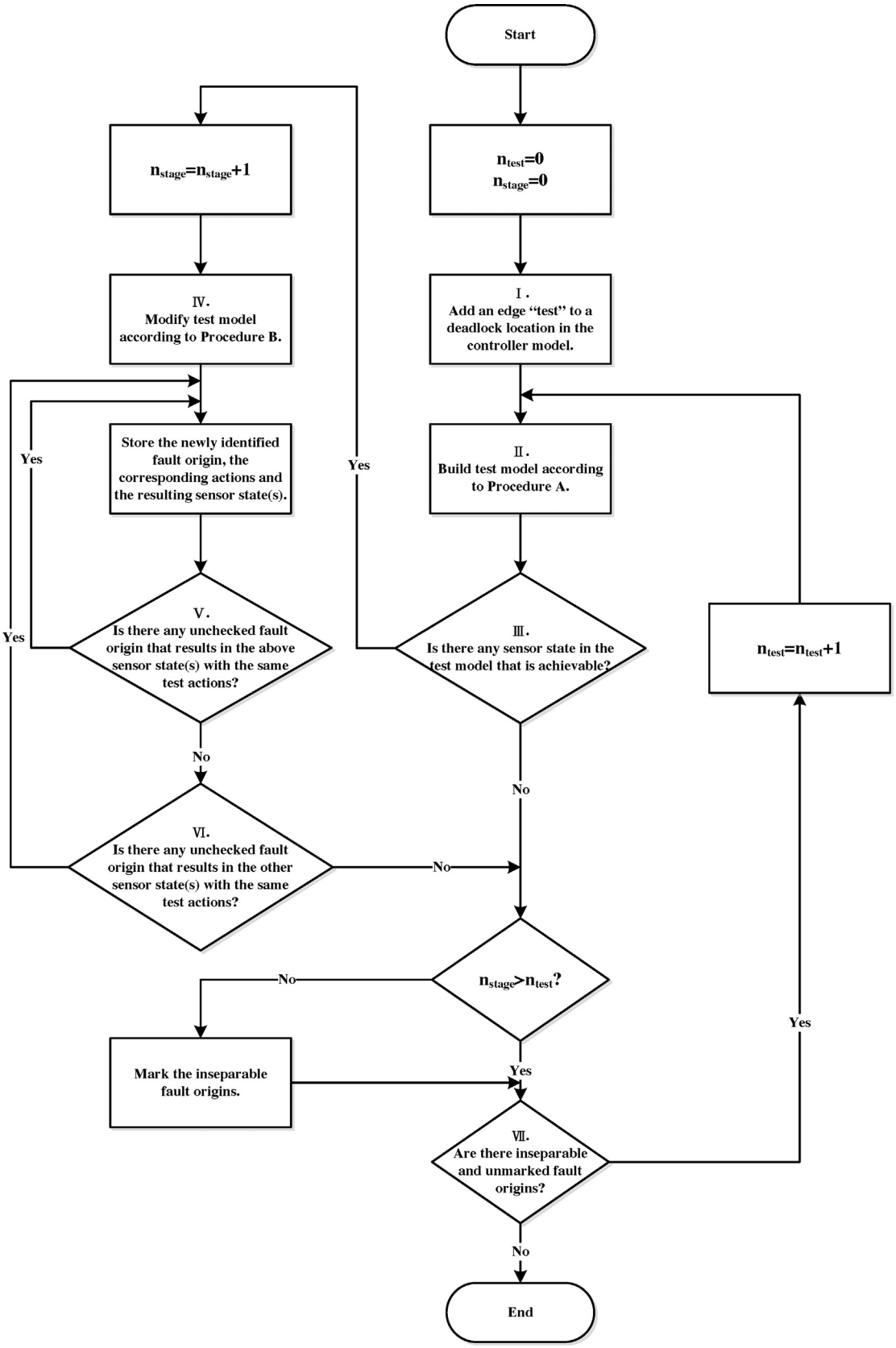

Fig. 8. Test-plan synthesis procedure.

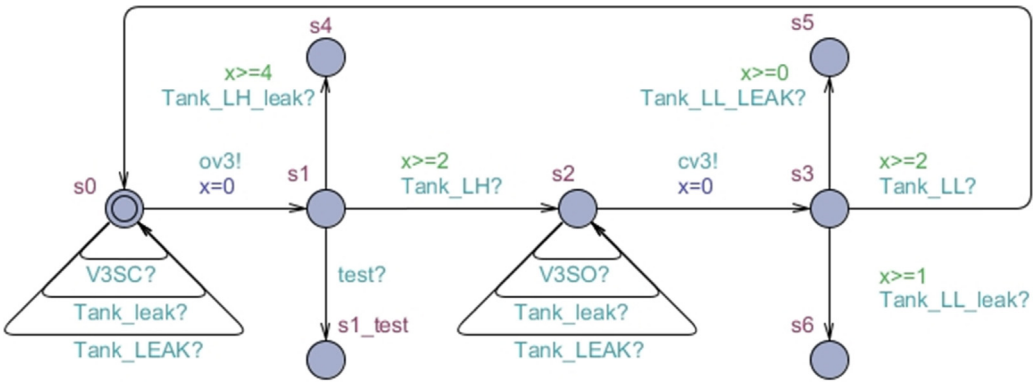

Fig. 9. Modified controller model for Trace 1 in Example 1.

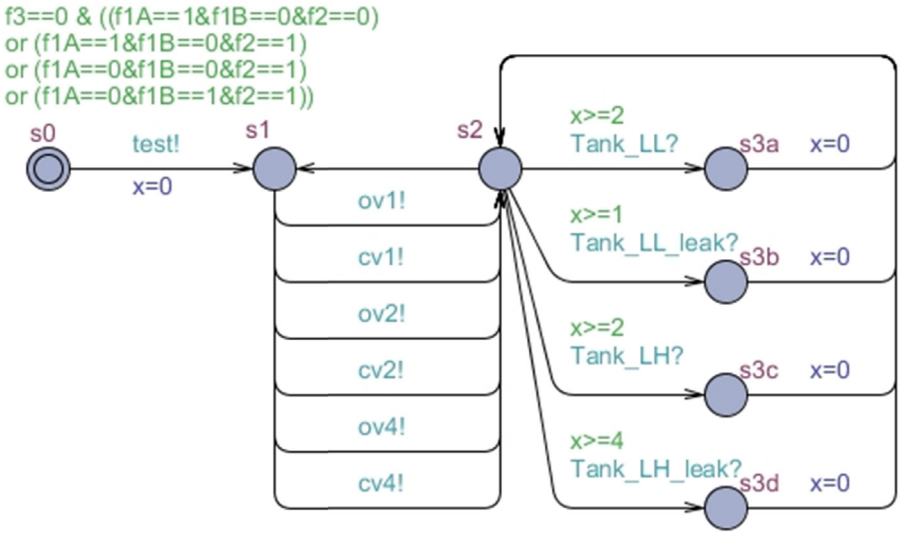

Fig. 10. Test model generated for Trace 1 with Procedure A in Example 1 ($n_{stage} = 0$).

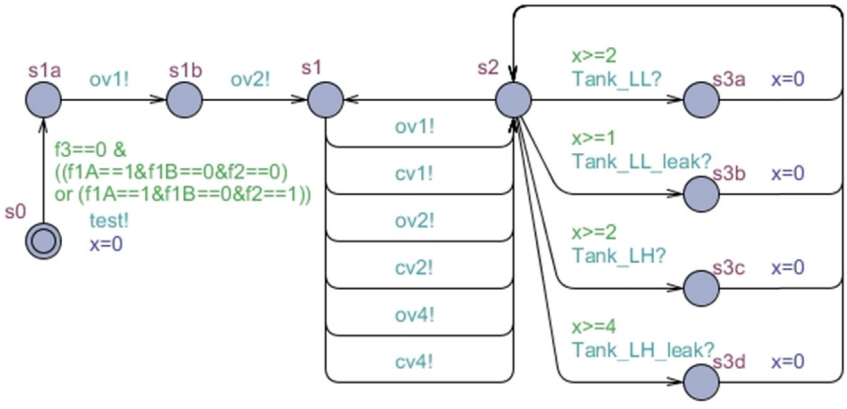

Fig. 11. Test model generated for Trace 1 with Procedure A in Example 1 ($n_{stage} = 1$).

superior since they are capable of differentiating different degree of tank leaks. This is due to the inherent feature of time automata allows proper representation of the elapsed time associated with every state-transition process.

5. Case studies

To verify the effectiveness of the proposed test-plan synthesis approach, a series of extensive case studies have been carried out and two of them are summarized below:

Example 2. A three-tank buffer system

Let us consider the P&ID in Fig. 15 and its normal operating procedure specified in Fig. 16. Notice that V-2 is a 3-way valve and V-1, V-3 and V-4 are traditional gate valves. The fluid in tank T-1 is directed to tank T-2 if V-2 is placed at the + position and pump is switched on, while transported to T-3 if switched to the – position. All three tanks are equipped with level sensors. The one on T-1 is designed to detect three distinct states reflecting the low, intermediate and high liquid levels, i.e., *T1L*, *T1M* and *T1H*, respectively,

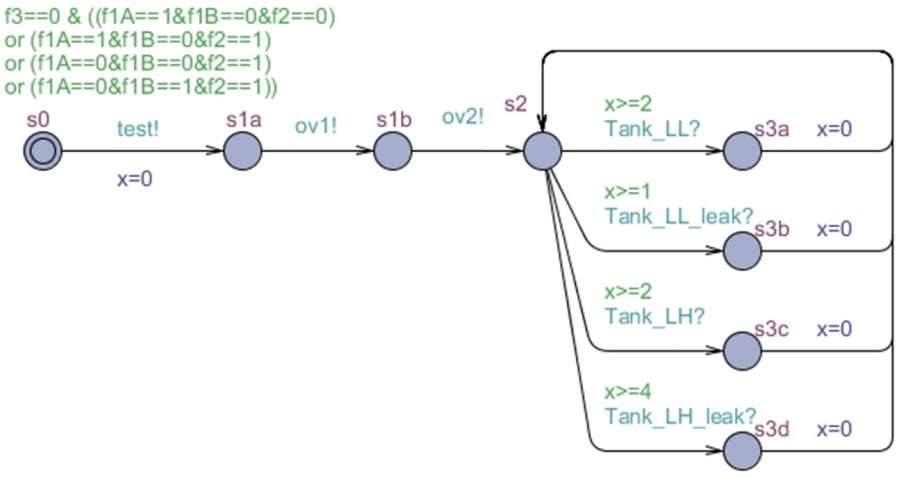

Fig. 12. Test model modified with Procedure B for Trace 1 in Example 1 (n_{stage} = 1).

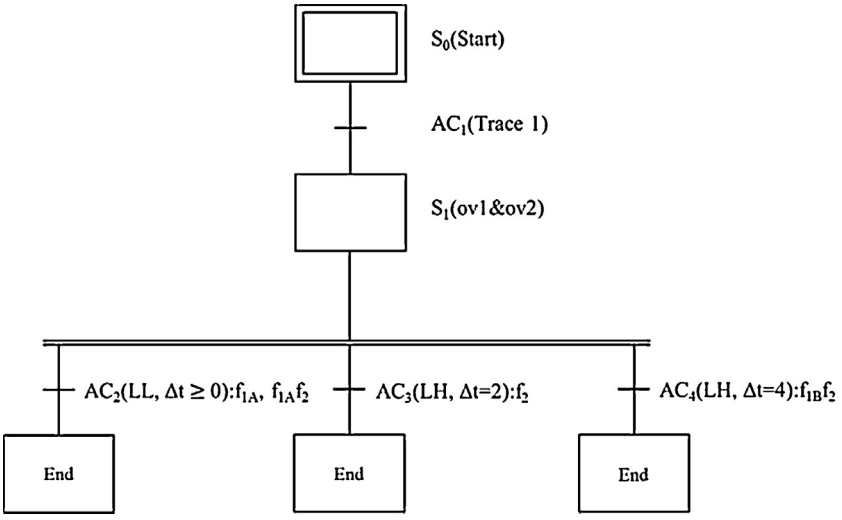

Fig. 13. Optimal test procedure for Trace 1 in Example 1.

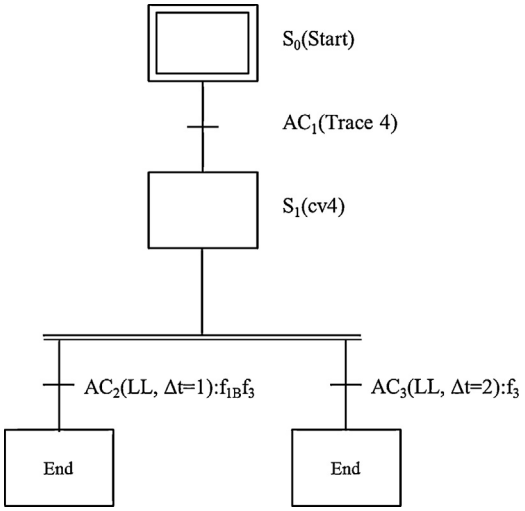

Fig. 14. Optimal test procedure for Trace 4 in Example 1.

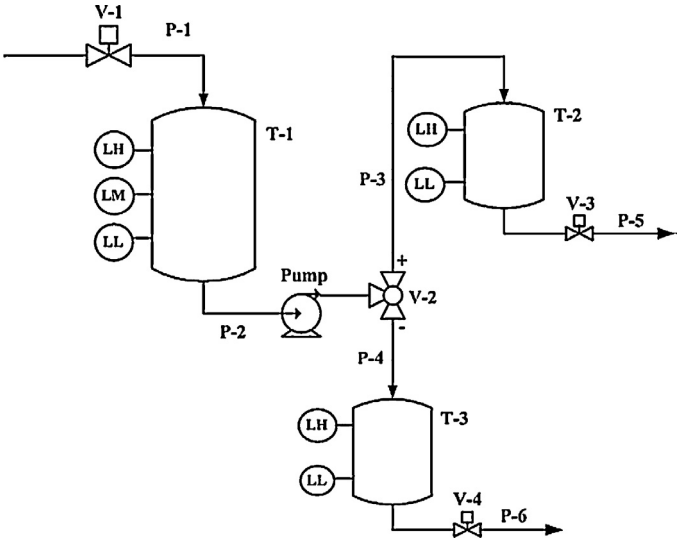

Fig. 15. P&ID of the three-tank buffer system in Example 2.

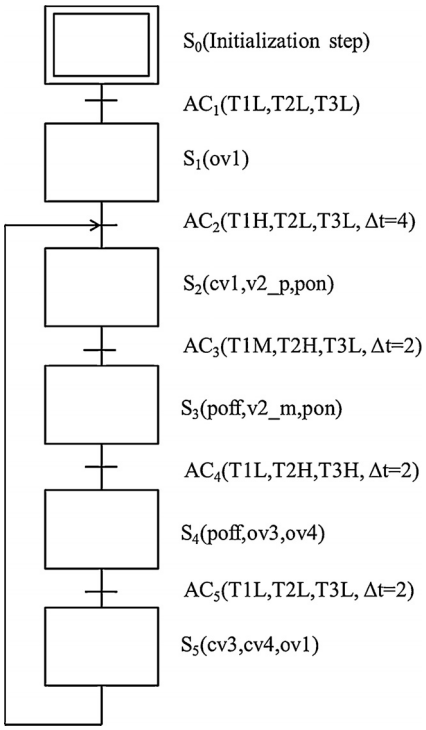

Fig. 16. Normal SFC of the three-tank buffer system in Example 2.

while that on T-2 (or T-3) is used to monitor only the states at low and high levels, i.e., *T2L* (or *T3L*) and *T2H* (or *T3H*). It is assumed that, initially, the liquid levels in all tanks are low, valves V-1, V-3 and V-4 are closed and V-2 is at the − position. In this example, let us consider the following seven failures:

i. f_1 (*V1SC*), i.e., V-1 fails at the closed position,
ii. f_2 (*V1SO*), i.e., V-1 fails at the open position,
iii. f_3 (*V2M_p*), i.e., V-2 is mistakenly switched to the + position,
iv. f_4 (*V2M_m*), i.e., V-2 is mistakenly switched to the − position,
v. f_5 (*V3SC*), i.e., V-3 fails at the closed position,
vi. f_6 (*V3SO*), i.e., V-3 fails at the open position, and
vii. f_7 (*T2_leak*), i.e., a leak develops in tank T-2.

The aforementioned modelling approach and the reasoning procedure in Fig. 6 have been followed to identify the diagnosable OETs in Fig. 17(a) and the undiagnosable ones in Fig. 17(b). For the sake of brevity, only the latter scenarios are illustrated specifically in the sequel:

- **Trace 9**: This trace develops before the initial cycle ($i=0$) can be completed. Four consecutive event sets are observed in the precedence order given below:
  (1) V-1 is opened (*ov1*) immediately after operation starts.
  (2) Liquid level in T-1 reaches *T1H* in 4 time units ($\Delta t=4$), while those in T-2 and T-3 are maintained at *T2L* and *T3L* respectively.
  (3) V-1 is closed (*cv1*), V-2 is switched to the + position (*v2_p*) and then pump is switched on (*pon*).
  (4) Liquid level in T-1 reaches *T1M* in 2 time units ($\Delta t=2$), while the abnormal states in T-2 and T-3, i.e., *T2L* and *T3H*, are observed at the same time.
  It can be observed that the system behaves normally in the initial stage until when the symptoms *T2L* and *T3H* show up. The possible root causes in this case should be: (1) f_3 and (2) f_3f_7.
- **Trace 10**: This event sequence also takes place in the initial cycle ($i=0$). The first three groups of normal events are the same as those on Trace 9, while abnormal conditions in all tanks appear afterwards in 2 time units ($\Delta t=2$), i.e., the liquid level in T-1 is kept unchanged at *T1H*, and those in T-2 and T-3 reach *T2L* and *T3H*, respectively. The implied fault origins in this case should be: (1) f_2f_3 and (2) $f_2f_3f_7$.
- **Trace 11**: Notice that the initial action on this trace is *ov1* (i.e., open V-1) and, on the next transition, the label *i cycles* denotes the event sequence in one or more completed normal cycle. The abnormal symptom *T1L*, i.e., the liquid level in T-1 is low, is maintained indefinitely after completion of i ($i\geq 1$) normal cycles and, thus, the desired activation condition *T1H* in AC_2 can never be satisfied in this scenario. The implied fault origins are: (1) f_1 and (2) f_1f_6.
- **Trace 12**: The initial event sequence of this trace, i.e., *ov1* and *i cycles*, is identical to that of Trace 11, while the remaining part is essentially the same as that of Trace 9 after the first control action *ov1*. When compared with Trace 9, one could observe that more fault origins can be implicated with this OET, i.e., (1) f_3, (2) f_3f_7, (3) f_3f_6, and (4) $f_3f_6f_7$.
- **Trace 13**: The initial event sequence of this trace, i.e., *ov1* and *i cycles*, is identical to that of Trace 11, while the remaining part is essentially the same as that of Trace 10 after the first control action *ov1*. The implied fault origins in this case, i.e., (1) f_2f_3 (2) $f_2f_3f_7$, (3) $f_2f_3f_6$ and (4) $f_2f_3f_6f_7$, are also more than those associated with Trace 10.
- **Trace 14**: The event sequence of this trace is essentially the same as those of Traces 12 and 13 except the final symptoms, i.e., the liquid levels in tanks T-1, T-2 and T-3 are always kept unchanged at *T1H*, *T2L* and *T3L* respectively. The first two tank states are abnormal since the previous control actions (*cv1*, *v2_p* and *pon*) have been applied to transfer material from T-1 to T-2. The corresponding fault origins could be (1) f_2f_6 and (2) $f_2f_6f_7$.
- **Trace 15**: The event sequence of this trace is also the same as those of Traces 12 and 13 except the final conditions, i.e., the liquid levels in tanks T-1, T-2 and T-3 become *T1M*, *T2L* and *T3L* respectively after 2 time units ($\Delta t=2$). The liquid level in T-2 is not expected in SFC and this abnormality may be attributed to (1) f_6 or (2) f_6f_7.

It was found that, after applying the proposed synthesis procedure in Fig. 8, the root causes implied by Traces 11, 14 and 15 cannot be further distinguished via diagnostic tests. On the other hand, note that the fault origins associated with Trace 9 actually form a subset of those corresponding to Trace 12 and, similarly, every fault origin implied by Trace 10 is also by Trace 13. Thus, only the test plans of Traces 12 and 13 are presented in Figs. 18 and 19 respectively and these plans are also summarized in the sequel:

- **Plan 2.Tr12**: As shown in Fig. 18, the test action *v2_p* (i.e., switch V-2 to the + position) in step S1 is called for as soon as AC_1 (Trace 12) is observed online. There may be three resulting scenarios: (1) AC_2 (*T1L*, *T2H*, *T3H*, $\Delta t=2$), and a single-failure fault origin f_3 can be confirmed; (2) AC_3 (*T1L*, *T2H*, *T3H*, $\Delta t=4$), and the two-failure fault origin f_3f_7 can be confirmed; (3) AC_4 (*T1L*, *T2L*, *T3H*, $\Delta t=2$), and two multi-failure fault origins can be implicated, i.e., f_3f_6 and $f_3f_6f_7$.
- **Plan 2.Tr13**: As shown in Fig. 19, the test action *v2_p* (i.e., switch V-2 to the + position) in step S1 is called for as soon as AC_1 (Trace 13) is observed online. This action may result in three possible outcomes: (1) AC_2 (*T1H*, *T2H*, *T3H*, $\Delta t=2$), and a two-failure fault origin f_2f_3 can be confirmed; (2) AC_3 (*T1H*, *T2H*, *T3H*, $\Delta t=4$), and the three-failure fault origin $f_2f_3f_7$ can be confirmed; (3) AC_4 (*T1H*, *T2L*, *T3H*, $\Delta t\geq 0$), and two multi-failure fault origins can be implicated, i.e., $f_2f_3f_6$ and $f_2f_3f_6f_7$.

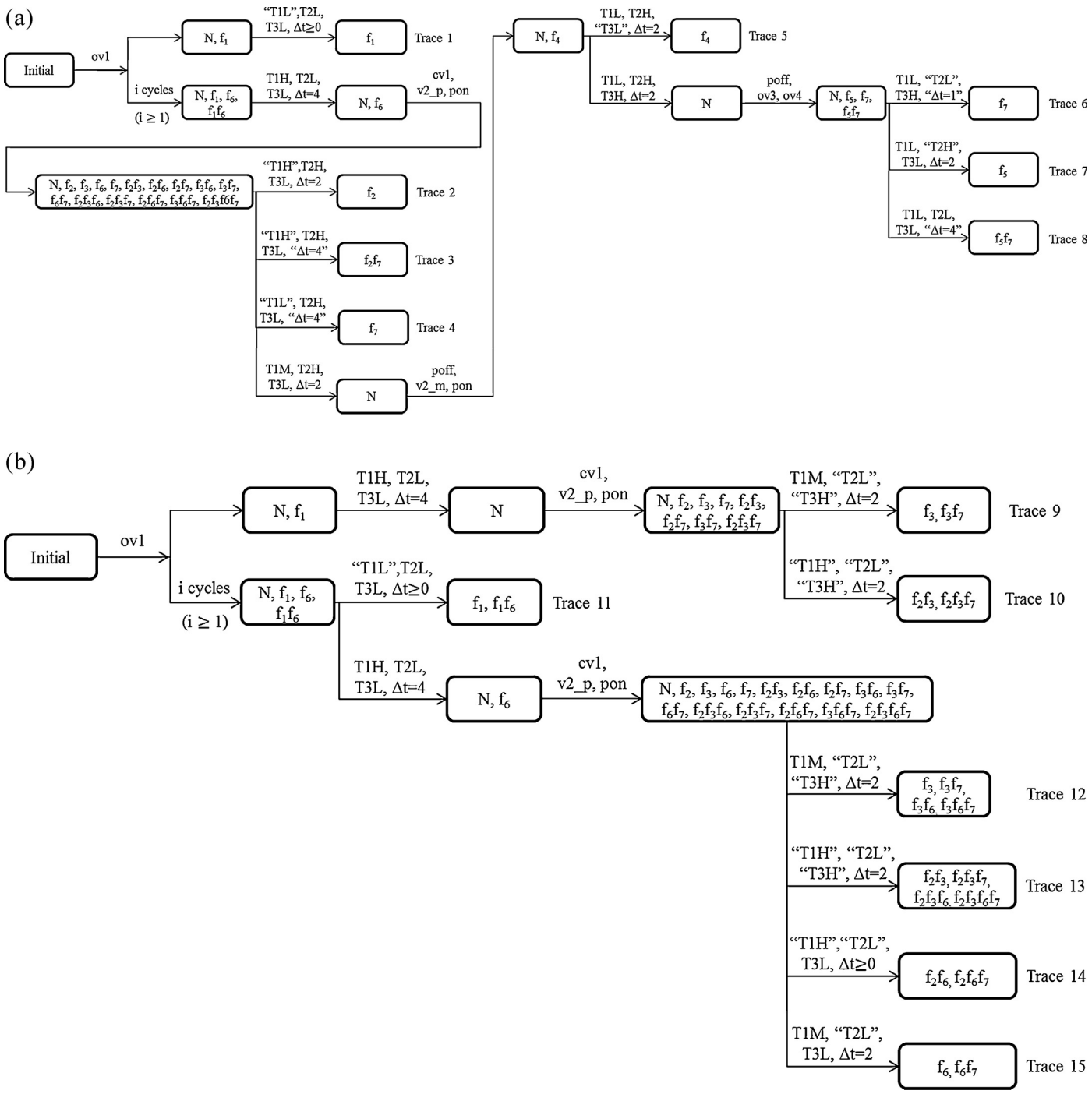

Fig. 17. (a) Diagnosable OETs in Example 2, and (b) undiagnosable OETs in Example 2.

Again, if compared with the results reported in Kang and Chang (2014), the test plans obtained in the present example can be adopted to achieve higher diagnostic resolution. Specifically, tank leaks of different magnitudes can be differentiated since the elapsed times of various level-changing processes can be properly modelled with timed automata.

Example 3. A batch evaporation system

This example is essentially an adapted version of the batch evaporation system studied in Bauer et al. (2004). Let us consider the P&ID in Fig. 20 and assume that:

(a) All actuators, i.e., the gate valves (V1–V4), the pump (P1) and the electric heaters (H1 and H2), can be manipulated with a programmable logic controller;
(b) The evaporator T1 is equipped with sensors to monitor level, temperature and a concentration. To facilitate normal and test operations, several target sensor readings of each variable must be acquired online and they are labelled in this example as:
- *LL* (low), *LM* (middle) and *LH* (high) for levels,
- *TL* (low), *TH* (high) and *THH* (higher than high) for temperatures, and
- *QL* (low) and *QH* (high) for concentrations.

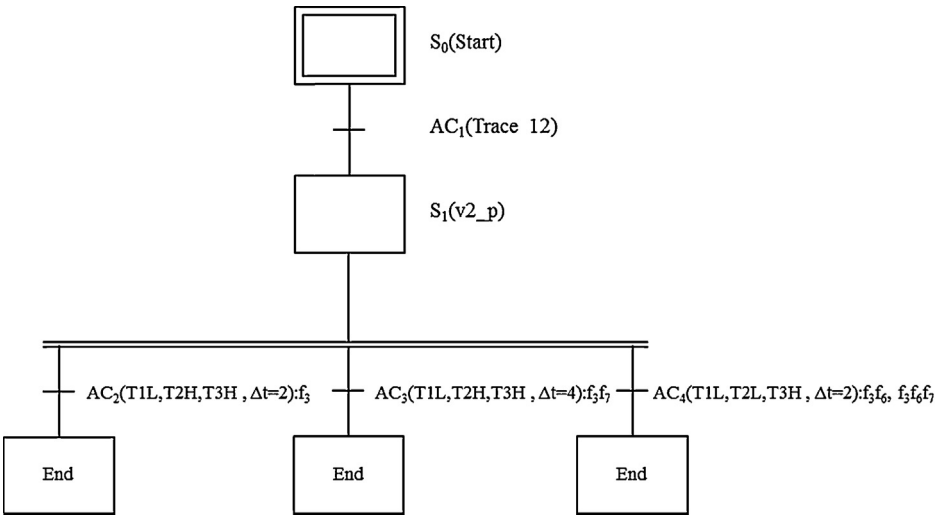

Fig. 18. Plan 2.Tr12 in Example 2.

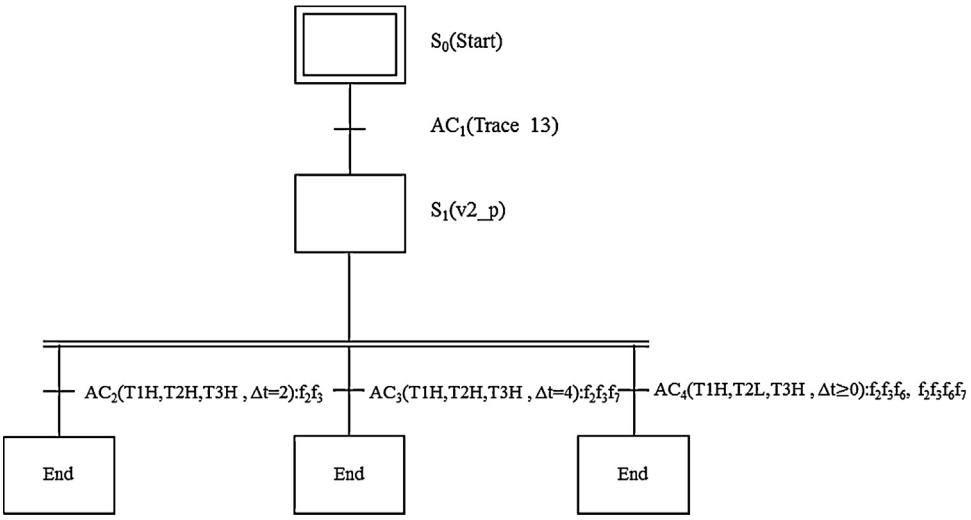

Fig. 19. Plan 2.Tr13 in Example 2.

(c) The condenser C1 is equipped with a flow sensor to determine if the flow rate of cooling medium reaches *FL* (low or zero) and *FH* (high).
(d) The buffer vessel T2 is also equipped with level and temperature sensors. The level targets are also denoted as *LL* (low) and *LH* (high), while the temperature targets *TL* (low) and *TH* (high).

The initial system state is set according to the additional assumptions that, before the evaporation operation begins, all actuators are switched off, evaporator T1 and buffer tank T2 are both empty, and cooling in C1 is not provided. Based on these initial conditions, the normal operating procedure adopted in the present example can be specified with the SFC in Fig. 21. To facilitate concrete discussions, let us consider the following seven hardware failures:

i. f_1 (*V1SC*), i.e., V1 fails at the closed position;
ii. f_2 (*V1SO*), i.e., V1 fails at the open position;
iii. f_3 (*ov4_M*), i.e., controller fails to open V4;
iv. f_4 (*H1_failure*), i.e., heater H1 fails;
v. f_5 (*T2_leak*), i.e., a leak develops in T2;
vi. f_6 (*P1_failure*), i.e., pump P1 fails;
vii. f_7 (*QIS_failure*), i.e., concentration analyzer on T1 fails.

The aforementioned modelling approach and the reasoning procedure in Fig. 6 have again been applied to produce the OETs in Fig. 22. For the sake of brevity, only the undiagnosable scenarios are described in detail in the sequel:

- **Trace 1**: After experiencing the initial state mentioned above and also the event sequence in i ($i \geq 0$) normal cycles, the control action *ov1* (i.e., opening valve V1) must be executed next. According to the SFC given in Fig. 21, the level sensors on T1 and T2 should reach targets *LH* and *LL*, respectively, in four time units ($\Delta t = 4$) and the corresponding reading of the flow sensor on C1 must be *FL*. However, the level reading of T1 at this time remains at *LL*, which may be attributed to f_1 or $f_1 f_5$.
- **Trace 2**: The initial sequence on this trace is the same as that on Trace 1, i.e., *i cycles* and then *ov1*. Instead of the final symptom observed in the previous scenario, the subsequent responses of control action *ov1* after 4 time units, i.e., the sensor readings *LH*, *TL* and *QL* for T1, *LL* for T2 and *FL* for C1, are all normal events in this case. According to the SFC in Fig. 21, the next moves should be to close V1 (*cv1*), switch on heater H1 (*heat_H1*) and open V4 (*ov4*). Although an immediate change to target *FH* in the cooling medium flow is expected, the online reading of flow sensor on C1 is kept at *FL* for an indefinite period of time in the present

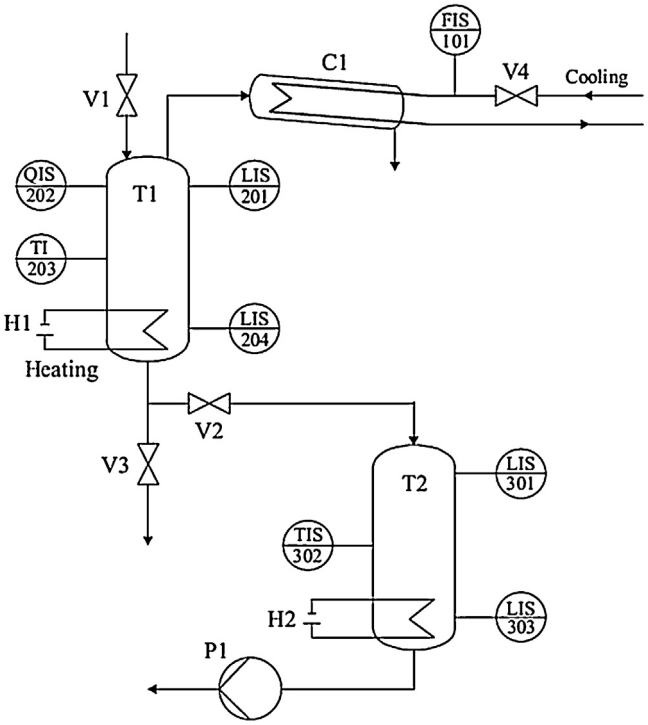

Fig. 20. P&ID of the batch evaporation system in Example 3.

scenario. It can be deduced that this outcome may be caused by 8 possible fault origins, i.e., (1) f_3, (2) f_2f_3, (3) f_3f_4, (4) f_3f_5, (5) $f_2f_3f_4$, (6) $f_2f_3f_5$, (7) $f_3f_4f_5$, and (8) $f_2f_3f_4f_5$.

- **Trace 3**: Except for the final symptom, the event sequence on Trace 2 also appears on Trace 3. The fault propagation path branches into two after the steps to close V1 (*cv1*), switch on heater H1 (*heat_H1*) and open V4 (*ov4*). The subsequent observations in the latter scenario concerning Trace 3, i.e., *LH*, *TL* and *QL* in T1, *LL* in T2, and *FH* in C1, can be made almost instantaneously and they are all normal. Note that, after 2 additional time units, the temperature in T1 should be raised to *TH* since heater H1 has already been switched on. However, this temperature target can never be realized on Trace 3 for the following reasons: (1) f_4; (2) f_2f_4; (3) f_4f_5; (4) $f_2f_4f_5$.
- **Trace 4**: Other than the last symptom, the event sequence on Trace 3 can be observed during normal operation and this sequence also appears on Trace 4. This propagation path branches into two distinct ones after closing V1 (*cv1*), switching on heater H1 (*heat_H1*), opening V4 (*ov4*), and observing *FH* in C1. On the present trace, the normal operating conditions can still be confirmed after 2 time units. The temperature in T1 is raised to *TH* at this time as expected, while the other sensor readings remain stable, i.e., *LH* and *QL* in T1 and *LL* in T2. An abnormal state is finally reached in this scenario after another 3 time units, i.e., the sensor readings show that the operating conditions of T1 are not responding to the heat input and stay at *LH*, *TH* and *QL* indefinitely. Note that the anticipated targets for T1 should be *LM*, *THH* and *QH*. The implied fault origins of this trace should be (1) f_2 and (2) f_2f_5.
- **Trace 5**: Other than the last symptom on Trace 4, its normal event sequence can also be found on Trace 5. The common propagation path is branched after the control actions *cv1*, *heat_H1*, and *ov4*, the immediate response of sensor reading on C1 to *FH*, and the changes in sensor readings on T1 to *TH* in 2 time units later. After another 3 time units, the level and temperature readings on T1 are still normal, i.e., *LM* and *THH*, while the concentration measurement is maintained at abnormally low value (*QL*) for a sufficiently long period of time. The possible root cause for this case should be either f_7 or f_5f_7.

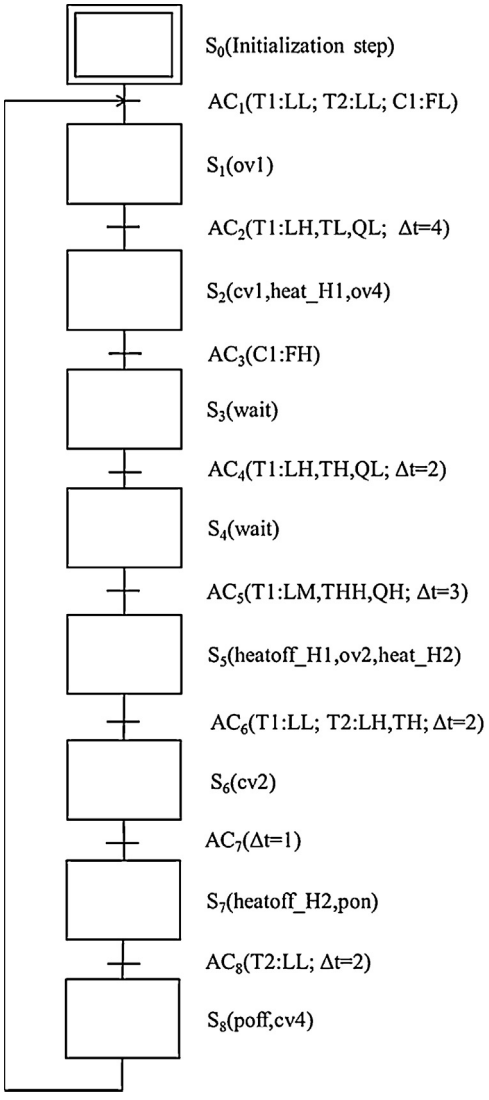

Fig. 21. Normal SFC of the batch evaporation system in Example 3.

After applying the proposed synthesis procedure to the above traces, it was found that the fault origins implied by Trace 1 cannot be further differentiated with diagnostic tests. For the sake of brevity, only the test plans of the remaining four traces are presented in Figs. 23–26 and, also, the activation conditions of these SFCs contain only the sensor readings that are affected by the test actions. A brief summary is presented in the sequel:

- **Plan 3.Tr2**: As shown in Fig. 23, the test actions *heatoff_H1* (i.e., switch off heater H1) and *ov2* (i.e., open V2) in step S_1 is executed as soon as AC_1 (Trace 2) is observed online. Four resulting activation conditions may appear: (1) AC_2 (T1:*LH*; T2:*LH*; $\Delta t = 2$); (2) AC_3 (T1:*LH*; T2:*LH*; $\Delta t = 2$); (3) AC_4 (T1:*LM*; T2:*LL*; $\Delta t = 2$); (4) AC_5 (T1:*LH*; T2:*LH*; $\Delta t = 4$). The subsequent diagnostic tests in the second stage are outlined below:
  - **AC_2** – The online test results in this case may be attributed to f_3 (i.e., controller fails to open V4) or f_3f_4 (i.e., controller fails to open V4 and also heater H1 fails). If the test actions suggested in step S_2 (i.e., *ov4*, *cv2* and *heat_H1*) are performed, there may be two possible outcomes, i.e., AC_6 (T1:*TH*; C1:*FH*; $\Delta t = 1$) and AC_7 (T1:*TL*; C1:*FH*; $\Delta t \geq 0$). A single fault origin can then be

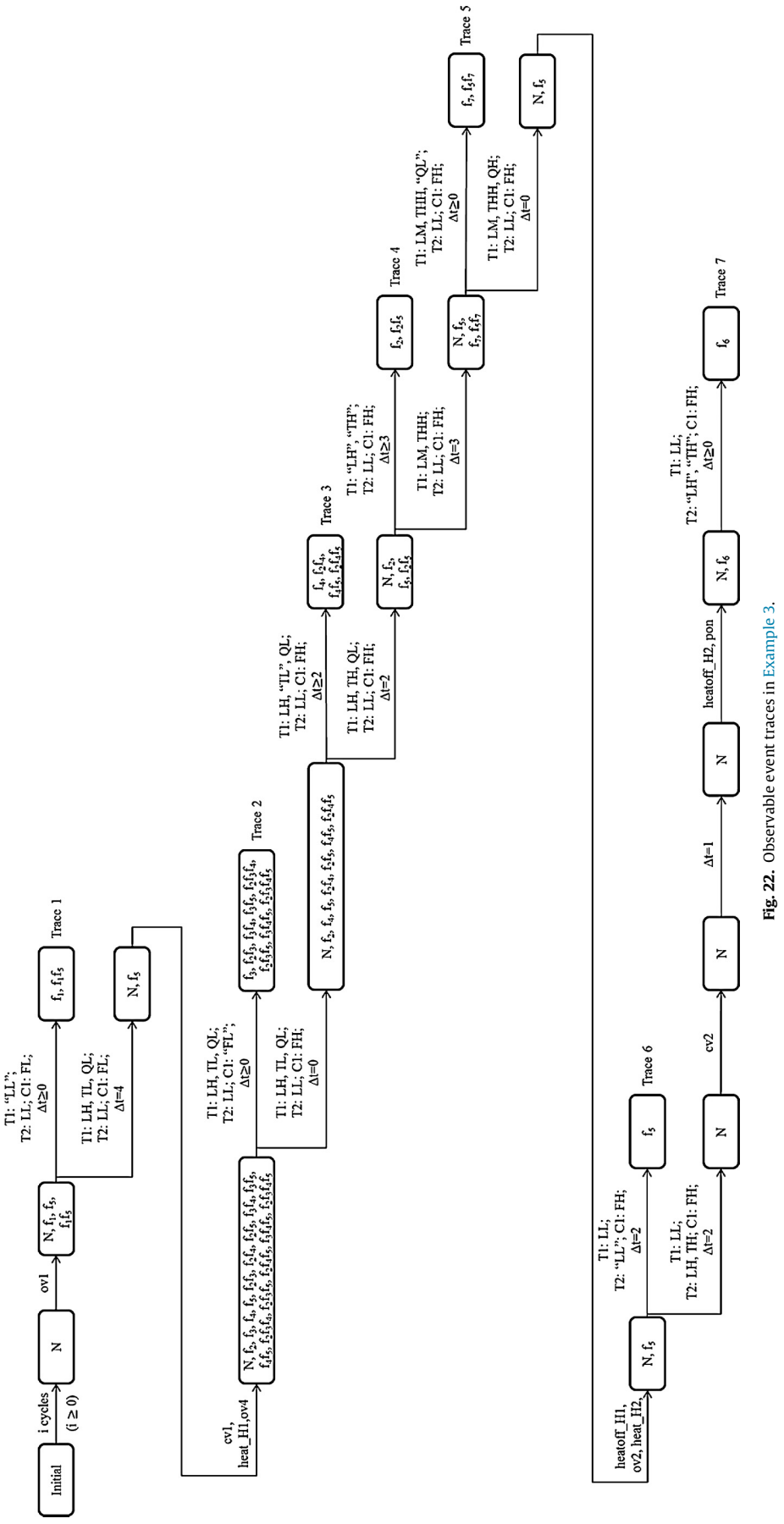

Fig. 22. Observable event traces in Example 3.

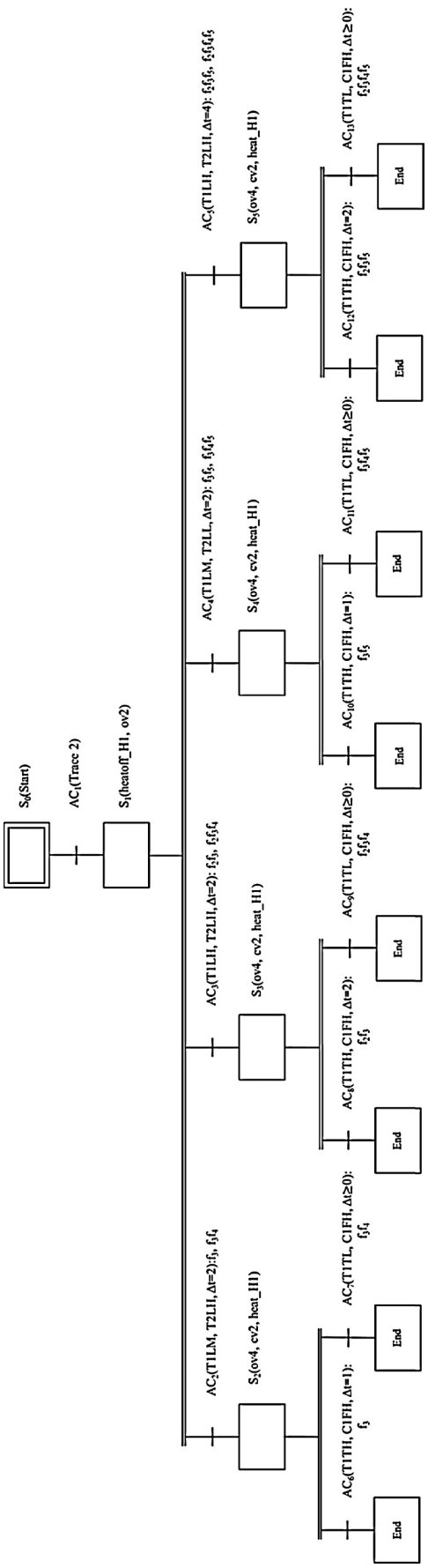

Fig. 23. Plan 3:Tr2 in Example 3.

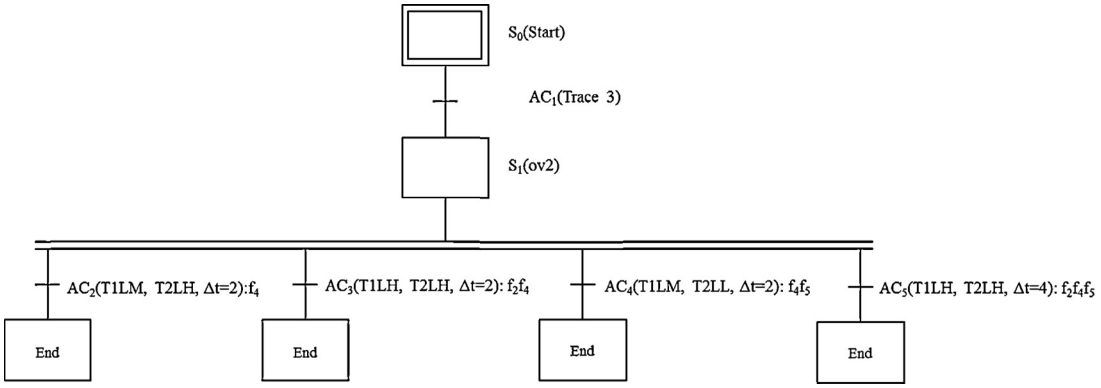

Fig. 24. Plan 3.Tr3 in Example 3.

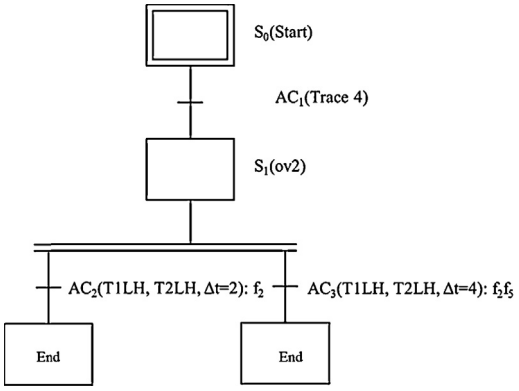

Fig. 25. Plan 3.Tr4 in Example 3.

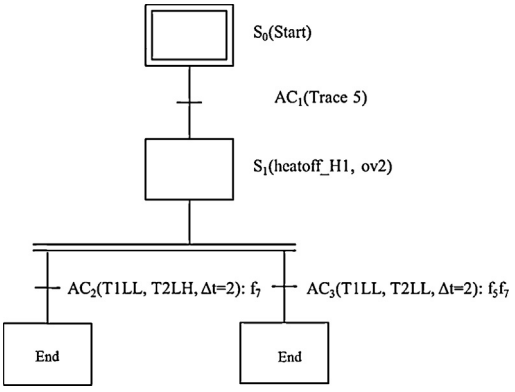

Fig. 26. Plan 3.Tr5 in Example 3.

diagnosed upon reaching each of these conditions. AC_6 indicates that the root cause is f_3, while AC_7 confirms f_3f_4.

- **AC_3** – The online test results in this activation condition may be caused by f_2f_3 (i.e., V1 sticks at the open position and controller fails to open V4) or $f_2f_3f_4$ (i.e., V1 sticks at the open position, controller fails to open V4 and also heater H1 fails). If the tests listed in step S_3 (i.e., *ov4*, *cv2* and *heat_H1*) are implemented, there may be two different responses, i.e., AC_8 (T1:*TH*; C1:*FH*; $\Delta t = 2$) and AC_9 (T1:*TL*; C1:*FH*; $\Delta t \geq 0$). AC_8 indicates that the root cause is f_2f_3, while AC_9 confirms $f_2f_3f_4$.
- **AC_4** – The online test results in this scenario are resulted from f_3f_5 (i.e., controller fails to open V4 and T2 leaks) or $f_3f_4f_5$ (i.e., controller fails to open V4, heater H1 fails and T2 leaks). If the diagnostic tests specified in step S_4 (i.e., *ov4*, *cv2* and *heat_H1*) are carried out, two sets of possible symptoms may be obtained, i.e., AC_{10} (T1:*TH*; C1:*FH*; $\Delta t = 1$) and AC_{11} (T1:*TL*; C1:*FH*; $\Delta t \geq 0$). The former suggests that the fault origin is f_3f_5, while the latter $f_3f_4f_5$.
- **AC_5** – The online test results in this case imply that $f_2f_3f_5$ (i.e., V1 sticks at the open position, controller fails to open V4 and T2 leaks) or $f_2f_3f_4f_5$ (i.e., V1 sticks at the open position, controller fails to open V4, heater H1 fails and T2 leaks). If the test actions given in step S_5 (i.e., *ov4*, *cv2* and *heat_H1*) are applied, the test results may be either AC_{12} (T1:*TH*; C1:*FH*; $\Delta t = 2$) or AC_{13} (T1:*TL*; C1:*FH*; $\Delta t \geq 0$). The former condition implies that the fault origin is $f_2f_3f_5$, while the latter $f_2f_3f_4f_5$.

- **Plan 3.Tr3**: As shown in Fig. 24, it is required to implement the test action *ov2* (i.e., open V2) in step S_1 when triggering AC_1 (Trace 3). There may be four subsequent scenarios: (1) AC_2 (T1:*LM*; T2:*LH*; $\Delta t = 2$); (2) AC_3 (T1:*LH*; T2:*LH*; $\Delta t = 2$); (3) AC_4 (T1:*LM*; T2:*LL*; $\Delta t = 2$); (4) AC_5 (T1:*LH*; T2:*LH*; $\Delta t = 4$). The corresponding fault origins are listed below:
  - **AC_2** – The corresponding test results may be attributed to a single-failure fault origin f_4 (i.e., heater H1 fails).
  - **AC_3** – The corresponding test results may be attributed to a two-failure fault origin f_2f_4 (i.e., V1 sticks at the open position and also heater H1 fails).
  - **AC_4** – The test results in this scenario are caused by a two-failure fault origin f_4f_5 (i.e., heater H1 fails and T2 leaks).
  - **AC_5** – The test results in this case imply that $f_2f_4f_5$ (i.e., V1 sticks at the open position, heater H1 fails and T2 leaks) is the sole three-failure fault origin.
- **Plan 3.Tr4**: As shown in Fig. 25, the action *ov2* (i.e., open V2) in step S_1 must be applied after activating AC_1 (Trace 4). There may be two possible outcomes, i.e., (1) AC_2 (T1:*LH*; T2:*LH*; $\Delta t = 2$) and (2) AC_3 (T1:*LH*; T2:*LH*; $\Delta t = 4$). The corresponding diagnosis can be summarized below:
  - **AC_2** – The corresponding test results may be attributed to a single-failure fault origin f_2 (i.e., V1 sticks at the open position).
  - **AC_3** – The test results in this case imply that f_2f_5 (i.e., V1 sticks at the open position and T2 leaks) is the root cause.
- **Plan 3.Tr5**: As shown in Fig. 26, the test operations *heatoff_H1* (i.e., switch off H1) and *ov2* (i.e., open V2) in step S_1 must be performed when observing AC_1 (Trace 5). There may be two possible responses, i.e., (1) AC_2 (T1:*LL*; T2:*LH*; $\Delta t = 2$) and (2) AC_3 (T1:*LL*; T2:*LL*; $\Delta t = 2$). The corresponding diagnosis results are outlined below:
  - **AC_2** – The corresponding test results may be attributed to a single-failure fault origin f_7 (i.e., concentration analyzer on T1 fails).
  - **AC_3** – The test results in this case suggest that the root cause is a two-failure fault origin f_5f_7 (i.e., T2 leaks and also concentration analyzer on T1 fails).

6. Conclusions

A standardized methodology has been proposed in this work to systematically construct timed automata for modelling all components in any given batch plant, and to enumerate the observable event traces accordingly. A generic synthesis procedure has also been developed for conjecturing the test plans of all undiagnosable traces. It should be noted that the proposed method is capable of differentiating various time delays caused by fault origins of the same type but with different intensities. This is a unique feature which has never been developed in the past. Extensive case studies have been carried out to demonstrate the feasibility and effectiveness of the proposed procedure synthesis strategy.

References

Baroni P, Lamperti G, Pogliano P, Zanella M. Diagnosis of large active systems. Artif Intell 1999;110:135–83.

Baroni P, Lamperti G, Pogliano P, Zanella M. Diagnosis of a class of distributed discrete-event systems. IEEE Trans Syst Man Cybern A: Syst Hum 2000;30:731–52.

Bauer N, Engell S, Huuck R, Lohmann S, Lukoschus B, Remelhe M, et al. Verification of PLC programs given as sequential function charts. In: Integration of software specification techniques for applications in Engineering. Berlin: Springer; 2004. p. 517–40.

Behrmann G, David A, Larsen KG. A tutorial on UPPAAL 4.0. Denmark: Aalborg University; 2006.

Benveniste A, Fabre E, Haar S, Jard C. Diagnosis of asynchronous discrete-event systems: a net unfolding approach. IEEE Trans Autom Control 2003;48:714–27.

Cassandras CG, Lafortune S. Introduction to discrete event systems. Boston: Kluwer Academic Publisher; 1999.

Chen YC, Yeh ML, Hong CL, Chang CT. Petri-net based approach to configure online fault diagnosis systems for batch processes. Ind Eng Chem Res 2010;49:4249–68.

Clarke EM, Emerson A, Sistla KL. Automatic verification of finite-state concurrent systems using temporal logic specification. ACM Trans Program Lang Syst 1986;8:244–63.

Dai Y, Zhao J. Fault diagnosis of batch chemical processes using a dynamic time warping (DTW)-based artificial immune system. Ind Eng Chem Res 2011;50:4534–44.

Debouk R, Lafortune S, Teneketzis D. Coordinated decentralized protocols for failure diagnosis of discrete event systems. Discret Event Dyn Syst Theory Appl 2000;10:33–86.

Gascard E, Simeu-Abazi Z. Modular modeling for the diagnostic of complex discrete-event systems. IEEE Trans Autom Sci Eng 2013;10:1101–23.

Ghosh K, Srinivasan R. Immune-system-inspired approach to process monitoring and fault diagnosis. Ind Eng Chem Res 2011;50:1637–51.

Kang A, Chang CT. Automata generated test plans for fault diagnosis in sequential material-and energy-transfer operations. Chem Eng Sci 2014;113:101–15.

Kim J, Moon I. Automatic verification of control logics in safety instrumented system design for chemical process industry. J Loss Prev Proc Ind 2009;22:975–80.

Kim J, Moon I. Model checking for automatic verification of control logics in chemical process. Ind Eng Chem Res 2011;50:905–15.

Kourti T, Macgregor JF. Process analysis, monitoring and diagnosis, using multivariate projection methods. Chemom Intell Lab Syst 1995;28:3–21.

Kourti T, Nomikos P, Macgregor JF. Analysis monitoring and fault-diagnosis of batch processes using multiblock and multiway PLS. J Proc Cont 1995;5:277–84.

Lahtinen J, Valkonen J, Björkman K, Frits J, Niemelä I, Heljanko K. Model checking of safety-critical software in the nuclear engineering domain. Rel Eng Syst Saf 2012;105:104–13.

Lee JM, Yoo CK, Lee IB. Fault detection of batch processes using multiway kernel principal component analysis. Comput Chem Eng 2004;28:1837–47.

Li JH, Chang CT, Jiang D. Systematic generation of cyclic operating procedures based on timed automata. Chem Eng Res Des 2014;92:139–55.

Lohmann S, Stursberg O, Engell S. Systematic design of logic controllers for processing plants starting from informal specifications. Comput Aided Chem Eng 2006;21:1317–22.

Nomikos P, MacGregor JF. Monitoring batch processes using multiway principal component analysis. AIChE J 1994;40:1361–75.

Nomikos P, MacGregor JF. Multivariate SPC charts for monitoring batch processes. Technometrics 1995;37:41–59.

Qiu WB, Kumar R. Decentralized failure diagnosis of discrete event system. IEEE Trans Syst Man Cybern A: Syst Hum 2006;36:384–95.

Sampath M, Lafortune S, Teneketzis D. Active diagnosis of discrete-event systems. IEEE Trans Autom Control 1998;43:908–29.

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis DC. Diagnosability of discrete-event systems. IEEE Trans Autom Control 1995;40:1555–75.

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis DC. Failure diagnosis using discrete-event models. IEEE Trans Control Syst Technol 1996;4:105–24.

Tan WL, Nor NM, Abu Bakar MZ, Ahmad Z, Sata SA. Optimum parameters for fault detection and diagnosis system of batch reaction using multiple neural networks. J Loss Prev Proc Ind 2012;25:138–41.

Undey C, Ertunc S, Cinar A. Online batch fed-batch process performance monitoring, quality prediction, and variable contribution analysis for diagnosis. Ind Eng Chem Res 2003;42:4645–58.

Yeh ML, Chang CT. An automaton-based approach to evaluate and improve online diagnostic schemes for multi-failure scenarios in batch processes. Chem Eng Res Des 2011;89:2652–66.

Zad SH, Kwong RH, Wonham WM. Fault diagnosis in discrete-event systems: framework and model reduction. IEEE Trans Autom Control 2003;48:1199–204.

Zhao C. Quality-relevant fault diagnosis with concurrent phase partition and analysis of relative changes for multiphase batch processes. AIChE J 2014;60:2048–62.