15th Triennial World Congress, Barcelona, Spain

DIAGNOSIS OF DISCRETE–EVENT SYSTEM DESCRIBED BY TIMED AUTOMATA

Jan Lunze * **Peerasan Supavatanakul** *,[1]

* *Ruhr–University Bochum,*
Institute of Automation and Computer Control,
D–44780 Bochum, Germany

Abstract: This paper presents the model–based diagnosis in the framework of discrete–event systems. Timed automata are used as a discrete–event representation which is suitable for consistency–based diagnosis. The diagnostic statement is based on the observation whether the measured event–time sequences are consistent with the timed automata. The diagnostic algorithm can be applied online because it determines the fault occurrence recursively for the measured event–time sequences. The result is applied to diagnose valve faults in a chemical process.

Keywords: discrete–event systems, modelling, timed automata, consistency–based diagnosis

1. INTRODUCTION

The task of general fault diagnosis is to decide if faults have occurred in the system and to identify them. Various systematic approaches for diagnosis have been gradually elaborated in the field of control engineering and artificial intelligence, cf for surveys (Hamscher *et al.*, 1992; Patton *et al.*, 1989). For most approaches, it is assumed that the signals from the system can be measured numerically precisely and the system can be represented by exact mathematical model. The diagnostic task can then be carried out by identifying the deviation of system states or parameters (Isermann, 1984).

In contrast, this paper concerns diagnosis of discrete–event systems as shown in Figure 1. The diagnosis deals with sequences of events, which contain enough information to discriminate the correct and faulty behaviours. The diagnosis of discrete–event systems has yet to be elaborated in detail. The early work has been reported in (Sampath *et al.*, 1995) which investigates the diagnosability of the discrete–event system described by automata, while (Lunze and Schröder, 2001; Förstner, 2001) investigate the diagnosis based on stochastic automata and nondeterministic automata respectively.

As an extension to these approaches, the diagnosis proposed is based on the timed discrete–event representation in the form of timed automata. The main motivation for dealing with timed discrete–event representation is that the temporal distance between events includes important information for diagnosis. For example, the degradation of systems due to a fault changes first the temporal behaviour between events and then the event sequence. (Lunze, 2000) has addressed the diagnosis based on the timed discrete–event representation using a Semi–Markov process but the method proposed here uses a coarser model and has, therefore, lower complexity.

This paper concerns the dynamic system that can be described by differential equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), f), \qquad \mathbf{x}(0) = \mathbf{x}_0 \qquad (1)$$

where the behaviour of the state vector $\mathbf{x} \in \mathbb{R}^n$ depends on the input vector $\mathbf{u} \in \mathbb{R}^m$ and the fault $f \in \mathbb{R}^s$ which occur in the system.

Since technological systems usually have restriction on the measurability of the signal values and many

[1] This work is supported by Deutsche Forschungsgemeinschaft (LU462/13).

signals cannot be precisely measured, e.g. the system is equipped with discrete level sensors so signals from the system can only be measured qualitatively such as "low", "medium", and "high". Thus the system can be regarded as a "*quantised system*" as described by (Lunze, 1994; Lunze *et al.*, 2001).

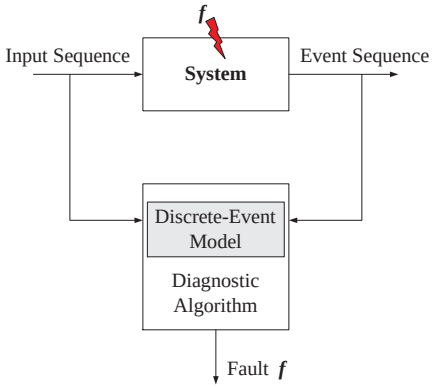

Fig. 1. Diagnosis of Discrete–Event System

This paper consists of two main parts. First, Section 2 deals with modelling of the discrete–event system using timed automata and second, Section 3 with the solution of the diagnostic problem based on this representation. Section 4 shows how a timed automaton can be set up for a given system. Finally, the approach presented is used for the diagnosis of valve faults in a chemical process.

2. TIMED AUTOMATA

Automata are nondeterministic finite state machines. They play an important role in discrete–event system theory. Timed automata are nondeterministic automata with timed transition behaviour. They are used as timed discrete–event representation of dynamic systems. For diagnostic purposes the timed automata have to include the dependence upon the fault f.

A *timed automaton* is described by

$$\mathcal{A}_T(\mathcal{N}_z, \mathcal{N}_v, \mathcal{N}_f, R, z_0) \tag{2}$$

where $\mathcal{N}_z$ denotes the set of automaton states, $\mathcal{N}_v$ the set of automaton inputs, and $\mathcal{N}_f$ the set of faults. R represents the state transition relation of the timed automaton and z_0 its initial state.

The state transition relation R of the timed automaton describes the dynamic behaviour of the automaton and is represented by

$$R = (L, T). \tag{3}$$

R consists of two main components. The two components together constitute the dynamic of the automaton:

(1) The state transition L of the automaton is described by the following relation:

$$L : \mathcal{N}_z \times \mathcal{N}_z \times \mathcal{N}_v \times \mathcal{N}_f \longrightarrow \{0, 1\}. \tag{4}$$

For $L(z', z, v, f) = 1$ the automaton can step from current state z to successor state z' if input v has been applied and the fault f is present.

(2) The temporal function T represents the following mapping

$$T : \mathcal{N}_z \times \mathcal{N}_z \times \mathcal{N}_v \times \mathcal{N}_f \to \mathbb{R}^+ \times \mathbb{R}^+. \tag{5}$$

T describes the *sojourn time* $\tau(z)$ of each automaton state $z \in \mathcal{N}_z$ before it moves to the successor state $z' \in \mathcal{N}_z$ under the influence of the input $v \in \mathcal{N}_v$ and fault $f \in \mathcal{N}_f$. $\tau(z)$ is always nonnegative and increases continuously. The temporal function T of one automaton state z can then be described with the time interval $[\tau_{\min}(z), \tau_{\max}(z)]$ where $\tau_{\min}(z)$ and $\tau_{\max}(z)$ denote the lower and upper bound of the time that passes after the state z but before the successor state z' in the timed automaton respectively.

The movement of the timed automaton from one state to another is described by R and is possible only if the following conditions hold true:

$$L \in \mathcal{N}_z \times \mathcal{N}_z \times \mathcal{N}_v \times \mathcal{N}_f \tag{6}$$

$$\tau(z) \in [\tau_{\min}(z), \tau_{\max}(z)]. \tag{7}$$

The first condition (6) means that the transition from state $z \in \mathcal{N}_z$ to state $z' \in \mathcal{N}_z$ is possible under the application of some input $v \in \mathcal{N}_v$ and the presence of some fault $f \in \mathcal{N}_f$. The second condition (7) implies that if a transition from one state z to its successor z' occurs, it must occur within the time $[\tau_{\min}(z), \tau_{\max}(z)]$ where z is the current state of the timed automaton.

Unlike the Semi–Markov process considered in (Lunze, 2000), the temporal function T of a timed automaton has no probability distribution. This means that it is equally likely that the automaton can move from the current state z to successor state z' at any time $t \in [\tau_{\min}(z), \tau_{\max}(z)]$. So the representation of a system by a timed automaton is simpler than the representation by a Semi–Markov process.

If the timed automaton is generalised for explicit discrete–event system representation, all states of the timed automaton $z \in \mathcal{N}_z$ can be replaced with event $e \in \mathcal{N}_e$. Thus

$$\mathcal{N}_e = \mathcal{N}_z = \bigcup_{i=1\ldots n} \{e_i \in \mathcal{N}_e\} \tag{8}$$

where n denotes the number of possible events in the given dynamic system. Then, all the preceding formulations can be written with e replacing z and the timed automaton for the representation of the discrete–event system is described by

$$\mathcal{A}_T(\mathcal{N}_e, \mathcal{N}_v, \mathcal{N}_f, R, e_0). \tag{9}$$

where e_0 is the initial event of the automaton.

In this case the states of the timed automaton correspond explicitly to the events in the discrete–event systems.

3. DIAGNOSIS OF TIMED AUTOMATA

For diagnosis, it is assumed that an unknown fault $f \in \mathcal{F} = \{f_0, f_1, \ldots, f_{n_F}\}$, where n_F is the number of faults considered, has occurred at time $t \leq 0$ and is present until the diagnostic algorithm is stopped. The measured event–time sequence during the time interval $[0, t_h]$ is denoted by $\mathbf{E}(0 \ldots t_h) = \{e_0, t_0; \ldots; e_h, t_h\}$. The input to the system is measured simultaneously with the occurrence of event, consequently the input–time sequence is denoted by $\mathbf{V}(0 \ldots t_h) = \{v_0, t_0; \ldots; v_h, t_h\}$. Thus the system is considered to have "synchronised I/O events" (Förstner, 2001). The main idea of *consistency–based diagnosis* is to answer the question:

> Can the system generate $\mathbf{E}(0 \ldots t_h)$ upon receiving $\mathbf{V}(0 \ldots t_h)$ during $[0, t_h]$?

For the timed automaton (9), it has to be tested whether event–time sequence generated by the system with the input–time sequence is consistent with the transition relation R of the timed automaton.

The diagnosis starts with no information about occurrence of fault. Therefore all faults $f \in \mathcal{F}$ may have occurred. In addition, the initial event e_0 of the system under consideration is assumed to be known. The diagnostic algorithm determines the occurrence of faults $P(f, t_h)$, for increasing time horizon, recursively for given $P(f, t_{h-1})$ as follows:

$$P(f, t_h) = \begin{cases} 1 \text{ if } P(f, t_{h-1}) = 1 \\ \quad \text{and } \mathbf{E}(0 \ldots t_h) \in R \\ \\ 0 \text{ otherwise.} \end{cases} \tag{10}$$

The first part of (10) concerns the case that the fault f cannot be excluded by using the observed event–time sequence from $t = 0$ to $t = t_{h-1}$. Then it is tested whether the newly observed event–time is consistent with the transition relation R of the timed automaton $\mathcal{A}_T$. The second condition of (10) says that the fault f is not the possible candidate for the diagnostic result at the time t_h if it has been previously excluded due to the inconsistency with the timed automaton modelled for fault f at time t_{h-1}. $P(f, t_h)$ can be determined for one observed event after another. Thus this diagnostic method can be used online.

Algorithm 1 summarises fault diagnostic method of the system described by timed automata. The main idea of the algorithm is to determine the set of fault candidates $\mathcal{F}(t_h) = \{f : P(f, t_h) \neq 0\}$ at time t_h from the fault set $\mathcal{F}$ which enables the movement of the timed automaton. This follows the idea of consistency–based diagnosis which means to exclude fault for an increasing time horizon. In the main loop of the algorithm (10) is applied. This is done at every occurrence of new event. In Step 1 of the algorithm, the time horizon t_h is updated at the time the next event has occurred.

Algorithm 1. *Diagnosis of timed automata*

Given: Timed Automata $\mathcal{A}_T$ for all $f \in \mathcal{F}$
Initial event e_0
Time horizon t_H

Initialise: $t_h = 0$, $P(f, 0) = 1$ for all $f \in \mathcal{F}$

1. Wait for next event e, measure event e, input v, occurrence time t_h
2. Determine the relation $P(f, t_h)$ according to (10)
3. Determine $\mathcal{F}(t_h)$ from $P(f, t_h)$ at time t_h:
   3.1 If $P(f, t_h) = 0$, then $f \notin \mathcal{F}(t_h)$
   3.2 If $P(f, t_h) = 1$, then $f \in \mathcal{F}(t_h)$
4. If $t_h \leq t_H$, go to Step 1

Result: $P(f, t_h)$ and possible set of fault $f \in \mathcal{F}(t_h)$ for increasing t_h

The diagnosis based on the timed automata yields the following results:

- *Fault detection:* If $P(f_0, t_h) = 0$ holds (where f_0 symbolises the faultless case), then some fault must have occurred in the system.
- *Fault identification:* If $P(f, t_h) = 0$ holds, the system has not been affected by the fault f.

Fault detection shows an important aspect of the consistency–based diagnosis. Even if only a model of the faultless system were available, fault detection would still be possible.

Fault identification by means of consistency based diagnosis means to exclude those faults that, according to the available information, are known not to have occur. This means that if the observed behaviour is inconsistent with the model for a certain fault f, then the fault f can be excluded as the primary reason for the faulty behaviour. Therefore (10) gives a basis for choosing the probable fault f from the set of possible fault set $\mathcal{F}$.

4. DETERMINATION OF TIMED AUTOMATA

This section shows how a timed automaton can be set up for a given dynamic system. Since the system under consideration has restriction on measurability of input and output, it can be considered as a "*quantised system*" whose structure is shown in Figure 2 (Lunze, 1994; Teneketzis *et al.*, 1994). The system under consideration is the continuous–variable system described by (1). The quantisers represent the partition of the state and input signals. For example the quantiser on the right hand side of Figure 2 introduces mapping of state space partition $\mathbb{R}^n$ into finite number of disjoint sets $\mathcal{Q}_x(z(t))$, which denote the set of states $\mathbf{x}(t) \in \mathbb{R}^n$ with the same qualitative value $z(t)$.

The mapping invoked by the quantiser is symbolised by $[\cdot]$.

$$[\mathbf{x}(t)] = z(t) \Leftrightarrow \mathbf{x}(t) \in \mathcal{Q}_x(z(t)) \qquad (11)$$

The quantisation of the input signals $\mathbf{u}(t)$ by the quantiser on the left hand side is analogous to the quantisation of the state space.

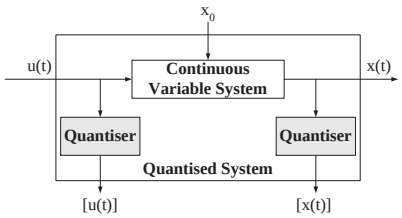

Fig. 2. Quantised System

Figure 3 shows the state space partition of the chemical process in Section 5. It is obtained from the discrete level sensors mounted on reactors $R1$ and $R2$ The state variables x_1 and x_2 correspond to the level of the solution in reactors. The two states are partitioned independently and contribute to 9 quantised states. The grey box represents the initial state of the system which lies in $\mathcal{Q}_x(1)$, and corresponds to low solution level in both reactors.

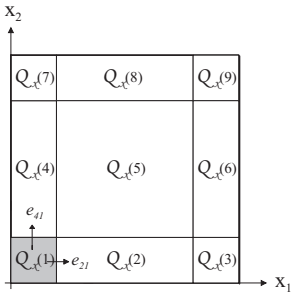

Fig. 3. Signal partitioning of a two dimensional state space

If the "*border*" between partitions is defined as

$$\delta\mathcal{Q}_{xij} = \delta\mathcal{Q}_x(x(i)) \cap \delta\mathcal{Q}_x(x(j)) \qquad (12)$$

where $\delta\mathcal{Q}_x(x(i))$ denotes the hull of $\mathcal{Q}_x(x(i))$. If the state trajectory $\mathbf{x}(t)$ crosses such borders, an "*event*" is generated. Figure 3 shows the occurrence of events e_{41} and e_{21} with the corresponding borders between partitions.

In addition the quantiser determines time t_k when an event e_{ij} is generated if $[\mathbf{x}(t_k + \delta t)] = \mathcal{Q}_x(x(i))$ and $[\mathbf{x}(t_k - \delta t)] = \mathcal{Q}_x(x(j))$ for $\mathcal{Q}_x(x(i)) \neq \mathcal{Q}_x(x(j))$ and small $\delta t > 0$.

Because quantised systems generate events and their occurrence time, timed automata are thus a suitable timed discrete–event representation of the quantised system. The construction of the timed automaton is to determine the transition relation R as defined in (3). Algorithm 2 is proposed for setting up the timed automaton.

The algorithm starts with the transition relation $R = 0$ ($L = 0$ and $T = 0$). Then it determines the case where $L(e', e, v, f) = 1$ with respect to the possible transitions to the successor events. To determine such transitions, at first the set of compatible successor events for the current event $e \in \mathcal{N}_e$ is determined. A check is then made to ensure that such event can occur in the given system. If the transition is possible ($L(e', e, v, f) = 1$), the algorithm then determines the the duration between the two successive events or sojourn time $\tau(e)$, which is finally appended to the temporal function T to get all possible sojourn time. Finally the current event is set to the new event and time to the latest event occurrence time, and the process is repeated.

Algorithm 2. *Abstraction algorithm*

Given: System (1), quantisers (11)
Time horizon t_H
Current event e = Initial event e_0

Initialise: Transition relation $R = 0$
Start time horizon $t = 0$

1. **Determine** transitions from current event e to new event e':
For (e', e, v, f) with e',$e \in \mathcal{N}_e$, $v \in \mathcal{N}_v$, $f \in \mathcal{N}_f$
1.1 **Construct** e' from e by appending e' to L.
1.2 **Check** the transition of e to e':
If (1) and (11) can generate e'
then $L(e', e, v, f) = 1$.
2. **Determine** elapsed time between e and e' for all (e', e, v, f)
2.1 **Construct** sojourn time $\tau(e)$ for event e, $\tau(e) = [\tau_{\min}(e), \tau_{\max}(e)]$.
2.2 **Append** $\tau(e)$ for e to T.
3. **Set** $t := t(e')$ and $e = e'$.
4. If $t \leq t_H$, go to Step 1.

Result: Automaton transition relation R

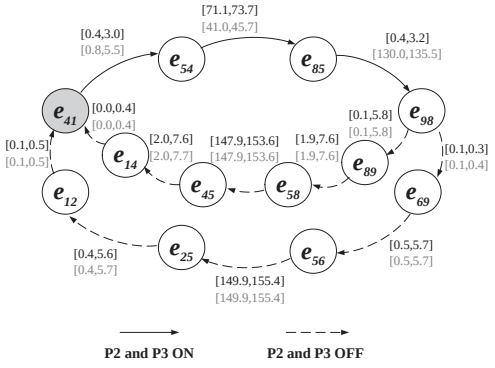

Fig. 4. Automaton graph of chemical process

Figure 4 shows an automaton graph of the chemical process considered in Section 5. The graph nodes represent events which occur in one cycle of the process starting from the initial event e_{41}. It is also possible to have e_{21} as the initial event but this is not shown for clarity of representation. The label above each arc

denotes the time between event (sojourn time). In the figure, the black numbers correspond to the faultless case while the grey numbers represent the sojourn time of the events in the chemical process with fault in V_2. Note that the blockage of V_2 does not affect the outflow. This automaton is used in the next section for diagnosing valve faults in the chemical process.

5. EXAMPLE

In this section an example is discussed to demonstrate the effectiveness of diagnostic algorithm described in the preceding section.

The system under consideration is the chemical process given in Figure 5 (Hanisch, 1992). The product of the process results from a reaction between a substance and a solvent in the reactors $R1$ and $R2$. A certain amount of substance is pumped by pumps P_2 and P_3 through valves V_1 and V_5 into $R1$ and $R2$ respectively. A certain quantity of solvent is provided to $R1$ and $R2$ from tank $B1$ through the valves V_2 and V_6. The process starts once the substance and the solvent are mixed in the reactors. The solution after reaction flows through the valves V_4 and V_8 into the tank $B2$. Finally the solution from $B2$ flows to the filter $F1$ where the solvent is separated from the final product. The solvent is then pumped back to $B1$ by P_1.

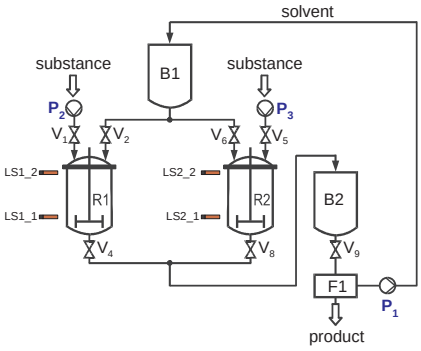

Fig. 5. A chemical process

The discrete levels from both reactors detected by the discrete level sensors as shown by LS in Figure 5 are the only available measurement information. They act as quantisers as explained in Section 4. Thus the available information from these sensors tells only whether each reactor is full, medium or empty. The possible faults $f \in \mathcal{F} = \{f_0, f_1, f_2, f_3\}$ can occur in the system, where

- f_0 : faultless case
- f_1 : blockage of Valve V_2
- f_2 : blockage of Valve V_4
- f_3 : blockage of Valve V_8.

The blockage of these valves are to be diagnosed by means of a discrete–event model described by timed automata.

Timed automata with different event–time sequence are obtained for the system subjected to each fault $f \in \mathcal{F}$ as described by Algorithm 2. Instead of an automaton graph, the event-time sequences from abstraction obtained for the initial event $e_0 = e_{41}$ are shown in Figure 6. Four different parts of the figure correspond to event–time behaviour of the system subjected to four different faults. The y–axis of each part is labelled with events which can occur in the chemical process provided that the initial event in the process is e_{41}. The time $t = 0$ denotes the occurrence of e_{41}. It is clear from the figure that each fault affects the chemical process in a different way.

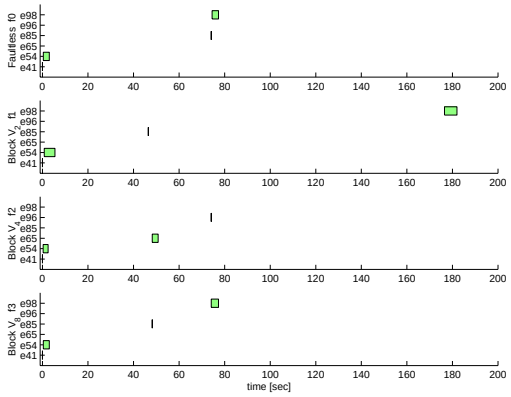

Fig. 6. Event–time sequences of the chemical process for all faults $f \in \mathcal{F}$

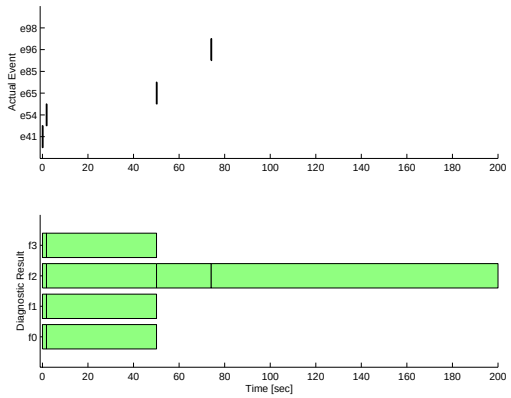

Fig. 7. Diagnostic result of the chemical process with the presence of f_2

The diagnosis method described in Section 3 is applied to detect the blockage of valves V_4 (f_2). Figure 7 represents the diagnostic results of the chemical process subjected to the blockage of valve V_4. The dark ticks in the upper part of the figure illustrate the actual occurrence of events in the chemical process and the corresponding occurrence time instant. The diagnostic algorithm has no information about the occurrence of the fault in the system so it is assumed that all faults can occur in the system once the initial event e_{41} has taken place. It can be seen that the first two event occurrences in the process are consistent with the model for all faults $f \in \mathcal{F}$ by comparing the upper part of Figure 7 with the events and their corresponding time for all faults in Figure 6. Therefore no fault can be excluded. After the occurrence of the

third event e_{65}, only the automaton of the system subjected to the blockage of V_4 (f_2) is consistent with the measurement. Therefore all other faults are excluded. Since the faults f_0, f_1, and f_3 are excluded, the next cycle of the diagnostic algorithm will not consider the occurrence of these faults. The fourth event also confirms the blockage of V_4. Therefore it can be concluded that the valve V_4 is blocked. This is shown in the diagnostic result in the lower part of Figure 7. Note that the blockage of V_4 can be detected because the inconsistencies of event occurrence of the actual events with the timed automata which represent the system subjected to fault f_0, f_1 and f_3. The system subjected to these faults cannot produce the event e_{65} as can be seen in Figure 6.

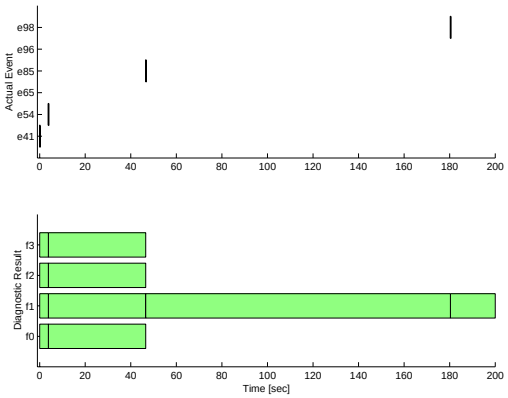

Fig. 8. Diagnostic result of the chemical process with the presence of f_1

Consider now the blockage of valve V_2 (f_1) with the same initial event e_{41}. The measurement of the actual events is shown in upper part of Figure 8. Again it can be seen that the first two events are consistent with all the models of the system subjected to all faults $f \in \mathcal{F}$. However when the third event e_{85} occurrs, the algorithm is able to detect the blockage of valve V_2 despite that the same event could have occurred also for the system subjected to all other faults $f \neq f_2$. This is because the temporal distance between events of other fault cases does not agree with the measurement. Therefore these faults can be excluded. This is shown in Figure 8 where the diagnostic algorithm says that the only possible fault in the chemical process is the blockage of valve V_2 after the occurrence of the third event e_{85}. The fourth event e_{98} also confirms this diagnostic result. Hence the diagnostic algorithm can detect the fault f_1 in the chemical process by using the temporal information.

Note that only a single–fault case is considered in this example. However, Algorithm 1 can also be used to detect multiple faults, and in such cases the modelling method described in Section 4 can be applied with f denoting combinations of faults.

6. CONCLUSIONS

The paper demonstrates a diagnostic method for dynamic systems, for which only the event- and input–time sequences are available as on–line information. This diagnostic approach uses timed automata as the representation of the discrete–event system. Timed automata can be obtained by abstraction from the quantitative description of the system as explained in Section 4. For diagnosis, the event- and input–time sequences are used to test for consistency with the timed automata for all faults $f \in \mathcal{F}$. If the inconsistency with the automaton setup for the faultless system occurs, a fault is detected. It is shown by example of a chemical process that faults can be detected shortly after their occurrence due to the change of temporal distance or event order. The diagnostic results can be improved by extending the consistency check for every sampling time step instead of every occurrence of event. In such case faults which are not consistent with the measurement will be excluded earlier, for similarity see (Lunze, 2000).

7. REFERENCES

Förstner, D. (2001). Qualitative Modellierung für die Prozeßdiagnose und deren Anwendung auf Dieseleinspritzsysteme. PhD thesis. TU Hamburg-Harburg.

Hamscher, W., L. Console and J. (eds) de Kleer (1992). *Readings in Model-Based Diagnosis*. Morgan Kaufmann.

Hanisch, H. (1992). *Petri–Netze in der Verfahrentechnik: Modellierung und Steuerung verfahrenstechnischer Systeme*. Oldenbourg.

Isermann, R. (1984). Process fault detection based on modeling and estimation methods - a survey. *Automatica* **20**, 387–404.

Lunze, J. (1994). Qualitative modelling of linear dynamical systems with quantised state measurements. *Automatica* **30**(3), 417–431.

Lunze, J. (2000). Diagnosis of quantised systems based on a timed discrete-event model. *IEEE Transactions on Systems, Man, and Cybernetics Part A* **SMC-30**(3), 322–335.

Lunze, J. and J. Schröder (2001). State observation and diagnosis of discrete-event systems described by stochastic automata. *Discrete Event Dynamic Systems: Theory and Applications*.

Lunze, J., J. Schröder and P. Supavatanakul (2001). Diagnosis of discrete event systems: the method and an example. In: *Proceedings of the Workshop on Principles of Diagnosis, DX'01*. pp. 111–118.

Patton, R. J., P. M. Frank and R. N. Clark (1989). *Fault Diagnosis in Dynamic Systems Theory and Application*. Prentice Hall New York.

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

Teneketzis, D., K. Sinnamohideen, M. Sampath, R. Sengupta and S. Lafortune (1994). Failure diagnosis using discrete event models. In: *Proceeding Conference on Decision and Control*. pp. 3110–3116.