

# Fault Diagnosis of Timed Discrete Event Systems

Chao Gao<sup>1</sup>, Dimitri Lefebvre<sup>2</sup>, Carla Seatzu<sup>3</sup>, Zhiwu Li<sup>4</sup>, and Alessandro Giua<sup>5</sup>

## *Abstract*

In this paper, we consider partially observable timed discrete event systems (DESs) endowed with a single clock that is reset at each event occurrence. A time interval with integer bounds is associated with each transition specifying at which clock values it may occur. This work deals with the fault diagnosis problem of such timed DESs, assuming that faulty behaviours are described by means of timed transitions. We present a zone automaton that provides a purely discrete event description of the behaviour of the timed DES with faults and construct a fault recognizer as the parallel composition of the zone automaton with a fault monitor that recognizes the occurrence of faults. The diagnosis approach allows one to compute the diagnosis state for each timed observation, which consists in a timed sequence of observed events.

## *Index Terms*

Discrete event system, timed discrete event system, observation, diagnosis.

## *Published as:*

C. Gao, D. Lefebvre, C. Seatzu, Z. Li and A. Giua, "Fault Diagnosis of Timed Discrete Event Systems," In *Proceedings of 22nd IFAC World Congress* (Yokohama, Japan), July 9-14, 2023.

---

<sup>1</sup> Chao Gao is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China; DIEE, University of Cagliari, Cagliari 09124, Italy, E-mail: [gaochao@stu.xidian.edu.cn](mailto:gaochao@stu.xidian.edu.cn).

<sup>2</sup> Dimitri Lefebvre is with GREAH Laboratory, Normandy University, 75 rue Bellot, Le Havre 76600, France, E-mail: [dimitri.lefebvre@univ-lehavre.fr](mailto:dimitri.lefebvre@univ-lehavre.fr).

<sup>3</sup> Carla Seatzu is with DIEE, University of Cagliari, Cagliari 09124, Italy, E-mail: [carla.seatzu@unica.it](mailto:carla.seatzu@unica.it).

<sup>4</sup> Zhiwu Li is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China; the Institute of Systems Engineering, Macau University of Science and Technology, Macau, E-mail: [zhwli@xidian.edu.cn](mailto:zhwli@xidian.edu.cn).

<sup>5</sup> Alessandro Giua is with DIEE, University of Cagliari, Cagliari 09124, Italy, E-mail: [giua@unica.it](mailto:giua@unica.it).

## 1. INTRODUCTION

The notion of time is fundamental in systems modeling and control. Within the domain of automatic control, a major focus is on *time-driven systems* described by difference or differential equations, where time plays the role of an independent variable. In the context of *discrete event systems* (DESs), associating a timing structure to a purely logical model allows one to characterize its performance and solve related optimization problems (Cassandras et al. (2009)). In this paper, timed DESs are considered, where a timing structure is treated as a set of additional constraints that the system's evolution needs to satisfy. We also assume that only a subset of the events is observable, namely a sensor is associated with such events, while the other events are unobservable since no sensors are deployed in the system to reveal their occurrences. Our main interest is to show how the knowledge of the timing structure and the knowledge of the time instants in which observable events occur, can be exploited for the purpose of fault diagnosis, i.e., determining if certain events (faults) could have occurred or must have occurred given an observation executed by a timed DES.

Timed automata, introduced by Alur et al. (1994), are a basic event-based model endowed with a finite set of clock variables that can be updated by the occurrence of events; the occurrence of events, in turn, depends on the current values of the clocks. This model provides a convenient framework for appropriately representing and efficiently reasoning about cyber-physical systems subject to real-time constraints, and has become a standard model for real-time systems (Alur et al. (1995); Alur et al. (1992); Henzinger (2000)). In the literature on diagnosis of timed automata, it is shown that the problem of synthesizing diagnosers in both deterministic timed automata and event-recording timed automata is decidable (Bouyer et al. (2005)). In addition, online approaches for diagnosis of timed automata can be proposed by checking the consistency of fault system and faultless system (Lunze et al. (2002); Supavatanakul et al. (2006)). The diagnosis problem of timed automata is explored by Tripakis (2002), Bouyer et al. (2018) and Bouyer et al. (2021), where online diagnosers are proposed. To the best of our knowledge, no general offline approach concerning the construction of an observer/diagnoser for timed automata has been proposed.

In the area of DES, plenty of works address the problem of inferring the evolution of a plant monitored through different observation structures, e.g., assuming only a subset of the event occurrences or, possibly, a function of the state, can be measured (Hadjicostis (2020); Tong et al. (2015)). In this general framework many problems have been proposed and solved, including state estimation and detectability (Giua et al. (2007); Shu et al. (2007)), diagnosis and diagnosability (Sampath et al. (1996); Sampath et al. (1995)). Some of these approaches have been extended to a probabilistic setting (Bertrand et al. (2014); Lefebvre et al. (2020)). In addition, a variety of timed DES model are available (Brave et al. (1988); Ostroff (1990); Brandin et al. (1994)). Concerning fault diagnosis of timed DES, verifiers can be used to check the occurrence of a large variety of timed patterns for DESs (Lefebvre et al. (2022)). To the best of our knowledge, a general approach for the inference of timed DESs is still missing.

This motivates us to explore the issue of state estimation/diagnosis of timed automata under partial observation. We consider a class of timed automata characterized by a single clock that is reset to zero after each event occurrence. A time interval is associated with each transition to specify when it may occur. In a preliminary work (Gao et al. (2020)), we addressed a very restrictive scenario where no observation was received by the plant, and we showed how in this particular case the state estimate could be updated as time elapses. In this work, we extend the approach in Gao et al. (2020) by considering the more general scenario in which some of the event occurrences are observable. In more detail, this work takes into account the information coming from the observation of new events at certain time instants, and aims at determining if a fault behaviour has occurred.

The solution proposed in this paper is based on a purely discrete event description of the behaviour of the timed DES, associating a finite state automaton called a *zone automaton*. Each state of the zone automaton is associated with a state of the timed automaton and a time interval, called a *zone*, which specifies how long the timed automaton may sojourn in that state. When time elapses the state of the zone automaton may change either because of the occurrence

of an event (event-driven evolution) or because a certain amount of time has elapsed with no observation (time-driven evolution). The fault recognizer can be constructed making the parallel composition of the zone automaton and a fault monitor that always marks the fault behaviour after it occurs. The fault diagnosis approach is based on analysing the reachability of the fault recognizer.

The rest of the paper is organized as follows. Section 2 introduces the background of discrete event systems, timed finite automata and time semantics used throughout the paper. Section 3 formally sets the problem of diagnosis of a partially observed timed automaton. Section 4 introduces the notion of *zone automaton* and provides an algorithm for its computation. Section 5 constructs a fault recognizer and investigates the dynamics of a timed DES with faults. Section 6 deals with diagnosis problem of timed DES in function of analysing the reachability of the fault recognizer. Finally, Section 7 concludes the paper.

## 2. PRELIMINARIES

A nondeterministic finite automaton (NFA) is a four-tuple  $G = (X, E, \Delta, X_0)$ , where  $X$  is a finite set of states,  $E$  is the alphabet,  $\Delta \subseteq X \times E \times X$  is a transition relation and  $X_0 \subseteq X$  is a set of initial states. The set of events  $E$  can be partitioned as  $E = E_o \cup E_{uo}$ , where  $E_o$  is the set of observable events, and  $E_{uo}$  is the set of unobservable events. Note that  $E_{uo}$  and  $E_o$  are two disjoint subsets. We denote by  $E^*$  the set of all finite strings on  $E$ , including the empty word  $\varepsilon$ . The *concatenation*  $s_1 \cdot s_2$  of two strings  $s_1 \in E^*$  and  $s_2 \in E^*$  is a string consisting of  $s_1$  immediately followed by  $s_2$ . The empty string  $\varepsilon$  is an identity element of concatenation, i.e., for any string  $s \in E^*$ , it holds that  $\varepsilon \cdot s = s = s \cdot \varepsilon$ . The number of occurrences  $e \in E$  in  $s$  is denoted by  $|s|_e$ .

We denote the sets of non-negative real numbers and natural numbers as  $\mathbb{R}_{\geq 0}$  and  $\mathbb{N}$ , respectively. The set of real numbers lying between a lower bound  $I_l \in \mathbb{N}$  and an upper bound  $I_u \in \mathbb{N} \cup \{+\infty\}$  is said to be a *time interval*. A *closed time interval* is denoted by  $[I_l, I_u]$ . In addition, an open segment  $(I_l, I_u)$  and semi-open segments  $[I_l, I_u)$  and  $(I_l, I_u]$  can also be *time intervals*. We denote the set of all time intervals and the set of all closed time intervals as  $\mathbb{I}$  and  $\mathbb{I}_c$ , respectively, where  $\mathbb{I}_c \subseteq \mathbb{I}$ .

We define the addition operation on two time intervals  $I_1, I_2 \in \mathbb{I}$  as  $I_1 \oplus I_2 = \{t_1 + t_2 \in \mathbb{R}_{\geq 0} \mid t_1 \in I_1, t_2 \in I_2\}$ . That is to say, given  $I_i = [I_{l,i}, I_{u,i}]$ ,  $i = 1, 2$ , we have  $I_1 \oplus I_2 = [I_{l,1} + I_{l,2}, I_{u,1} + I_{u,2}]$ . The addition operation can be extended to  $n$  ( $n > 1$ ) time intervals in a set  $\{I_1, \dots, I_n\}$ , i.e.,  $I_1 \oplus \dots \oplus I_n = ((I_1 \oplus I_2) \oplus \dots) \oplus I_n$ , denoted as  $\bigoplus_{i=1}^n I_i$ . A *timed finite automaton* (TFA) (Gao et al. (2020)) is a five-tuple  $G = (X, E, \Delta, \Gamma, X_0)$ , where  $X$  is a finite set of states,  $E$  is an alphabet,  $\Delta \subseteq X \times E \times X$  is a transition relation,  $\Gamma : \Delta \rightarrow \mathbb{I}_c$  is a timing function and  $X_0 \subseteq X$  is a set of initial states. In simple words, a TFA  $G = (X, E, \Delta, \Gamma, X_0)$  is a NFA  $G = (X, E, \Delta, X_0)$  endowed with a timing structure that associates with each transition in  $\Delta$  a time interval in  $\mathbb{I}_c$ .

We assume that a TFA operates under a single clock, which is reset upon the occurrence of any event in  $E$ . The transition relation and the timing function specify the dynamics of the TFA. In more detail, given two states  $x, x' \in X$  and an event  $e \in E$ ,  $(x, e, x') \in \Delta$  denotes that the occurrence of event  $e$  leads to state  $x'$  when the TFA is in state  $x$ . The timing function  $\Gamma$  maps the transition  $(x, e, x')$  to a time interval, which specifies a range of clock values at which the event  $e$  may occur. We further define  $\Gamma_l : \Delta \rightarrow \mathbb{N}$  (resp.,  $\Gamma_u : \Delta \rightarrow \mathbb{N} \cup \{+\infty\}$ ) as the lower (resp., upper) timing function associating a transition in  $\Delta$  to the left (resp., right) bound of the time interval associated with it. Therefore  $\Gamma((x, e, x')) = [\Gamma_l((x, e, x')), \Gamma_u((x, e, x'))]$ .

A TFA  $G = (X, E, \Delta, \Gamma, X_0)$  can be represented by a graph, where a state  $x \in X$  corresponds to a node, and each initial state in  $X_0$  is marked by an input arrow. For each transition  $(x, e, x') \in \Delta$  with  $\Gamma((x, e, x')) = I$ , there exists a directed edge from  $x$  to  $x'$  labeled with the symbol  $e$  and the time interval  $I$ .

*Example 1.* Consider the TFA  $G = (X, E, \Delta, \Gamma, X_0)$  with  $X = \{x_0, x_1, x_2, x_3\}$ ,  $E = \{a, b, c\}$  and  $X_0 = \{x_0\}$ . Let the transition relation  $\Delta$  and the timing function  $\Gamma$  be defined as in Table 1. The graphical representation of  $G$  is visualized in Fig. 1.  $\square$

Table 1. The timing function of the TFA  $G$  in Fig. 1.

| $\delta \in \Delta$ | $\Gamma(\delta)$ |
|---------------------|------------------|
| $(x_0, c, x_1)$     | $[0, 1]$         |
| $(x_0, c, x_2)$     | $[1, 2]$         |
| $(x_1, b, x_0)$     | $[1, 2]$         |
| $(x_1, d, x_3)$     | $[0, 0]$         |
| $(x_2, a, x_2)$     | $[0, 1]$         |
| $(x_2, b, x_1)$     | $[1, 2]$         |
| $(x_3, b, x_3)$     | $[1, 2]$         |

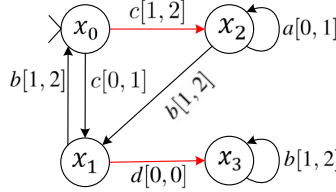


Fig. 1. A TFA  $G$ , where fault transitions are shown in red.

The behaviour of a TFA is described via its timed runs. Given  $G = (X, E, \Delta, \Gamma, X_0)$ , a *timed run*  $\rho$  of length  $k \geq 0$  from 0 to  $t_k \in \mathbb{R}_{\geq 0}$  is a sequence of  $k + 1$  states  $x_{(i)} \in X$  ( $i \in \{0, \dots, k\}$ ), and  $k$  pairs  $(e_i, t_i) \in E \times \mathbb{R}_{\geq 0}$  ( $i \in \{1, \dots, k\}$ ), represented as

$$\rho : x_{(0)} \xrightarrow{(e_1, t_1)} x_{(1)} \cdots \xrightarrow{(e_{k-1}, t_{k-1})} x_{(k-1)} \xrightarrow{(e_k, t_k)} x_{(k)}$$

such that the following two conditions are satisfied for all  $i \in \{1, \dots, k\}$  by letting  $t_0 = 0$ :

$$(x_{(i-1)}, e_i, x_{(i)}) \in \Delta, \quad (1)$$

$$t_i - t_{i-1} \in \Gamma((x_{(i-1)}, e_i, x_{(i)})). \quad (2)$$

We define the *timed word generated by*  $\rho$  as  $\sigma(\rho) = (e_1, t_1)(e_2, t_2) \cdots (e_k, t_k) \in (E \times \mathbb{R}_{\geq 0})^*$ . We also define the *logical word generated by*  $\rho$  as  $S(\sigma(\rho)) = e_1 e_2 \cdots e_k$  via a function defined as  $S : (E \times \mathbb{R}_{\geq 0})^* \rightarrow E^*$ . Given a timed run  $\rho$  of length 0 that only contains the starting state  $x_{(0)}$  and no transition, we denote  $\lambda$  as the *empty timed word* in  $E \times \mathbb{R}_{\geq 0}$ . The timed word and the logical word generated by  $\rho$  are denoted respectively as  $\sigma(\rho) = \lambda$  and  $S(\sigma(\rho)) = S(\lambda) = \varepsilon$ . For the timed word  $\sigma(\rho)$  generated from an arbitrary timed run  $\rho$ , it is  $\lambda \cdot \sigma(\rho) = \sigma(\rho) = \sigma(\rho) \cdot \lambda$ . The *starting state* and the *ending state* of a timed run  $\rho$  are denoted by  $x_{st}(\rho) = x_{(0)}$  and  $x_{en}(\rho) = x_{(k)}$ , respectively. The *starting time* and the *ending time* of  $\rho$  are denoted by  $t_{st}(\rho) = 0$  and  $t_{en}(\rho) = t_k$ , respectively. In addition, the *duration of*  $\rho$  is denoted as  $T(\rho) = t_k$ . Note that Eq. (2) clearly implies  $T(\rho) \in \bigoplus_{i=0}^{k-1} \Gamma(x_{(i)}, e_{i+1}, x_{(i+1)})$ . The set of timed runs generated by  $G$  is denoted as  $\mathcal{R}(G)$ .

*Example 2.* Given the TFA in Fig. 1, a timed run of length 3 from time 0 to 2 is  $\rho : x_0 \xrightarrow{(c,1)} x_2 \xrightarrow{(b,2)} x_1 \xrightarrow{(d,2)} x_3$ . The timed word  $\sigma(\rho) = (c, 1)(b, 2)(d, 2)$  corresponds to events  $c$ ,  $b$ , and  $d$  occurring at time instants  $t_1 = 1$ ,  $t_2 = 2$ , and  $t_3 = 2$ , respectively. It starts from  $x_{st}(\rho) = x_0$  at the starting time  $t_{st}(\rho) = 0$  and terminates in  $x_{en}(\rho) = x_3$  at the ending time  $t_{en}(\rho) = 2$ . The logical word generated by  $\rho$  is  $S(\sigma(\rho)) = cbd$ . It involves three transitions, namely  $(x_0, c, x_2)$ ,  $(x_2, b, x_1)$ , and  $(x_1, d, x_3)$ . In addition, we have  $t_1 \in \Gamma(x_0, c, x_2)$ ,  $t_2 - t_1 \in \Gamma(x_2, b, x_1)$  and  $t_3 - t_2 \in \Gamma(x_1, d, x_3)$ .  $\square$

Different types of semantics pose additional constraints on how long a TFA dwells at each state while generating a timed run. In this paper we consider a type of time semantics that specifies the *maximal dwell time* at a state. Given a TFA  $G = (X, E, \Delta, \Gamma, X_0)$ , the maximal dwell time at state  $x \in X$  is defined as  $d_{max}(x) = \max\{\Gamma_u((x, e, x')) \mid (x, e, x') \in \Delta\}$

if there exist  $x' \in X$  and  $e \in E$  such that  $(x, e, x') \in \Delta$ ; otherwise  $d_{max}(x) = \infty$ . The TFA cannot stay in  $x \in X$  if the clock takes a value larger than the maximal dwell time at  $x$ , i.e.,  $d_{max}(x)$ . If there exists no enabled transition at  $x \in X$ , then  $d_{max}(x) = \infty$ , implying that  $G$  can stay at  $x$  indefinitely. Meanwhile, if there exists one or more enabled transitions at  $x$ , the maximal dwell time at  $x$  is equal to the maximum upper bound of the intervals of such transitions. It implies that all such transitions are candidates to occur. However, a transition has to be fired once the clock at  $x$  reaches the maximal dwell time at  $x$ .

Given a TFA  $G = (X, E, \Delta, \Gamma, X_0)$ , a timed run  $\rho$  of length  $k \geq 0$  and a time instant  $t \in \mathbb{R}_{\geq 0}$ , a *timed evolution of  $G$  from 0 to  $t$*  is defined by a pair  $(\sigma(\rho), t) \in (E \times \mathbb{R}_{\geq 0})^* \times \mathbb{R}_{\geq 0}$ , where  $0 \leq t - t_{en}(\rho) \leq d_{max}(x_{en}(\rho))$ . Furthermore, we denote as

$$\mathcal{E}(G, t) = \{(\sigma(\rho), t) \mid (\exists \rho \in \mathcal{R}(G)) x_{st}(\rho) \in X_0, \\ 0 \leq t - t_{en}(\rho) \leq d_{max}(x_{en}(\rho))\}$$

the *timed language of  $G$  from 0 to  $t$* .

In other words, a timed evolution of  $G$  from 0 to  $t$  is defined as a pair whose first entry is a timed word  $\sigma(\rho)$ , where  $\rho$  starts at 0 from an initial state in  $X_0$ , and whose second entry is the time instant  $t$ , where the time semantics constrains the time that the system may stay in the ending state  $x_{en}(\rho)$ , namely  $t - t_{en}(\rho)$ , to be less than or equal to the maximal dwell time of  $x_{en}(\rho)$ . The timed language of  $G$  from 0 to  $t$  contains all possible timed evolutions of  $G$  from 0 to  $t$ .

### 3. PROBLEM STATEMENT

In this work we model a partially observed timed DES as a TFA  $G = (X, E, \Delta, \Gamma, X_0)$  with a partition of the alphabet  $E$  into a set of observable events  $E_o$  and a set of unobservable events  $E_{uo}$ , namely  $E = E_o \cup E_{uo}$ . We assume that the timed system may be affected by a set of faults described by timed transitions whose occurrence changes the state of the plant and resets the clock. Two types of fault transitions are considered in this paper: observable fault transitions labeled with a symbol in  $E_o$ ; unobservable fault transitions labeled with a symbol in  $E_{uo}$ . The set of transitions modeling a regular behaviour is denoted as  $\Delta_{reg}$ , while the set of transitions modeling a fault behaviour is denoted as  $\Delta_{fault}$ . Clearly, it is  $\Delta = \Delta_{reg} \cup \Delta_{fault}$ .

Next we preliminarily define a *projection function* on timed words.

*Definition 1.* Given a TFA  $G$  with  $E = E_o \cup E_{uo}$ , a *projection function*  $P : (E \times \mathbb{R}_{\geq 0})^* \rightarrow (E_o \times \mathbb{R}_{\geq 0})^*$  is defined as  $P(\lambda) = \lambda$ , and

$$P(\sigma(\rho) \cdot (e, t)) = \begin{cases} P(\sigma(\rho)) & \text{if } e \in E_{uo} \\ P(\sigma(\rho)) \cdot (e, t) & \text{if } e \in E_o \end{cases}$$

for the timed word  $\sigma(\rho) \in (E \times \mathbb{R}_{\geq 0})^*$  generated from any timed run  $\rho \in \mathcal{R}(G)$  and for all  $(e, t) \in E \times \mathbb{R}_{\geq 0}$ .  $\square$

In other words, the projection operator  $P$  simply erases the pairs in a timed word, whose first entry is an unobservable event in  $E_{uo}$  and whose second entry is the time of its occurrence. Given a TFA  $G$  and a timed run  $\rho \in \mathcal{R}(G)$ , the projection function  $P$  always maps the timed word  $\sigma(\rho)$  to an observed word  $\sigma_o \in (E_o \times \mathbb{R}_{\geq 0})^*$ . The pair  $(\sigma_o, t) = (P(\sigma(\rho)), t)$  is the *timed observation* related to  $(\sigma(\rho), t)$ .

We define a *diagnosis function* for a set of fault transitions  $\Delta_{fault}$  as  $\phi : (E_o \times \mathbb{R}_{\geq 0})^* \times \mathbb{R}_{\geq 0} \rightarrow \{F, N, U\}$  associated to each timed observation  $(\sigma_o, t)$  a diagnosis state  $\phi((\sigma_o, t))$ , where  $\phi((\sigma_o, t)) = F$  (resp.,  $\phi((\sigma_o, t)) = N$ ) denotes that a fault transition in  $\Delta_{fault}$  has (resp., not) been executed while producing  $(\sigma_o, t)$ , and  $\phi((\sigma_o, t)) = U$  denotes that a fault transition may or may not have been executed. This paper aims at diagnosing a fault behaviour based on a timed observation  $(\sigma_o, t)$ , namely computing  $\phi((\sigma_o, t))$ . Note that this implies that we are not distinguishing among different fault transitions. According to the notation used in the most of the literature on fault diagnosis of discrete event systems (Sampath et al. (1996); Sampath et al. (1995)), this means that we assume that all faults belong to the same class.

#### 4. ZONE AUTOMATON

In this section, we introduce the notion of *zone automaton* that is a finite state automaton providing a purely discrete event description of the behaviour of a TFA of interest. We first propose several preliminary definitions as follows.

*Definition 2.* Given a TFA  $G$ , an *extended state* is defined as a pair  $(x, \theta)$ , where  $x$  is a state of  $G$  and  $\theta \in [0, d_{max}(x)]$  is the current value of the clock.  $\square$

In other words<sup>6</sup>, an extended state  $(x, \theta)$  keeps track of the current clock assignment  $\theta$  while  $G$  dwells at state  $x$ .

*Definition 3.* Given a TFA  $G = (X, E, \Delta, \Gamma, X_0)$ , the *set of active transitions at an extended state*  $(x, \theta) \in X \times \mathbb{R}_{\geq 0}$  is defined as  $\mathcal{A}(x, \theta) = \{(x, e, x') \in \Delta \mid (\exists e \in E)(\exists x' \in X) \theta \in \Gamma((x, e, x'))\}$ .  $\square$

In simple words, the set of active transitions at an extended state  $(x, \theta)$  includes all the transitions that may fire from  $x$  with a clock value  $\theta$ . The set of active transitions at  $(x, \theta)$  may vary for different values of  $\theta$  in  $[0, d_{max}(x)]$ . This leads to the definition of clock *zones* associated with a given state  $x \in X$ .

*Definition 4.* Given a TFA  $G = (X, E, \Delta, \Gamma, X_0)$ , the *set of zones* of  $x \in X$  is defined as  $Z(x) = \{[0, +\infty)\}$  if  $d_{max}(x) = \infty$ ; otherwise it is defined as a set of time intervals  $Z(x) = \{z_0, \dots, z_n\} \subseteq \mathbb{I}$ ,  $n \geq 0$ , where the following conditions hold:

- $z_0 = [0, 0]$ ;
- $\bigcup_{i=0}^n z_i = [0, d_{max}(x)]$ ;
- $\theta < \theta'$  holds for all  $\theta \in z_{i-1}$ ,  $\theta' \in z_i$ ,  $i \in \{1, \dots, n\}$ ;
- $\mathcal{A}(x, \theta) = \mathcal{A}(x, \theta')$  holds for all  $\theta, \theta' \in z_i$ ,  $i \in \{0, \dots, n\}$ ;
- $\mathcal{A}(x, \theta) \neq \mathcal{A}(x, \theta')$  holds for all  $\theta \in z_{i-1}$ ,  $\theta' \in z_i$ ,  $i \in \{2, \dots, n\}$ .

In addition,  $prec(z_i) = z_{i-1}$  (resp.,  $succ(z_i) = z_{i+1}$ ) is said to be the *preceding zone* (resp., *succeeding zone*) of  $z_i \in Z(x)$ , where  $i \in \{1, \dots, n\}$  (resp.,  $i \in \{0, \dots, n-1\}$ ).  $\square$

If there exists no transition originating from  $x$ ,  $G$  stays at  $x$  indefinitely: in such a case the set of zones of  $x$  is a singleton  $\{[0, +\infty)\}$ . Otherwise, the set of zones of a state  $x$  follows from the partitioning of the dwell time at  $x$  into several time intervals to which the clock may belong. The union of all zones in  $Z(x)$  covers the interval  $[0, d_{max}(x)]$ . Any two zones of  $x$  are disjoint. If  $\theta, \theta' \in z_i$ , where  $i \in \{1, \dots, n\}$ , the sets of active transitions at two extended states  $(x, \theta)$  and  $(x, \theta')$  are identical. In addition, the firability of transitions differs between  $(x, \theta)$  and  $(x, \theta')$  if  $\theta \in z_{i-1}$ ,  $\theta' \in z_i$ ,  $i \in \{2, \dots, n\}$ . Particularly,  $z_0 = [0, 0]$  is defined to be a zone associated with each state of  $G$ , apart from the case of  $d_{max}(x) = +\infty$ . This originates from the considered time semantics, according to whom the clock is reset whenever  $G$  arrives at a state in  $X$ . Then the clock evolves discretely from a time instant  $\theta \in z_{i-1}$  to another time instant  $\theta' \in z_i$ ,  $i \in \{1, \dots, n\}$ .

Given a state  $x$  and two zones  $z, succ(z) \in Z(x)$ , a new event  $\tau$  denotes that in a state  $x$  the clock value may evolve from any  $\theta \in z$  to any  $\theta' \in succ(z)$  as time elapses. We now formalize the definition of zone automaton.

*Definition 5.* Given a TFA  $G = (X, E, \Delta, \Gamma, X_0)$ , the *zone automaton* of  $G$  is an NFA  $ZA(G) = (V, E_\tau, \Delta_z, V_0)$ , where

- $V \subseteq X \times \bigcup_{x \in X} Z(x)$  is the finite set of states,
- $E_\tau \subseteq E \cup \{\tau\}$  is the alphabet,
- $\Delta_z \subseteq V \times E_\tau \times V$  is the transition relation, where the transitions in  $\Delta_z$  are defined by the following rules:
  - $((x, z), \tau, (x, succ(z))) \in \Delta_z$  if  $z, succ(z) \in Z(x)$ ;
  - $((x, z), e, (x', z_0)) \in \Delta_z$  if  $z \in Z(x)$ ,  $z_0 \in Z(x')$ ,  $(x, e, x') \in \mathcal{A}(x, \theta)$  for all  $\theta \in z$ ,
- $V_0 = \{(x, z_0) \mid x \in X_0\} \subseteq V$  is the set of initial states.  $\square$

<sup>6</sup> According to the usual terminology in hybrid systems community, the extended state  $(x, \theta)$  is the hybrid state of the timed automaton, while  $x$  and  $\theta$  are the discrete and the continuous states of the timed automaton, respectively.

We use the zone automaton to describe the time-driven and event-driven evolution of a TFA  $G = (X, E, \Delta, \Gamma, X_0)$ . Each state in a zone automaton is a pair  $(x, z)$  with  $x \in X$  and  $z \in Z(x)$ . The alphabet is composed of the events in  $E$  and event  $\tau$ . The transition relation specifies the dynamics of the automaton: starting from a state  $(x, z)$ , a transition  $((x, z), \tau, (x, succ(z))) \in \Delta_z$  corresponds to a time-driven evolution of  $G$  from a clock value in  $z$  to another clock value in  $succ(z)$  while  $G$  is at  $x$ ; a transition  $((x, z), e, (x', z_0)) \in \Delta_z$  goes from state  $(x, z)$  to state  $(x', z_0)$ , indicating that the occurrence of event  $e$  yields state  $x'$  when the current state of the system is  $x$  and the current clock is in  $z$ . The set of initial states is the set of pairs of a state  $x \in X_0$  and  $z_0 \in Z(x)$ .

Given a TFA  $G = (X, E, \Delta, \Gamma, X_0)$ , the zone automaton  $ZA(G) = (V, E_\tau, \Delta_z, V_0)$  can be constructed by Algorithm 1. A temporary set of states  $V_{new}$  is introduced, containing all states that still need to be explored in order to compute their output transitions. A *while* loop is repeated until  $V_{new} = \emptyset$ . A transition  $((x, z), \tau, (x, succ(z)))$  is set in  $\Delta_z$  if  $succ(z)$  is a zone at  $x$ . For each transition  $(x, e, x') \in \Delta$  satisfying  $z \subseteq \Gamma((x, e, x'))$ , a transition labeled with  $e$  is set from  $v = (x, z)$ . Note that if the maximal dwell time of  $x'$  is  $+\infty$  (resp., if it is not), the transition labeled with  $e$  would lead to state  $(x', [0, +\infty))$  (resp., state  $(x', z_0)$ ). To avoid redundant repetitions of the *while* loop, the state  $v'$  is included in  $V_{new}$  if  $v'$  is neither in  $V$  nor in  $V_{new}$ . The *while* loop stops once all states in  $V_{new}$  have been explored. A numerical example to illustrate the zone automaton will be given in Section 5.

---

**Algorithm 1:** Construction of a zone automaton of a TFA

---

**Input:** A TFA  $G = (X, E, \Delta, \Gamma, X_0)$  with  $E = E_o \cup E_{uo}$

**Output:** A zone automaton  $ZA(G) = (V, E_\tau, \Delta_z, V_0)$

```

1 let  $V = \emptyset$ ,  $E_\tau = E \cup \{\tau\}$ ,  $\Delta_z = \emptyset$ ,  $V_0 = \{(x, z_0) \mid x \in X_0\}$ , and  $V_{new} = V_0$ 
2 while  $V_{new} \neq \emptyset$  do
3   select a  $v = (x, z) \in V_{new}$ 
4   if  $succ(z) \in Z(x)$  then
5     let  $\bar{v} = (x, succ(z))$ ,  $\Delta_z = \Delta_z \cup \{(v, \tau, \bar{v})\}$ , and  $V_{new} = V_{new} \cup \{\bar{v}\}$ 
6   for each  $(x, e, x') \in \Delta$  do
7     if  $z \subseteq \Gamma((x, e, x'))$  then
8       if  $d_{max}(x') \neq +\infty$  then
9         let  $v' = (x', z_0)$ 
10      else
11        let  $v' = (x', [0, +\infty))$ 
12      let  $\Delta_z = \Delta_z \cup \{(v, e, v')\}$ 
13      if  $v' \notin V \cup V_{new}$  then
14        let  $V_{new} = V_{new} \cup \{v'\}$ 
15 let  $V = V \cup \{v\}$  and  $V_{new} = V_{new} \setminus \{v\}$ 
16 return  $ZA(G) = (V, E_\tau, \Delta_z, V_0)$ 

```

---

## 5. FAULT RECOGNIZER

In this section, we construct a *fault recognizer* that recognizes the occurrence of faults. We first transform the model  $G$  of the plant with faults into a *canonical plant*  $G_f$  with faults. For the canonical plant  $G_f$ , the zone automaton  $ZA(G_f)$  is constructed. The particular structure of the canonical plant allows us to construct a *fault recognizer* by synchronizing  $ZA(G_f)$  with a *fault monitor* that recognizes the occurrence of a fault denoted by a symbol  $f$ .

*Definition 6.* Consider a partially observed TFA  $G = (X, E, \Delta, \Gamma, X_0)$  with  $E = E_o \cup E_{uo}$ . The canonical plant is modeled as a TFA  $G_f = (X \cup X_f, E \cup \{f\}, \Delta_f, \Gamma_f, X_0)$ , where  $f$  is an additional unobservable event used to model the

occurrence of a fault transition. The set of additional states  $X_f$ , the transition relation  $\Delta_f$ , and the timing function  $\Gamma_f$  are defined according to each  $\delta = (x, e, x'') \in \Delta$  as follows:

- if  $\delta \in \Delta_{fault}$  and  $e \in E_{uo}$ , we define  $\delta_f = (x, f, x'') \in \Delta_f$  and  $\Gamma_f(\delta_f) = \Gamma(\delta)$ ;
- if  $\delta \in \Delta_{fault}$  and  $e \in E_o$ , we define  $\{\delta_1, \delta_2\} \subseteq \Delta_f$ ,  $\Gamma_f(\delta_1) = \Gamma(\delta)$ , and  $\Gamma_f(\delta_2) = [0, 0]$ , where  $\delta_1 = (x, f, x')$ ,  $\delta_2 = (x', e, x'')$ , and  $x' \in X_f$ ;
- if  $\delta \in \Delta_{reg}$ , we define  $\delta \in \Delta_f$  and  $\Gamma_f(\delta) = \Gamma(\delta)$ . □

In the canonical plant  $G_f$ , the new fault event  $f$  is introduced: this will allow construct the fault recognizer by synchronization with the fault monitor. Given a transition  $\delta = (x, e, x'') \in \Delta$ ,  $G$  can generate a timed run from initial time 0 ending with  $x \xrightarrow{(e,t)} x''$ , where  $t \in \Gamma(\delta)$ . If  $e$  is associated with an observable fault transition,  $G_f$  can generate a timed run ending with  $x \xrightarrow{(f,t)} x' \xrightarrow{(e,t)} x''$ , implying that  $\delta$  is replaced by  $\delta_1 = (x, f, x')$  satisfying  $\Gamma_f(\delta_1) = \Gamma(\delta)$  and following by  $\delta_2 = (x', e, x'')$  that occurs immediately, i.e.,  $\Gamma_f(\delta_2) = [0, 0]$ . In other words, considering an observable fault transition,  $G_f$  keeps track of both the occurrence of the fault  $f$  and the observation of  $e$ .

On the contrary, it is not necessary to keep track of the occurrence of the unobservable event. If  $e$  is associated with an unobservable fault transition, a new unobservable symbol  $f$  is labeled with the transition  $\delta = (x, f, x'') \in \Delta_f$  of  $G_f$ . If  $\delta \in \Delta_{reg}$ , we let  $\delta \in \Delta_f$  and  $\Gamma_f(\delta) = \Gamma(\delta)$ . Note that in  $G_f$ , the set of unobservable events is extended to  $E_{uo} \cup \{f\}$ .

*Example 3.* Consider the TFA  $G$  in Fig. 1 with  $E_o = \{a, b, c\}$  and  $\Delta_{fault} = \{(x_0, c, x_2), (x_1, d, x_3)\}$ . A canonical plant  $G_f = (X \cup \{x_f\}, E \cup \{f\}, \Delta_f, \Gamma_f, X_0)$  is depicted in Fig. 2, where fault transitions are shown in red. The transition  $(x_1, d, x_3) \in \Delta_{fault}$  with an unobservable fault  $d$  is replaced by  $(x_1, f, x_3) \in \Delta_f$  in  $G_f$ , and  $(x_0, c, x_2) \in \Delta_{fault}$  with an observable fault  $c$  is replaced by two consecutive transitions in  $G_f$ , namely  $(x_0, f, x_f)$  and  $(x_f, c, x_2)$  satisfying that  $\Gamma((x_0, f, x_f)) = [1, 2]$  and  $\Gamma((x_f, c, x_2)) = [0, 0]$ .

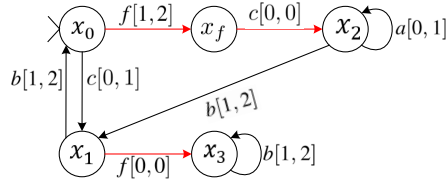


Fig. 2. The canonical plant  $G_f$  associated with the TFA  $G$  in Fig. 1 as described in Example 3.

Next we consider the zones of each discrete state of  $G_f$ . For  $x_0$ , from which there exist two transitions originating, namely  $(x_0, c, x_2) \in \Delta$  with  $\Gamma((x_0, c, x_2)) = [1, 2]$  and  $(x_0, c, x_1) \in \Delta$  with  $\Gamma((x_0, c, x_1)) = [1, 2]$ , the maximal dwell time at  $x_0$  is  $d_{max}(x_0) = 2$ . Consequently, the set of zones of  $x_0$  is  $Z(x_0) = \{[0, 0], (0, 1), [1, 1], (1, 2)\}$ . The set of active transitions at  $(x_0, \theta)$ , where  $\theta$  is a time instant in  $z \in Z(x_0)$ , are reported in Table 2. As for the set of zones of other states of  $G_f$ , we have  $Z(x_1) = \{[0, 0], (0, 1), [1, 2]\}$ ,  $Z(x_2) = \{[0, 0], (0, 1), [1, 1], (1, 2)\}$ ,  $Z(x_3) = \{[0, 0], (0, 1), [1, 2]\}$  and  $Z(x_f) = \{[0, 0]\}$ .

Table 2. Sets of active transitions at  $(x_0, \theta)$  for the TFA  $G$  in Fig. 1, where  $\theta \in z_i$ ,  $i \in \{0, 1, 2, 3\}$ .

| $i$ | $z_i$    | $\mathcal{A}(x_0, \theta), \theta \in z_i$ |
|-----|----------|--|
| 0   | $[0, 0]$ | $\{(x_0, c, x_1)\}$                        |
| 1   | $(0, 1)$ | $\{(x_0, c, x_1)\}$                        |
| 2   | $[1, 1]$ | $\{(x_0, c, x_1), (x_0, b, x_2)\}$         |
| 3   | $(1, 2)$ | $\{(x_0, b, x_2)\}$                        |

The zone automaton  $ZA(G_f) = (V, E_\tau, \Delta_z, V_0)$  is shown in Fig. 3. The initial state is  $(x_0, [0, 0])$ , implying that  $G$  starts from  $x_0$  at clock value 0. A transition labeled with an event  $\tau$  implies a time-driven evolution of  $G$ . For instance, a transition  $((x_0, [0, 0]), \tau, (x_0, (0, 1)))$  represents that the clock may evolve from the value in  $[0, 0]$  to any value in  $(0, 1)$



if  $G$  is at  $x_0$ . Meanwhile, a transition labeled with an event in  $E \cup \{f\}$  implies an event-driven evolution of  $G$ . For instance, a transition labeled with  $c$  goes from  $(x_0, [1, 1])$  to  $(x_2, [0, 0])$ . It represents a state evolution from  $x_0$  to  $x_2$  under the occurrence of an event  $c$ , upon which the clock is reset.  $\square$

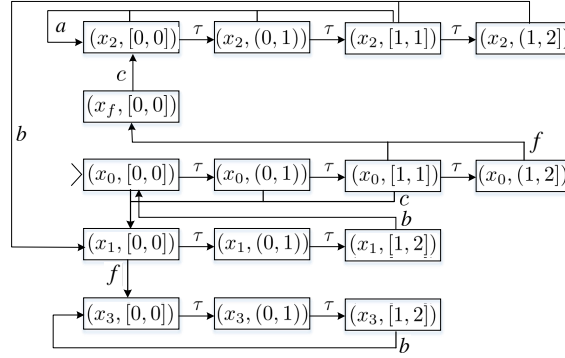


Fig. 3. Zone automaton  $ZA(G_f)$  of  $G_f$  in Fig. 2.

We now introduce a deterministic untimed automaton called *fault monitor* and denote it as  $M = (\{N, F\}, \{f\}, \{(N, f, F), (F, f, F)\}, N)$  shown in Fig. 4, where state  $N$  (resp.,  $F$ ) denotes that no fault (resp., a fault) has occurred, and the state always evolves from  $N$  and  $F$  to  $F$  upon each occurrence of  $f$ . To deal with fault diagnosis, we construct a *fault recognizer*  $Rec(G_f)$  by composing  $ZA(G_f)$  with  $M$  by parallel composition such that labels  $N$  and  $F$  are attached to states of  $Rec(G_f)$  to recognize the occurrence of faults.

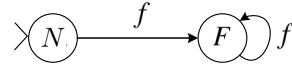


Fig. 4. Fault monitor  $M$  for diagnosing event  $f$ .

*Definition 7.* Consider a timed DES with faults modeled by a TFA  $G_f = (X \cup X_f, E \cup \{f\}, \Delta_f, \Gamma_f, X_0)$ . Given zone automaton  $ZA(G_f) = (V, E \cup \{f, \tau\}, \Delta_z, v_0)$  and a fault monitor  $M = (\{N, F\}, \{f\}, \{(N, f, F), (F, f, F)\}, N)$ , the *fault recognizer* is the automaton  $Rec(G_f) = (X_{rec}, E_{rec}, \Delta_{rec}, X_{rec0})$ , where  $X_{rec} \subseteq V \times \{N, F\}$ ,  $E_{rec} = E \cup \{f, \tau\}$ ,  $X_{rec0} = V_0 \times \{N\}$ , and a transition  $\delta_{rec} \in \Delta_{rec}$  satisfies the following conditions:

- if  $e = f$ , then  $\{(v, N), f, (v', F)\}, \{(v, F), f, (v', F)\} \subseteq \Delta_{rec}$  holds for each  $(v, f, v') \in \Delta_z$ ;
- if  $e \in E \cup \{\tau\}$ , then  $((v, N), e, (v', N)) \in \Delta_{rec}$  holds for each  $(v, e, v') \in \Delta_z$ .  $\square$

For instance, the fault recognizer  $Rec(G_f)$  is depicted in Fig. 5, where  $G_f$  is shown in Fig. 2.

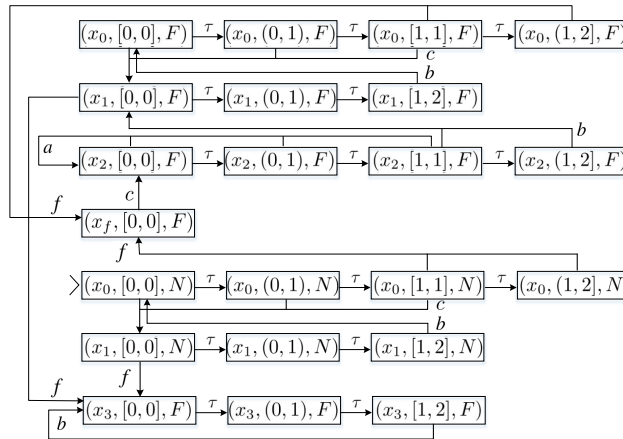


Fig. 5. Fault recognizer  $Rec(G_f)$  of  $G_f$  in Fig. 2.

## 6. FAULT DIAGNOSIS OF TIMED DES

In this section, we deal with fault diagnosis of timed DES with faults modeled by a TFA  $G_f = (X \cup X_f, E \cup \{f\}, \Delta_f, \Gamma_f, X_0)$ . We first study the dynamics of its fault recognizer  $Rec(G_f) = (X_{rec}, E_{rec}, \Delta_{rec}, X_{rec0})$  via the following definitions.

*Definition 8.* A  $\tau$ -run at  $x \in X$  is defined as a sequence of  $k+1$  states in  $(x, z_i, s_{fault}) \in X_{rec}$  ( $0 \leq i \leq k$ ) and the event  $\tau$ , represented as  $\rho_\tau(x) : (x, z_0, s_{fault}) \xrightarrow{\tau} \cdots \xrightarrow{\tau} (x, z_k, s_{fault})$ , such that  $((x, z_{i-1}, s_{fault}), \tau, (x, z_i, s_{fault})) \in \Delta_{rec}$  holds for  $i \in \{1, \dots, k\}$ . We denote the *starting state* (resp., the *ending state*) of  $\rho_\tau(x)$  as  $q_{st}(\rho_\tau(x)) = (x, z_0, s_{fault})$  (resp.,  $q_{en}(\rho_\tau(x)) = (x, z_k, s_{fault})$ ). The *duration range* of  $\rho_\tau(x)$  is denoted as  $d(\rho_\tau(x)) = z_k$ . The *fault label* of  $\rho_\tau(x)$  is denoted as  $f_{label}(\rho_\tau(x)) = s_{fault}$ .  $\square$

In other words, a  $\tau$ -run at  $x$  essentially represents the time elapsing discretely while  $G_f$  is at  $x \in X$ . The zone  $z_i \in Z(x)$  ( $0 \leq i \leq k$ ) (resp.,  $s_{fault} \in \{N, F\}$ ) provides the range of the possible clock values (resp., the fault label) associated with  $x$ .

*Definition 9.* A *run in*  $Rec(G_f) = (X_{rec}, E_{rec}, \Delta_{rec}, X_{rec0})$  of length  $k$  is defined as a sequence of  $\tau$ -runs  $\rho_\tau(x_{(i)})$  ( $i \in \{0, \dots, k\}$ ) at  $x_{(i)} \in X$ , and  $k$  events  $e_i \in E$  ( $i \in \{1, \dots, k\}$ ), represented as

$$\bar{\rho} : \rho_\tau(x_{(0)}) \xrightarrow{e_1} \rho_\tau(x_{(1)}) \cdots \xrightarrow{e_k} \rho_\tau(x_{(k)}),$$

such that  $(q_{en}(\rho_\tau(x_{(i-1)})), e_i, q_{st}(\rho_\tau(x_{(i)}))) \in \Delta_{rec}$  holds for  $i \in \{1, \dots, k\}$ . In addition, it is  $f_{label}(\rho_\tau(x_{(j)})) = F$  if  $e_i = f$  for  $i \leq j \leq k$ .

We denote the *starting state* (resp., the *ending state*) of  $\bar{\rho}$  as  $q_{st}(\bar{\rho}) = q_{st}(\rho_\tau(x_{(0)}))$  (resp.,  $q_{en}(\bar{\rho}) = q_{en}(\rho_\tau(x_{(k)}))$ ). The *fault label* of  $\bar{\rho}$  is denoted as  $f_{label}(\bar{\rho}) = f_{label}(\rho_\tau(x_{(k)}))$ . The *duration range* of  $\bar{\rho}$  is denoted as  $d(\bar{\rho}) = \bigoplus_{i=0}^k d(\rho_\tau(x_{(i)}))$ . The logical word generated by  $\bar{\rho}$  is denoted as  $s(\bar{\rho}) = e_1 \cdots e_k$  via a function defined as  $s : E_\tau^* \rightarrow E^*$ . The set of runs generated by  $G_z$  is defined as  $\mathcal{R}(Rec(G_f))$ .  $\square$

The  $\tau$ -runs involving an elapsed time  $\tau$  with no executed events essentially represent the time elapsing discretely. A run in  $Rec(G_f)$  represents the evolutions of  $G_f$  that involve both time elapsing and events occurrence. After an event  $e_i, i \in \{1, \dots, k\}$  is executed, the state of  $G_f$  evolves from  $x_{(i-1)}$  to  $x_{(i)}$ . The fault label is  $\bar{\rho} = F$  once a fault has occurred. The logical word of  $\bar{\rho}$  is the sequence of events in  $E \cup \{f\}$  that have been involved in  $\bar{\rho}$ . The duration range of  $\bar{\rho}$  is evaluated by summing up the duration of each  $\tau$ -run at  $x_{(i)}, i \in \{0, \dots, k\}$ .

Next we focus on the fault diagnosis problem when a timed observation is received as a pair of a non-empty timed word and a time instant. We propose and prove the following theorem as follows.

*Theorem 1.* Consider a TFA  $G = (X, E, \Delta, \Gamma, X_0)$  with a set of fault transitions, its canonical plant  $G_f = (X \cup X_f, E \cup \{f\}, \Delta_f, \Gamma_f, X_0)$ , and its fault recognizer  $Rec(G_f) = (X_{rec}, E_{rec}, \Delta_{rec}, X_{rec0})$ . Given a timed observation  $(\sigma_o, t) \in (E_o \times \mathbb{R}_{\geq 0})^* \times \mathbb{R}_{\geq 0}$ , where  $\sigma_o = (e_{o1}, t_1) \cdots (e_{on}, t_n), n \geq 1$ , and  $0 = t_0 \leq t_1 \leq \cdots \leq t_n \leq t$ , then there exists a timed run  $\bar{\rho} \in \mathcal{R}(Rec(G_f))$ , defined as  $\bar{\rho} : \bar{\rho}_{(0)} \xrightarrow{e_{o1}} \bar{\rho}_{(1)} \cdots \xrightarrow{e_{on}} \bar{\rho}_{(n)}$ , such that the following conditions are satisfied:

- (a)  $t \in d(\bar{\rho}), t - t_n \in d(\bar{\rho}_{(n)})$  and  $t_i - t_{i-1} \in d(\bar{\rho}_{(i-1)})$  for  $i \in \{1, \dots, n\}$ ;
- (b)  $P_l(s(\bar{\rho})) = e_{o1} \cdots e_{on}, P_l(s(\bar{\rho}_{(i)})) = \varepsilon$  for  $i \in \{1, \dots, n\}$ , where  $P_l : (E \cup \{f\})^* \rightarrow E_o^*$ ;
- (c)  $f_{label}(\bar{\rho}) = N$  if  $|s(\bar{\rho})|_f = 0$ ; else,  $f_{label}(\bar{\rho}) = F$ .  $\square$

**Proof.** Given a timed observation  $(\sigma_o, t) \in (E_o \times \mathbb{R}_{\geq 0})^* \times \mathbb{R}_{\geq 0}$ , there exists a timed run of  $G$  defined as  $\rho : \rho_0 \xrightarrow{(e_{o1}, t_1)} \cdots \xrightarrow{(e_{on}, t_n)} \rho_n$ , such that  $S(\sigma(\rho_i)) = \varepsilon$  for  $i \in \{0, \dots, n\}$ ,  $(x_{en}(\rho_{i-1}), e_i, x_{st}(\rho_i)) \in \Delta, T(\rho_{i-1}) = t_i - t_{i-1}$  for  $i \in \{1, \dots, n\}$ , and  $T(\rho_n) = t - t_n$ .

If a fault transition labeled with an observable event has been executed, there exists an associated timed run of  $G_f$  defined as  $\rho_f : \rho_0 \xrightarrow{(e_{o1}, t_1)} \cdots \xrightarrow{(e_{oi-1}, t_{i-1})} \rho_{i-1} \xrightarrow{(f, t_i)} x_f \xrightarrow{(e_{oi}, t_i)} \cdots \xrightarrow{(e_{on}, t_n)} \rho_n$ . Based on that, there exists a run of  $Rec(G_f)$

defined as  $\bar{\rho} : \bar{\rho}_0 \xrightarrow{e_{o1}} \dots \xrightarrow{e_{oi-1}} \bar{\rho}_{i-1} \xrightarrow{f} (x_f, [0, 0], F) \xrightarrow{e_{oi}} \bar{\rho}_i \dots \xrightarrow{e_{on}} \bar{\rho}_n$  such that  $(q_{en}(\bar{\rho}_{p-1}), e_{op}, q_{st}(\bar{\rho}_p)) \in \Delta_{rec}$  and  $t_p - t_{p-1} \in d(\bar{\rho}_{p-1})$  hold for  $1 \leq p \leq n$ . Conditions (a), (b), and (c) can be inferred accordingly.

If a fault transition labeled with an unobservable event has been executed, we have a timed run of  $G_f$  defined as  $\rho_f : \rho_0 \xrightarrow{(e_{o1}, t_1)} \dots \xrightarrow{(e_{oi}, t_i)} \rho_i \dots \xrightarrow{(e_{on}, t_n)} \rho_n$ , where  $\rho_i : x_{(i0)} \xrightarrow{(e_{i1}, t_{i1})} \dots \xrightarrow{(f, t_{ij})} x_{(ij)} \xrightarrow{(e_{ij+1}, t_{ij+1})} x_{(ij+1)} \dots \xrightarrow{(e_{im}, t_{im})} x_{(im)}$  ( $m \geq 1, 0 \leq i \leq n$ ) and  $e_{ij} = f$  ( $1 \leq j \leq m$ ). Accordingly, there exists an associated run of  $Rec(G_f)$  defined as  $\bar{\rho} : \bar{\rho}_0 \xrightarrow{e_{o1}} \dots \xrightarrow{e_{oi}} \bar{\rho}_i \dots \xrightarrow{e_{on}} \bar{\rho}_n$ , where  $\bar{\rho}_i : \rho_\tau(x_{(i0)}) \xrightarrow{e_{i1}} \dots \xrightarrow{f} \rho_\tau(x_{(ij)}) \dots \xrightarrow{e_{im}} \rho_\tau(x_{(im)})$ . Thus, conditions (a), (b), and (c) can be inferred.  $\blacksquare$

In other words, taking into account the information coming from the observation of new events at certain time instants, the occurrence of faults can be analysed by exploring all the runs in  $Rec(G_f)$  consistent with the given observation. The fault label  $f_{label}(\bar{\rho})$  associated with  $\bar{\rho}$  denoted whether the run contains a fault ( $f_{label}(\bar{\rho}) = F$ ) or not ( $f_{label}(\bar{\rho}) = N$ ). By denoting the set of runs consistent with  $(\sigma_o, t)$  as  $\mathcal{R}(Rec(G_f), (\sigma_o, t))$ , an approach for fault diagnosis can be generated by the following rules:

- $\phi((\sigma_o, t)) = N$  (resp.,  $\phi((\sigma_o, t)) = F$ ) if  $f_{label}(\bar{\rho}) = N$  (resp.,  $f_{label}(\bar{\rho}) = F$ ) holds for each  $\bar{\rho} \in \mathcal{R}(Rec(G_f), (\sigma_o, t))$ ;
- otherwise, it is  $\phi((\sigma_o, t)) = U$ .

In simple words, the fault diagnosis of the faulty behaviour  $f$  can be done via exploring all the runs in  $Rec(G_f)$ . If the fault label of each run  $\bar{\rho} \in \mathcal{R}(Rec(G_f), (\sigma_o, t))$  is  $f_{label}(\bar{\rho}) = F$  (resp.,  $f_{label}(\bar{\rho}) = N$ ), we may conclude that  $f$  has (resp., has not) been occurred for sure; otherwise,  $f$  may or may not have been occurred.

*Example 4.* Consider the TFA  $G = (X, E, \Delta, \Gamma, X_0)$  in Fig. 1 with  $E_o = \{a, b, c\}$  and  $\Delta_{fault} = \{(x_0, c, x_2), (x_1, d, x_3)\}$ . Given a timed observation  $(\sigma_o, 4)$ , where  $\sigma_o = (c, 1)(b, 2)(c, 3.5)$ , the diagnosis procedures from  $t = 0$  to  $t = 4$  is summarized in Table 3. We explain the process of diagnosis while the observation  $(\sigma_o, t)$  is progressively updated over time as follows.

- The TFA  $G$  produces observation  $(\sigma_o, t) = (\lambda, t)$  for  $t \in [0, 1]$ . The union of ending states of all runs in  $\mathcal{R}(Rec(\lambda, 1))$  is  $\{(x_0, [1, 1], N), (x_f, [0, 0], F)\}$  at  $t \in [1, 1]$ . Thus it is  $\phi(\lambda, 1) = U$ .
- The TFA  $G$  produces observation  $(\sigma_o, t) = ((c, 1), t)$  for  $t \in [1, 2]$ . The union of ending states of all runs in  $\mathcal{R}(Rec((c, 1), 2))$  is  $\{(x_1, [1, 2], N), (x_2, [1, 1], F), (x_3, [1, 2], F)\}$  at  $t \in (2, 3)$ . Thus it is  $\phi(((c, 1), 2)) = U$ .
- The TFA  $G$  produces observation  $(\sigma_o, t) = ((c, 1)(b, 2), t)$  for  $t \in [2, 3.5]$ . The union of ending state of all runs in  $\mathcal{R}(Rec((c, 1)(b, 2), 3.5))$  is  $\{(x_0, (1, 2], N), (x_f, [0, 0], F), (x_1, [1, 2], F), (x_3, [1, 2], F)\}$  at  $t \in (3, 4)$ . Thus it is  $\phi(((c, 1)(b, 2), 3.5)) = U$ .
- The TFA  $G$  produces observation  $(\sigma_o, t) = ((c, 1)(b, 2)(c, 3.5), t)$  for  $t \in [3.5, 4]$ . The union of ending state of all runs in  $\mathcal{R}(Rec(G_f), (\sigma_o, t))$  is  $\{(x_2, (0, 1), F)\}$  at  $t \in [4, 4]$ . Thus it is  $\phi(((c, 1)(b, 2)(c, 3.5), 4)) = F$ .  $\square$

## 7. CONCLUSIONS

In this paper we consider timed automata endowed with a single clock that is reset upon an event occurrence. A time interval is associated with each transition specifying at which clock values it may occur. We consider time semantics that impose constraints to the dwell time spent at each state of a TFA. A timed word generated by a TFA is defined as a sequence of pairs (event, time instant at which the event occurs). Assuming that faulty behaviours are described by means of timed transitions, we deal with the problem of fault diagnosis in function of measured timed observations. The proposed solution is based on a zone automaton that provides a purely discrete description of the behaviour of the timed DES with faults. The problem of fault diagnosis is solved by constructing a fault recognizer that recognizes the occurrence of faults. The fault diagnosis approach allows one to compute the diagnosis state for each timed observation, which consists in a timed sequence of observed events. As a future work, we plan to explore the diagnosability of TFA.

Table 3. Diagnosis of the TFA  $G$  in Fig. 1 with  $E_f = \{c, d\}$  and  $(\sigma_o, t)$ ,  $t \in [0, 4]$ .

| $\sigma_o$             | Time instant $t$ | $\bigcup_{\bar{\rho} \in \mathcal{R}(\text{Rec}(G_f), (\sigma_o, t))} q_{en}(\bar{\rho})$ | $\phi((\sigma_o, t))$ |
|------------------------|------------------|---|-----------------------|
| $\lambda$              | [0,0]            | $\{(x_0, [0, 0], N)\}$  | $N$                   |
|                        | (0,1)            | $\{(x_0, (0, 1), N)\}$  | $N$                   |
|                        | [1,1]            | $\{(x_0, [1, 1], N), (x_f, [0, 0], F)\}$  | $U$                   |
| $(c, 1)$               | [1,1]            | $\{(x_1, [0, 0], N), (x_2, [0, 0], F), (x_3, [0, 0], F)\}$                                | $U$                   |
|                        | (1,2)            | $\{(x_1, (0, 1), N), (x_2, (0, 1), F), (x_3, (0, 1), F)\}$                                | $U$                   |
|                        | [2,2]            | $\{(x_1, [1, 2], N), (x_2, [1, 1], F), (x_3, [1, 2], F)\}$                                | $U$                   |
| $(c, 1)(b, 2)$         | [2,2]            | $\{(x_0, [0, 0], N), (x_1, [0, 0], F), (x_3, [0, 0], F)\}$                                | $U$                   |
|                        | (2,3)            | $\{(x_0, (0, 1), N), (x_1, (0, 1), F), (x_3, (0, 1), F)\}$                                | $U$                   |
|                        | [3,3]            | $\{(x_0, [1, 1], N), (x_f, [0, 0], F), (x_1, [1, 2], F), (x_3, [1, 2], F)\}$              | $U$                   |
|                        | (3,4)            | $\{(x_0, (1, 2), N), (x_f, [0, 0], F), (x_1, [1, 2], F), (x_3, [1, 2], F)\}$              | $U$                   |
| $(c, 1)(b, 2)(c, 3.5)$ | (3,4)            | $\{(x_2, [0, 0], F), (x_2, (0, 1), F)\}$  | $F$                   |
|                        | [4,4]            | $\{(x_2, (0, 1), F)\}$  | $F$                   |

In particular, it is worthy investigating if one can certainly detect a fault in a given time interval, which could be of great interest for real time systems.

#### REFERENCES

- R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
- R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229. Springer, 1992.
- R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
- N. Bertrand, S. Haddad, and E. Lefaucheux. Foundation of diagnosis and predictability in probabilistic systems. In *Proceedings of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 29, pages 417–429, 2014.
- C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Springer, 2009.
- C. Gao, D. Lefebvre, C. Seatzu, Z. Li, and A. Giua. A region-based approach for state estimation of timed automata under no event observation. In *Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation*, volume 1, pages 799–804. IEEE, 2020.
- A. Giua, C. Seatzu, and D. Corona. Marking estimation of Petri nets with silent transitions. *IEEE Transactions on Automatic Control*, 52(9):1695–1699, 2007.
- C. Hadjicostis. *Estimation and Inference in Discrete Event Systems*. Springer, 2020.
- T. Henzinger. The theory of hybrid automata. In *Verification of Digital and Hybrid Systems*, pages 265–292. Springer, 2000.
- D. Lefebvre and C. Hadjicostis. Privacy and safety analysis of timed stochastic discrete event systems using markovian trajectory-observers. *Discrete Event Dynamic Systems*, pages 1–28, 2020.
- M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.
- M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 4(2):105–124, 1996.
- S. Shu, F. Lin, and H. Ying. Detectability of discrete event systems. *IEEE Transactions on Automatic Control*, 52(12):2356–2359, 2007.
- Y. Tong, Z. Li, and A. Giua. On the equivalence of observation structures for Petri net generators. *IEEE Transactions on Automatic Control*, 61(9):2448–2462, 2015.

- S. Tripakis. Fault diagnosis for timed automata. In *Proceedings of the 7th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems: Co-sponsored by IFIP WG 2.2*, pages 205–224, 2002.
- P. Bouyer, S. Jaziri, and N. Markey. Efficient timed diagnosis using automata with timed domains. In *Proceedings of the 18th Workshop on Runtime Verification (RV'18)*, 11237: 205–221, 2018.
- P. Bouyer, L. Henry, S. Jaziri, T. Jérón and N. Markey. Diagnosing timed automata using timed markings. *International Journal on Software Tools for Technology Transfer*, 23: 229–253, 2021.
- P. Bouyer, C. Fabrice and D. D'Souza. Fault diagnosis using timed automata. In *Proceedings of International Conference on Foundations of Software Science and Computation Structures*, 219–233, 2005.
- J. Lunze and P. Supavatanakul. Diagnosis of discrete-event system described by timed automata. In *IFAC Proceedings Volumes*, 35(1): 77–82, 2002.
- P. Supavatanakul and J. Lunze. Diagnosis of timed automata based on an observation principle. In *IFAC Proceedings Volumes*, 39(13): 1270–1275, 2006.
- Y. Brave and M. Heymann. Formulation and control of real time discrete event processes. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 1131–1132, 1988.
- J. S. Ostroff. Deciding properties of timed transition models. *IEEE Transactions on Parallel and Distributed Systems*, 1(02): 170–183, 1990.
- B. Brandin and W. M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic control*, 39(2): 329–342, 1994.
- D. Lefebvre, Z. Li, and Y. Liang. Verifiers for the detection of timed patterns in discrete event systems. In *Proceedings of IFAC-PapersOnLine*, 55(28): 264-269, 2022.