Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-24 (2018) 1059–1066

SMT-based Diagnosability Analysis of Real-Time Systems

Lulu He *, **Lina Ye** **, **Philippe Dague** ***

* *School of Computer Technology, Xidian University, China (e-mail: llhe@stu.xidian.edu.cn).*
** *LRI, Univ. Paris-Sud and CentraleSupélec, Univ. Paris-Saclay, Orsay, France (e-mail: firstname.lastname@lri.fr)*
*** *LRI, Univ. Paris-Sud and CNRS, Univ. Paris-Saclay, Orsay, France (e-mail: firstname.lastname@lri.fr)*

Abstract: Fault diagnosis is a crucial and challenging task in the automatic control of complex systems, whose efficiency depends on a system property called diagnosability. Diagnosability describes the capability of a system to determine with certainty whether a fault has effectively occurred based on a sequence of observations. The diagnosability problem of discrete event systems has received considerable attention in the literature. However, up to now little work takes into account explicit time constraints during this analysis, which are however naturally present in real-life systems and thus cannot be neglected considering their impact on this property. In this paper, we first rephrase diagnosability definition for timed automata before showing the impact of time constraints on this property. Then we propose a new SMT-based approach to verify bounded time diagnosability on timed automata. The idea is to encode the sufficient and necessary condition for diagnosability in SMT. Finally, the experimental results are presented to show the efficiency of the SMT-based paradigm.

Keywords: fault diagnosis, formal verification, embedded systems, timed automata, mathematical models

1. INTRODUCTION

Fault diagnosis is a crucial and challenging task in the automatic control of complex systems (Reiter (1987); Struss (1997); Debouk et al. (2002); Pencolé and Cordier (2005); Console et al. (2007); Grastien et al. (2007); Bertrand et al. (2014); Haar et al. (2013)), whose efficiency depends on a system property called diagnosability. The diagnosability problem for discrete event systems (DESs) has received considerable attention in the literature. Diagnosability describes the system ability to determine whether a fault has effectively occurred based on the observations. In a given DES, the existence of two infinite behaviors, with the same observations but exactly one containing the considered fault, violates diagnosability. The existing work searches for such ambiguous behaviors both in centralized (Sampath et al. (1995); Jiang et al. (2001); Rintanen (2007); Cimatti et al. (2003); Germanos et al. (2014)) and distributed (Pencolé (2004); Schumann and Huang (2008); Ye and Dague (2010)) ways. The most classical method is to construct a structure called twin plant that captures all pairs of observationally equivalent behaviors to directly check the existence of such ambiguous pairs.

Note that up to now little work takes into account explicit time constraints during diagnosability analysis, which are however naturally present in real-life systems (e.g., transmission delays, response time, etc...) and thus cannot be neglected considering their impact on this property. For example, two ambiguous behaviors for a normal DES may be distinguishable by adding explicit time constraints, e.g., the delay between some two successive observable events is always different in them. Since classical models (e.g., finite automata, Petri nets) cannot express such real-time constraints, we will analyze diagnosability on timed automata (TA), which are one of the most studied models for real-time systems (RTSs) since their introduction by Alur and Dill (1994). In such a model, quantitative properties of delays between events can easily be expressed. Executions traces of TA are modeled by timed words, i.e., sequences of events which are attached to the time at which they occur. Hence, TA is seen as acceptors of languages of timed words.

In this paper, we propose a new approach to efficiently encode diagnosability problem in Satisfiability Modulo Theories (SMT). SMT is an extended form of Boolean satisfiability (SAT), where literals are interpreted w.r.t. a background theory. In terms of expressiveness, with SMT, one can provide a natural symbolic representation for TA. Precisely, the discrete parts of TA can be represented by the Boolean part and the continuous clock evolutions can be expressed by the theory description. In terms of the verification techniques, the approach allows for a direct extension of the SAT-based model checking algorithms, taking advantage of the advanced features of modern SMT solvers, such as incrementality, unsatisfiable core extraction, etc.

We provide several contributions in the domain of SMT-based formal verification of diagnosability on TA. First, diagnosability definition is rephrased in the temporal framework by taking into account time constraints. Second, a

10.1016/j.ifacol.2018.09.721

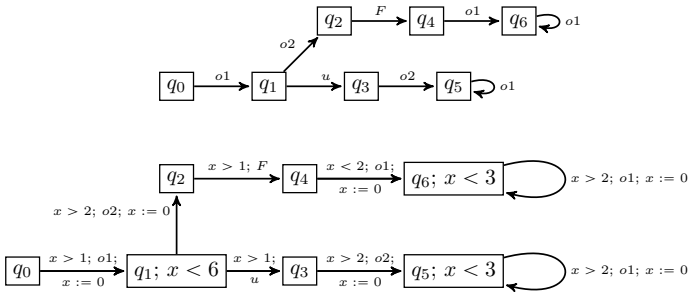

Fig. 1. A system example modeled by a finite automaton (top) and by a timed automaton (bottom).

sufficient and necessary condition for diagnosability of TA is proposed before being encoded in SMT for its bounded time variant. Third, we also give experimental results on benchmarks. And finally our approach is compared with the existing work in the literature.

2. PRELIMINARIES

In this section, the classical diagnosability is recalled before introducing TA in order to rephrase its definition by taking into account time constraints. Then, a necessary and sufficient condition is provided to verify it.

2.1 Diagnosability for Finite Automata

In a classical setting, a DES is modeled by a finite automaton.

Definition 1. (System Model) A DES is modeled as a *finite automaton* (FA), denoted by $G = (Q, \Sigma, \delta, q_0)$, where:

- Q is a finite set of states;
- Σ is a finite set of events;
- $\delta \subseteq Q \times \Sigma \times Q$ is a finite set of transitions;
- $q_0 \in Q$ is the initial state;

The set of events Σ is divided into three disjoint parts: $\Sigma = \Sigma_o \uplus \Sigma_u \uplus \Sigma_f$, where Σ_o is the set of observable events, Σ_u the set of unobservable normal events and Σ_f the set of unobservable fault events. For the transition set, it is easy to extend $\delta \subseteq Q \times \Sigma \times Q$ to $\delta \subseteq Q \times \Sigma^\star \times Q$ as follows:

- $(q, \epsilon, q) \in \delta$, where ϵ is the null event;
- $(q, se, q1) \in \delta$ if $\exists q' \in Q, (q, s, q') \in \delta$ and $(q', e, q1) \in \delta$, where $s \in \Sigma^\star, e \in \Sigma$.

Consider the simple system model depicted in the top part of Figure 1, where $\Sigma_o = \{o1, o2\}$, $\Sigma_u = \{u\}$, $\Sigma_f = \{F\}$, and q_0 is the initial state.

Given a system model G, the prefix-closed language $L(G) \subseteq \Sigma^\star$, which describes the normal and faulty behaviors of the system, is the set of words produced by G: $L(G) = \{s \in \Sigma^\star \mid \exists q \in Q, (q_0, s, q) \in \delta\}$. We assume that from any state starts at least one transition, so that $L(G)$ is live. In the following, we call a word from $L(G)$ a *trajectory* in G and a sequence $q_0\sigma_0q_1\sigma_1...$, with $(q_i, \sigma_i, q_{i+1}) \in \delta$ for all i, a *path* in G whose associated trajectory $\sigma_0\sigma_1...$ is thus the *trace* of this path (path and trajectory are defined identically starting from any *reachable* state, i.e., any state reached from q_0 by a path). Given $s \in L(G)$, we denote the post-language of $L(G)$ after s by $L(G)/s$, formally defined as: $L(G)/s = \{t \in \Sigma^\star \mid s.t \in L(G)\}$. The projection of the trajectory s to observable events is denoted by $P(s)$. These notions extend to infinite paths and trajectories that we will use.

A usual assumption about partially observable system models is that there is no unobservable infinite behavior.

Assumption 1: (Observably living system) The DES G is *observably living*, i.e., any infinite trajectory has infinitely many observable events occurrences.

Intuitively, a predefined fault is considered as diagnosable if one can be sure about its occurrence after sufficient observations, which can be formally defined as follows (Sampath et al. (1995)), where s^F denotes a trajectory s that ends with the fault F.

Definition 2. (Diagnosability of FA) A fault $F \in \Sigma_f$ is *diagnosable* in a DES G iff

$$\exists k \in N, \forall s^F \in L(G), \forall t \in L(G)/s^F, (|t| \geqslant k \Rightarrow \forall p \in L(G), (P(p) = P(s^F.t) \Rightarrow F \in p)).$$

The above definition states that for each trajectory s^F in G, for each t that is an extension of s^F with sufficient events, every trajectory p in G that is observationally equivalent to $s^F.t$ should contain in it F. In other words, the diagnosability checking consists in verifying the non-existence of a pair of trajectories p and p' satisfying the following three conditions: 1) p contains F and p' does not; 2) p and p' are infinite; 3) $P(p) = P(p')$. Such a pair is called a *critical pair* (Cimatti et al. (2003)), which has been proven to violate Definition 2 and thus witnesses non-diagnosability. Consider the top example of Figure 1, where the pair of trajectories $o1.o2.F.o1^\omega$ and $o1.u.o2.o1^\omega$ is a critical pair since it satisfies the above three conditions. In other words, once we observe the sequence of events $o1.o2.o1^\omega$, we can never be sure about the occurrence of the fault since in this system, there does exist one trajectory containing the fault and the other without the fault, while both have exactly this same sequence of observations.

Theorem 1. Given G a FA, a fault F is diagnosable in G iff there is no critical pair w.r.t. F in G.

For the sake of simplicity, we conventionally assume, in this paper, that there is only one fault F (with multiple occurrences), i.e. $\Sigma_f = \{F\}$, which can be directly extended to the case of a set of faults by applying the approach as many times as the number of faults.

2.2 Diagnosability for Timed Automata

TA constitute a theory for modeling and verifying real-time systems. A TA is essentially a finite automaton, thus with a finite set of states and a finite set of labeled transitions between them, extended with a finite set of real-valued variables modeling *clocks*. During a run of a TA, clock values are initialized with zero when starting in the initial state, and then are increased all with the same speed. Clock values can be compared to constants or between them. These comparisons form guards that may enable or disable instantaneous transitions and by doing so constrain the possible behaviors of the TA. Furthermore, clocks can be also reset to zero on some of the transitions.

Before introducing the formal definition of TA, we first give the set of possible clock constraints considered in this paper. The timing constraints that are handled here constitute the set of so-called diagonal constraints, formally described by:

$$g ::= true \mid x \bowtie c \mid x - y \bowtie c \mid g \wedge g,$$
where x, y are clock variables, c is a constant and
$$\bowtie \in \{<, \leqslant, =, \geqslant, >\}.$$

Note that a TA allowing such clock constraints is exponentially more concise than its classical variant with only diagonal-free constraints (where the comparison can be done only between a clock value and a constant) but both have same expressiveness. Let X be a finite set of clock variables. A clock valuation over X is a function $v : X \to R$, where R denotes the set $\mathbb{R}_+$ of non-negative real numbers (actually, for implementation, the set $\mathbb{Q}_+$ of non-negative rational numbers is used to have an exact computer representation). Then the set of all clock valuations over X is denoted by R^X and the set of time constraints over X by $\mathbb{C}(X)$, where such a constraint is given by a collection of clock constraints. If a clock valuation v satisfies the time constraint g, then it is denoted by $v \models g$. In the following, we denote $[\![g]\!]$ the set of clock valuations that satisfy g, i.e., $[\![g]\!] = \{v \in R^X \mid v \models g\}$.

Definition 3. (Timed Automaton) A *timed automaton* (TA) is a tuple $A = (Q, \Sigma, X, \delta^X, q_0, I)$, where:

- Q is a finite set of states;
- Σ is a finite set of events;
- X is a finite set of clock variables;
- $\delta^X \subseteq Q \times \mathbb{C}(X) \times \Sigma \times 2^X \times Q$ is a finite set of transitions;
- $q_0 \in Q$ is the initial state;
- $I : Q \to \mathbb{C}(X)$ is a function that assigns invariants to states.

The example depicted at the bottom of figure 1 is a TA obtained by adding some time constraints to the system model at the top of the same figure. In this example, x is a clock variable that is used to impose certain period between events or to restrain the possible time during which one is allowed to stay in some states. For example, $(q_0, x > 1, o1, \{x\}, q_1) \in \delta^X$ means that only when the guard $x > 1$ is satisfied, i.e., the clock value is greater than 1, the event $o1$ can occur, inducing an instantaneous state change from q_0 to q_1 and simultaneously the reset to 0 of the clock x. We denote this transition also as $q_0 \xrightarrow{x>1;\ o1;\ x:=0} q_1$. Furthermore, $I(q_1) = x < 6$ imposes that we can stay in the state q_1 only when the clock value is smaller than 6. In other words, once the invariant ceases to be satisfied, one is obliged to leave the corresponding state (for readability reasons, we did not represent invariants that define also sojourn time upper bounds in states q_2, q_3, q_4).

We call a state with a clock valuation an extension state, i.e., (q, v) with $q \in Q$ and $v \in R^X$. Let $t \in R$, the valuation $v + t$ is defined by $(v + t)(x) = v(x) + t, \forall x \in X$. Suppose $X' \subseteq X$, we denote by $v[X' \leftarrow 0]$ the valuation such that $\forall x \in X'$, $v[X' \leftarrow 0](x) = 0$ and $\forall x \in X \backslash X'$, $v[X' \leftarrow 0](x) = v(x)$. A TA gives rise to an infinite transition system with two types of transitions between extension states. One is a *time* transition representing time passage in the same state q, during which the invariant $inv = I(q)$ for q should be always respected. The other one is a *discrete* transition issued from a labeled transition $q \xrightarrow{g;\ \sigma;\ r} q'$ for TA, associated with an event σ, which is fired (a necessary condition being that the guard g is satisfied) and should be executed instantaneously, i.e., the clock valuation cannot be modified by the transition itself but only by the reset to 0 of those clock variables belonging to r, if any. In the following, both are denoted by $(q, v) \xrightarrow{\nu} (q', v')$, where $\nu \in \Sigma \cup R$. Thus, if $\nu \in \Sigma$, then v should satisfy the guard g in the corresponding TA transition and $v' = v[r \leftarrow 0]$ for r the clock variables reset to 0 in this transition, if any. Otherwise, if $\nu \in R$, then $q' = q$ and $v' = v + \nu$, where all of $v + t$, for $0 \leqslant t \leqslant \nu$, should satisfy the invariant inv associated to the state q.

Given A a TA, a sequence of such transitions $(q_0, v_0 = 0) \xrightarrow{\nu_1} (q_1, v_1) \ldots \xrightarrow{\nu_n} (q_n, v_n)$ is a feasible execution in A if $\forall i \in \{0, ..., n-1\}, (q_i, v_i) \xrightarrow{\nu_{i+1}} (q_{i+1}, v_{i+1})$ is either a time or a discrete transition in it. Then the word $\nu_1 ... \nu_n \in (\Sigma \cup R)^\star$ is called a *timed trajectory* or a *run*. This extends to infinite sequences and trajectories. The set of timed trajectories for A is denoted by $L(A)$, assumed to be live. By summing up successive time periods, we can always assume that between any two successive events there is exactly one time period, i.e., periods and events alternate in a timed trajectory. For sake of simplicity, we will assume that this time period between two successive events is not null. Actually, our SMT-based diagnosability checking extends to systems where multiple (ordered) events occur in zero time by adopting the superdense model of time (Cataldo et al. (2006)) and processing instantaneous observable events at a given time with an order semantics. For ρ a timed trajectory, we denote by $time(\rho) \in R \cup \{+\infty\}$ the total time duration for ρ, i.e., $time(\rho) = \sum_{\nu_i \in R \wedge \nu_i \in \rho} \nu_i$. We redefine a projection operator P for TA as follows. Given a timed trajectory ρ and a set of events $\Sigma' \subseteq \Sigma, P(\rho, \Sigma')$ is the timed trajectory obtained by erasing from ρ all events not in Σ' and summing the periods between successive events in the resulting sequence. For example, if $\rho = 2\ o1\ 3\ u\ 2\ o2\ 3\ o1$, then $P(\rho, \{o1, o2\}) = 2\ o1\ 5\ o2\ 3\ o1$. In the following, Σ is divided into three disjoint parts as above and we simply denote $P(\rho)$ the projection of the timed trajectory ρ to observable events, i.e., $P(\rho) = P(\rho, \Sigma_o)$.

We make for TA the analog assumption done for DES about the necessity to observe any infinite behavior.

Assumption 2: (Timed observably living system) The TA A is *timed observably living*, i.e., it has no time infinite execution from a reachable state without any observable event, and thus any time infinite timed trajectory ρ (time$(\rho) = +\infty$) has infinitely many observable events occurrences.

This implies in particular that the system cannot stay an infinitely long time in a same state.

We rephrase a useful notion, originally introduced by Tripakis (2002).

Definition 4. (Δ-*faulty* runs) Given A a TA, let $\rho = \nu_1 \nu_2 \ldots$ be a timed trajectory; ρ is faulty if, for some $i \in \{1, ...\}$, $\nu_i = F$. Let then j be the smallest i such that $\nu_i = F$ and let $\rho' = \nu_{j+1} \ldots$. We denote the *period*

from (the first occurrence of) *fault F in* ρ by $time(\rho, F)$ = time(ρ'). If time$(\rho, F) > \Delta$, where $\Delta \in R$, then we say that more than Δ time units pass after the first occurrence of F in ρ, or, in short, that ρ is Δ-*faulty*.

Notice that we chose in the definition greater than Δ instead of greater or equal because some technical aspects in the encoding become easier, but there is no difference in substance for the following, as a Δ-faulty run (for $>$) is Δ-faulty (for $\geqslant$) and a Δ-faulty run (for $\geqslant$) is Δ'-faulty (for $>$) for any $\Delta' < \Delta$. Now we adapt Definition 2 to define diagnosability of TA.

Definition 5. (Diagnosability of TA) Given a TA A and a fault F:

(1) given $\Delta \in R$, F is Δ-*diagnosable* in A iff
$\forall \rho \in L(A)\ (\rho\ \Delta\text{-}faulty \Rightarrow$
$\forall \rho' \in L(A)\ (P(\rho) = P(\rho') \Rightarrow F \in \rho'))$.
(2) F is *diagnosable* in A iff
$\exists \Delta \in R$ such that F is Δ-*diagnosable* in A.

Now a critical pair in the timed framework, called *timed critical pair* in the following, is a pair of timed trajectories ρ and ρ' satisfying the following conditions: 1) ρ contains F and ρ' does not; 2) time(ρ) = time(ρ') = $+\infty$; 3) $P(\rho) = P(\rho')$. Note that this implies that ρ and ρ' are infinite. It is obvious to prove that the existence of such a pair violates the diagnosability for TA in Definition 5 (2), and the converse has been proved too by Tripakis (2002).

Theorem 2. Given A a TA, F is diagnosable in A iff there is no timed critical pair w.r.t. F in A.

Thus, in a way similar to FA, the diagnosability verification for TA consists in checking the non-existence of timed critical pairs. For example, consider the system modeled by the TA of Figure 1 (bottom part). The system has both *faulty* behaviors (where F occurs) and normal ones. Indeed, in all behaviors, the observable events occur in the same order, i.e., $o1.o2.o1^*$, as in the FA without time constraints (top part of Figure 1). However, in every faulty behavior, the time duration between the successive observable events $o2$ and $o1$ is smaller than 2 time units. While in every normal behavior, this duration is greater than 2. Thus, a diagnoser observing $o2$ and then $o1$, and measuring the duration between them can tell with certainty whether a fault has occurred or not. One can clearly see that adding time constraints sometimes makes a non-diagnosable system diagnosable by distinguishing temporally the two trajectories that are considered as a critical pair in the untimed setting, e.g., with different durations between two successive observable events.

3. ENCODING BOUNDED DIAGNOSABILITY

Recall that the existence of timed critical pairs violates diagnosability of TA. However, the time duration for both trajectories should be infinitely long. For a real system that satisfies diagnosability defined in Definition 5 (2), i.e., for which such a timed critical pair does not exist, it is possible that a faulty trajectory may be distinguished from normal ones only after an unacceptably long time. To be more practical when applied to real systems, we use $\Delta \in R$ to represent a time upper bound after the fault occurrence to identify it, i.e., we rest on the concept of Δ-diagnosability as defined in Definition 5 (1). So, saying that F is not Δ-diagnosable means that it exists a Δ-faulty run which cannot be distinguished by observation from at least one normal timed trajectory. We thus define formally a Δ-critical pair, whose existence violates Δ-diagnosability.

Definition 6. (Δ-critical pair) Given a TA A, the considered fault F and $\Delta \in R$, two timed trajectories ρ, $\rho' \in L(A)$ are called a Δ-*critical pair* (Δ-CP) if the following conditions are satisfied: 1) ρ is Δ-faulty; 2) ρ' does not contain F; 3) $P(\rho) = P(\rho')$.

Theorem 3. Given A a TA, F is Δ-diagnosable iff there is no Δ-critical pair w.r.t. F in A.

From Theorem 2, diagnosability (resp. timed critical pair) corresponds thus to $+\infty$-diagnosability (resp. $+\infty$-critical pair), if we extend Definition 4 to $\Delta = +\infty$ by replacing $> \Delta$ by $= +\infty$. Consider the case of an infinite Δ-CP. As time$(\rho, F) > \Delta$ (possibly $+\infty$), it exists a finite Δ-faulty prefix of ρ ending by a time transition. It results that we can replace, in the statement of Theorem 3, Δ-CP by finite Δ-CP ending by a time transition.

3.1 Encoding TA

In this section, we will show how to logically encode in SMT Δ-CP for a TA such that, if the SMT solver finds a model for the proposed logic formula, then the considered fault is not Δ-diagnosable in this TA (and conversely if the formula length reaches a given theoretical upper bound). In this case, the corresponding model, that is actually a Δ-CP, is finally returned. As we saw it is enough to look for a finite Δ-CP, it is thus possible to encode it by a finite formula. We will then consider bounded length timed trajectories with length parameter k, i.e., diagnosability checking is done on timed trajectories with length k, denoted by $L^k(A) = \{\rho \in L(A) \mid |\rho| = k\}$. As explained in Section 2.2, we can assume that (non-null) time and discrete transitions alternate in any timed trajectory. Hence, we rewrite $(q, v) \xrightarrow{t} (q, v'') \xrightarrow{\sigma} (q', v')$, where $t \in R$, $t > 0$, and $\sigma \in \Sigma$, as $(q, v) \xrightarrow{t,\sigma} (q', v')$. In the following, we consider this kind of combined time-discrete transition during the encoding. Accordingly, a timed trajectory of length k is a finite sequence $(t_0, \sigma_0), (t_1, \sigma_1), ..., (t_{k-1}, \sigma_{k-1})$, where $t_i \in R$, $t_i > 0$, $\sigma_i \in \Sigma$, and $\forall i$, $0 \leqslant i \leqslant k-1$, $(q_i, v_i) \xrightarrow{t_i,\sigma_i} (q_{i+1}, v_{i+1})$ is allowed by A. As seen above, we can assume that the timed trajectory ends by a time transition, that we will represent by setting $\sigma_{k-1} = \epsilon$ as an unobservable event. For the bottom example of Figure 1, one 4-length timed trajectory is $\rho = (1.5, o1), (3, u), (0.5, o2), (1, \epsilon)$ that is witnessed by the feasible execution $(q_0, x = 0) \xrightarrow{1.5,o1} (q_1, x = 0) \xrightarrow{3,u} (q_3, x = 3) \xrightarrow{0.5,o2} (q_5, x = 0) \xrightarrow{1,\epsilon} (q_5, x = 1)$.

Given a TA $A = (Q, \Sigma, X, \delta^X, q_0, I)$, we encode essential static parts in A as follows:

- the set of states is encoded by positive integers through the function $E_Q : Q \to Q^E = \{1, ..., \|Q\|\}$.
- the set of events is encoded by positive integers $E_\Sigma : \Sigma \to \Sigma^E = \{1, ..., \|\Sigma\|\}$, where $\Sigma^E = \Sigma_o^E \uplus \Sigma_u^E \uplus \Sigma_f^E$, corresponding to $\Sigma = \Sigma_o \uplus \Sigma_u \uplus \Sigma_f$. We assume that normal events $\Sigma_n = \Sigma_o \uplus \Sigma_u$ are encoded by

integers from 1 to $\|\Sigma_n\|$ and fault events by integers from $\|\Sigma_n\|+1$ to $\|\Sigma\|$.
- the set of symbolic transitions is encoded by a set of tuples $E_{\delta^X} : \delta^X \rightarrow \delta^E = (Q^E \times \mathbb{C}(X) \times \Sigma^E \times 2^X \times Q^E)$ such that $E_{\delta^X}(q,g,\sigma,r,q') = (E_Q(q), g, E_\Sigma(\sigma), r, E_Q(q'))$.

3.2 Encoding Bounded Diagnosability

In this section, given a TA with fault F and Δ, we show how to define, for arbitrary integers $k, \hat{k}$, a formula $\Psi_\Delta^{k,\hat{k}}$ whose satisfiability is equivalent to the existence of a Δ-CP $(\rho, \hat{\rho})$ with $|\rho| = k$ and $|\hat{\rho}| = \hat{k}$. In order to describe this formula as intuitively as possible, we present it with different separate parts. Since the satisfiability of $\Psi_\Delta^{k,\hat{k}}$ represents the existence of a Δ-CP, two timed trajectories ρ and $\hat{\rho}$ are concerned, which makes this property more complicated to encode than for example safety properties, for which it is not necessary to compare between different timed trajectories. To distinguish the value of variables between the two timed trajectories, the variables equipped with a hat are associated to the normal trajectory $\hat{\rho}$ of length $\hat{k}$ while the variables without a hat are attached to the faulty trajectory ρ of length k.

- The integer-valued variables $e_0, ..., e_{k-1}$ (resp. $\hat{e}_0, ..., \hat{e}_{\hat{k}-1}$) encode the events in the faulty (resp. normal) timed trajectory (with e_{k-1} and $\hat{e}_{\hat{k}-1}$ encoding ϵ).
- The integer-valued variables $s_0, ..., s_k$ (resp. $\hat{s}_0, ..., \hat{s}_{\hat{k}}$) represent the states in the faulty (resp. normal) timed trajectory.
- The real-valued variables $t_0, ..., t_{k-1}$ (resp. $\hat{t}_0, ..., \hat{t}_{\hat{k}-1}$) encode the time periods in the faulty (resp. normal) timed trajectory.
- The real-valued variables $v_0^x, ..., v_k^x$, $\forall x \in X$ (resp. $\hat{v}_0^x, ..., \hat{v}_{\hat{k}}^x$) represent the values of the corresponding clock x in each state in the faulty (resp. normal) timed trajectory, initialized as 0, i.e., $v_0^x = \hat{v}_0^x = 0$.
- The real-valued variables $v_0^t, ..., v_k^t$ (resp. $\hat{v}_0^t, ..., \hat{v}_{\hat{k}}^t$) encode the values of an additional global clock that should be initialized as 0 but never be reset to 0.
- The additional real-valued variables $v_0^F, ..., v_k^F$ represent the time elapsed after the first fault occurrence in the faulty timed trajectory (-1 by convention before the fault occurrence).

Initialization The two timed trajectories should start in the initial state with the initialization of all clock variables.

- for the faulty timed trajectory:
$\Phi^{Init} := (\bigwedge_{x \in X \cup \{t\}} v_0^x = 0) \wedge (v_0^F = -1) \wedge (s_0 = E_Q(q_0))$
- for the normal timed trajectory:
$$\hat{\Phi}^{Init} := (\bigwedge_{x \in X \cup \{t\}} \hat{v}_0^x = 0) \wedge (\hat{s}_0 = E_Q(q_0))$$

Well-formedness of timed trajectories The well-formedness of timed trajectories represents the fact that each time period between two discrete transitions should be positive. Furthermore, the value of integer-valued variables representing all events in the two trajectories should be in $\{1...\|\Sigma\|\}$ for the faulty one and in $\{1...\|\Sigma_n\|\}$ for the normal one. This is encoded as follows.

- for the faulty timed trajectory:
$$\Phi^{WF} := (\bigwedge_{i=0}^{k-1} 0 < t_i) \wedge (\bigwedge_{i=0}^{k-1} 1 \leqslant e_i \wedge e_i \leqslant \|\Sigma\|) \wedge (\bigwedge_{i=0}^{k-1} 1 \leqslant s_i \wedge s_i \leqslant \|Q\|)$$
- for the normal timed trajectory:
$$\hat{\Phi}^{WF} := (\bigwedge_{i=0}^{\hat{k}-1} 0 < \hat{t}_i) \wedge (\bigwedge_{i=0}^{\hat{k}-1} 1 \leqslant \hat{e}_i \wedge \hat{e}_i \leqslant \|\Sigma_n\|) \wedge (\bigwedge_{i=0}^{\hat{k}-1} 1 \leqslant \hat{s}_i \wedge \hat{s}_i \leqslant \|Q\|)$$

Acceptance of timed trajectories We formalize here that the two timed trajectories represented by values for the predefined variables as described above should be accepted by A. Precisely, in each timed trajectory, each pair of adjacent states has to be connected by a transition that is allowed in A.

- for the faulty timed trajectory:
$$\Phi^{Acc} := (\bigwedge_{i=0}^{k-1} (\bigvee_{(s_i,g,e_i,r,s_{i+1}) \in \delta^E} [[g]]_i \wedge \P_i^r))$$
Here $[[g]]_i$ represents that the clock valuations after the i-th period in the faulty timed trajectory, i.e., $v_i^x + t_i$, should satisfy the guard g, such as:
  - $[[x \bowtie c]]_i := (v_i^x + t_i) \bowtie c$
  - $[[x - y \bowtie c]]_i := (v_i^x - v_i^y) \bowtie c$
  - $[[g_1 \wedge g_2]]_i := [[g_1]]_i \wedge [[g_2]]_i$

  $\P_i^r$ in the above expression formalizes the time progression, i.e., time transition, by resetting clocks in the subset r and by increasing all other clocks, including the global one (and also the time elapsed from the first fault occurrence if triggered) with the corresponding period t_i:
$$\P_i^r := (\bigwedge_{x \in r} v_{i+1}^x = 0) \wedge (\bigwedge_{x \in (X \setminus r) \cup \{t\}} v_{i+1}^x = v_i^x + t_i) \wedge (0 \leqslant v_i^F \Rightarrow v_{i+1}^F = v_i^F + t_i)$$
- for the normal timed trajectory:
$$\hat{\Phi}^{Acc} := (\bigwedge_{i=0}^{\hat{k}-1} (\bigvee_{(\hat{s}_i,g,\hat{e}_i,r,\hat{s}_{i+1}) \in \delta^E} [[\hat{g}]]_i \wedge \hat{\P}_i^r))$$
In a similar way, $[[\hat{g}]]_i$ for the normal timed trajectory is encoded as follows:
  - $[[\widehat{x \bowtie c}]]_i := (\hat{v}_i^x + \hat{t}_i) \bowtie c$
  - $[[\widehat{x - y \bowtie c}]]_i := (\hat{v}_i^x - \hat{v}_i^y) \bowtie c$
  - $[[\widehat{g_1 \wedge g_2}]]_i := [[\hat{g}_1]]_i \wedge [[\hat{g}_2]]_i$

  The following is the time progression for the normal timed trajectory:
$$\hat{\P}_i^r := (\bigwedge_{x \in r} \hat{v}_{i+1}^x = 0) \wedge (\bigwedge_{x \in (X \setminus r) \cup \{t\}} \hat{v}_{i+1}^x = \hat{v}_i^x + \hat{t}_i)$$

Fault The faulty timed trajectory contains a fault occurrence (with one fault type, the fault occurrence can be simplified as $\|\Sigma_n\| + 1 = e_i$)). Furthermore, after the first occurrence of a fault at step i, the value of the variable v_{i+1}^F is assigned to 0 to trigger counting the time elapsed from this fault occurrence (otherwise it stays equal to -1).

$$\Phi^F := (\bigvee_{i=0}^{k-1} \|\Sigma_n\| < e_i) \wedge (\bigwedge_{i=0}^{k-1} (v_i^F = -1 \Rightarrow ((\|\Sigma_n\| < e_i \Rightarrow v_{i+1}^F = 0) \wedge (e_i \leqslant \|\Sigma_n\| \Rightarrow v_{i+1}^F = -1))))$$

Equivalent observations The next condition to consider for a timed critical pair is that both timed trajectories should have exactly the same observations, i.e., the same observable event should be observed at the same time, which can be guaranteed by the global clock (that is never reset). This condition can be encoded as follows (extension allowing multiple simultaneous events occurrences would be done by requiring that, at a same given time, all simultaneous observable events occur in the same order in both trajectories).

$$\Phi^{\equiv_{obs}} := (\bigwedge_{\sigma\in\Sigma_o^E}(\bigwedge_{i=0}^{k-1}(e_i=\sigma \Rightarrow (\bigvee_{j=0}^{\hat{k}-1}(\hat{e}_j=\sigma \wedge v_i^t=\hat{v}_j^t)))) \wedge(\bigwedge_{i=0}^{\hat{k}-1}(\hat{e}_i=\sigma \Rightarrow (\bigvee_{j=0}^{k-1}(e_j=\sigma \wedge v_j^t=\hat{v}_i^t)))))$$

Time elapsed after fault The last condition is that the time elapsed after the first occurrence of a fault is greater than Δ, which is encoded as follows.

$$\Phi_\Delta^{Time} := v_k^F > \Delta$$

Bounded diagnosability checking We have formalized all conditions required for two timed trajectories to constitute a Δ-CP. Now we are ready to define:

$$\Psi_\Delta^{k,\hat{k}} := \Phi^{Init} \wedge \hat{\Phi}^{Init} \wedge \Phi^{WF} \wedge \hat{\Phi}^{WF} \wedge \Phi^{Acc} \wedge \hat{\Phi}^{Acc} \wedge \Phi^{F} \wedge \Phi^{\equiv_{obs}} \wedge \Phi_\Delta^{Time}$$

Note that for the sake of simplicity, in the proposed formula, we do not handle invariants of states. However, one can extend $\Psi_\Delta^{k,\hat{k}}$ by adding such constraints in a quite straightforward way. It suffices to enrich Φ^{Acc} and $\hat{\Phi}^{Acc}$ by verifying that the clock valuations in each state do not violate the corresponding invariant, which has to be done only when entering the state and leaving it.

Theorem 4. Given a TA A and a considered fault F, F is Δ-diagnosable iff, for all $k, \hat{k}$, $\Psi_\Delta^{k,\hat{k}}$ is not satisfiable in A.

Proof :
Suppose that $\Psi_\Delta^{k,\hat{k}}$ is satisfiable in A for some $k, \hat{k}$. From the satisfiability of $\Psi_\Delta^{k,\hat{k}}$, there exists at least a pair of timed trajectories satisfying $\Psi_\Delta^{k,\hat{k}}$, i.e., $\rho = (t_0, e_0), (t_1, e_1), ..., (t_{k-1}, e_{k-1})$ and $\hat{\rho} = (\hat{t}_0, \hat{e}_0), (\hat{t}_1, \hat{e}_1), ..., (\hat{t}_{\hat{k}-1}, \hat{e}_{\hat{k}-1})$. Since Φ^{Init}, $\hat{\Phi}^{Init}$, Φ^{WF}, $\hat{\Phi}^{WF}$, Φ^{Acc} and $\hat{\Phi}^{Acc}$ are satisfied, we have $\rho \in L^k(A)$, $\hat{\rho} \in L^{\hat{k}}(A)$ and $F \notin \hat{\rho}$. Then Φ^F encodes $F \in \rho$ as well as the fact that once the fault occurs, then the clock representing the time elapsed after the fault is triggered and Φ_Δ^{Time} encodes that this time exceeds Δ at the end of ρ. Hence, with $\Phi^{\equiv_{obs}}$ expressing the same observations for both timed trajectories, including observable events with their occurrence moment, it can be deduced that ρ and $\hat{\rho}$ constitute a Δ-CP according to Definition 6.
Conversely, if it exists a finite Δ-CP $(\rho, \hat{\rho})$, then one can deduce that ρ and $\hat{\rho}$ form a model of $\Psi_\Delta^{k,\hat{k}}$ with $k = |\rho|$ and $\hat{k} = |\hat{\rho}|$. Thus $\Psi_\Delta^{k,\hat{k}}$ is satisfiable in A.
We have established that it exists a finite Δ-CP w.r.t. F in A iff $\exists k, \hat{k}$ such that $\Psi_\Delta^{k,\hat{k}}$ is satisfiable. Then the result follows from Theorem 3 and the subsequent remark about Δ-CP finiteness.

The formula $\Psi_\Delta^{k,\hat{k}}$ is not very convenient as it depends on two independent parameters k and $\hat{k}$, the respective lengths of the timed trajectories constituting a possible Δ-CP (i.e., their total number of events, when only their observable events are in equal number, say h, in both trajectories). For implementation, it is better to use only one parameter l, the length of the critical pair as itself, considered as a timed trajectory in the product of A by itself, synchronized on observable events, i.e., $l = k + \hat{k} - h$. So, we replace k and $\hat{k}$ by l in the formulas above to obtain the new formula Ψ_Δ^l. The only change is in Φ^{Acc} and $\hat{\Phi}^{Acc}$, as now we cannot require that both individual trajectories (as components of the critical pair trajectory) progress by a transition at each step from i to $i+1$, but only that at least one progresses (actually both progress simultaneously only in case of a transition by an observable event). Thus we add in Φ^{Acc} and $\hat{\Phi}^{Acc}$ the possibility at each step for exactly one of both trajectories to not progress between i and $i+1$, which is equivalent to allow a transition $(0, \epsilon)$ of null time period and silent unobservable discrete transition ϵ.

For practical use, it is important to get an upper bound for l, to which it is enough to check Ψ_Δ^l unsatisfiability to conclude Δ-diagnosability of A. For diagnosability of a FA, as a critical pair has to be infinite, i.e., to contain a cycle, an upper bound is obtained as the maximal length of a path without cycle in the synchronized product of the automaton by itself, i.e., the states number of this product, namely $|Q|^2$. For Δ-diagnosability of a TA, as a Δ-CP has to have a period of time from fault F greater than Δ, it is not necessary to extend a pair under investigation when the faulty trajectory has already reached a period after fault at least Δ. If for the TA A, it exists a positive minimal sojourn time d_s in each state (which can be possibly deduced from the knowledge of invariants, guards and resets), then an upper bound for l is given by $l_F + \lceil \frac{\Delta}{d_s} \rceil$, where l_F is the maximal length of a trajectory without cycle up to a first occurrence of F. More generally, if it exists a positive minimal duration time d_c of any cycle in A, then an upper bound for l is given by $|Q|^2(\lceil \frac{\Delta}{d_c} \rceil + 1)$ (a length less than $|Q|^2$ to reach F and then, for each progression of the length by $|Q|^2$, a time at least d_c is spent).

4. IMPLEMENTATION AND VALIDATION

We have theoretically proved the correctness of our algorithm. To show its feasibility, we realized a prototype implementation done in Python by using the SMT solver Z3. All our experimental results are obtained by running our programs on a Mac OS laptop running on a 1.7 GHz Intel Core i7 processor with 8 Go 1600 MHz DDR3 of memory. We report on several versions of two benchmarks from the literature modified by adding different temporal constraints since their original versions were finite automata. Furthermore, considering that such literature examples are normally quite small, to show the scalability we tested also some hand-crafted examples whose state space was generated in a partially random way. One lit-

	l/Δ	$\|clauses\|$	$mem.$	$\|trans.\|$	$time$	$SAT?$
$hvac_1$	9/6	345756	33	15	26	Yes
$hvac_2$	5/3	496963	69	15	30	No
$hvac_3$	9/5	465393	52	15	21	Yes
$hvac_4$	6/3	737395	63	15	39	No
$hvac_5$	15/7	1353321	75	156	152	Yes
$hvac_6$	7/6	1709069	129	156	306	No
ex_1	21/15	246563	21	123	21	Yes
ex_2	9/6	903296	285	123	135	No

Table 1. Experimental results

erature benchmark is the HVAC system from Sampath et al. (1995). $hvac_2$ is a version modified by adding a clock to constrain the delay between the observable events *open_valve* and *close_valve* such that this delay is always different for a normal trajectory and for a faulty one. The system is diagnosable (No SAT). Then by modifying this delay on faulty trajectories without any constraint, the system becomes not diagnosable, as shown by example $hvac_1$. Then we added two clocks to constrain the delay between *open_valve* and *close_valve*, together with the delay between *start_pump* and *stop_pump* for both diagnosable and non-diagnosable systems, as shown in examples $hvac_4$ and $hvac_3$. Furthermore, $hvac_6$ and $hvac_5$ are corresponding hand-crafted versions. Similarly, we modified also the example presented in Jiang et al. (2001) for different versions, denoted by ex_i. Our hand-crafted examples are obtained by adding arbitrary number of events in the system in a way such that the verdict remains the same by enlarging the size of TA.

Table 1 shows part of our experimental results, where the 2nd column shows the upper bound l for the length of the critical pair (as explained in Section 3.2) corresponding to the time bound Δ after the fault chosen, the 3rd shows the size of the formula expressed by its number of clauses, the 4th is the required memory, the 5th presents the number of transitions of the system, the 6th gives the execution time in seconds while the last one is the satisfiability. From Table 1, one can see that our proposed approach is feasible since even the hand-crafted versions terminated well within the timeout (that we set to 900 seconds). Furthermore, the formula size of different examples shows that SMT solvers can check relatively large formulas for satisfiability. As usual our approach is more efficient to verify bounded non-diagnosability, i.e., SAT cases. However, note that bounded non-diagnosability does not mean non-diagnosability, while bounded diagnosability implies diagnosability. This is why we chose bigger Δ's for SAT cases (bounded non-diagnosability).

5. RELATED WORK

Diagnosability is a property introduced to remove the ambiguity of diagnosis decisions. The classical diagnosability methods consist in checking the existence of two indistinguishable behaviors, i.e., a critical pair, whose absence witnesses diagnosability. In the literature, to verify the diagnosability of DESs, the first way is to construct a deterministic automaton to check the existence of critical pairs (Sampath et al. (1995)), however with exponential complexity in the number of system states. Then Jiang et al. (2001) proposed the twin plant method with polynomial complexity. This is done based on the construction of a non-deterministic automaton before synchronization on observations to search for critical pairs.

The systems that can explicitly present time constraints on event occurrences or legal time during which one can stay in a given state are called RTSs. TA used to model RTSs are largely studied since their introduction by Alur and Dill (1994). Tripakis (2002) proposed for the first time the diagnosability definition of TAs before giving a sufficient and necessary condition by adapting twin plant method and proving the PSPACE-completeness of this checking process. As indicated by the author, the reachability in twin plant for timed automata can be checked in the model-checking tools such as Kronos. However, it is worth noting that the tools such as Kronos and UPPAAL that have implemented the standard forward reachability algorithm based on zones have been shown to be incorrect for diagonal constraints due to the problem of the abstraction operator over zones (Bouyer (2003); Bengtsson and Yi (2003)). There is no such problem when using normal SMT solvers since they do not use abstraction techniques. Then Bouyer et al. (2005) analyzed the diagnosability problem for RTSs by constraining the class of diagnosers considered and demonstrated that it is 2EXPTIME-complete for a deterministic TA diagnoser, by using timed game construction.

On the other hand, some works handle the verification on TA by using SMT techniques with quite good results. In Badban and Lange (2011), a new SMT-based approach is proposed to incrementally analyze TA for some special decidable problems, including universality for deterministic timed automata and language inclusion of a nondeterministic one into a deterministic one. This is done by adopting bounded version for the sake of efficiency. To verify reachability for TA, Kindermann et al. (2012) introduced a SMT-based bounded model checking to handle non-lasso-shaped infinite runs by integrating region abstraction. All these verifications concern only one trajectory in TA. While the diagnosability consists in comparing the observations of two trajectories to search for critical pairs, which makes its verification more complicated. More recently, attention was payed to verification of special failure models, called Failure Propagation Models (FPMs), where failure propagation information is abstracted from the original system model. The approach proposed in Bozzano et al. (2015) presents how to encode in SMT the diagnosability problem for a given timed FPM. It is worth noting that TA are totally different from FPMs, the former being considered as original system models, based on which FPMs can be abstracted. However, this transformation is not trivial at all, as demonstrated in (Priesterjahn et al. (2013); Bittner et al. (2016a,b)). To the best of our knowledge, this is the first work that verifies diagnosability directly on TA by using SMT techniques. We provide an alternative to systems for which the abstraction to a FPM is not convenient.

6. CONCLUSION

We have proposed a new SMT-based approach to verify bounded diagnosability of TA. To do this, both TA and the sufficient and necessary condition of this property, i.e., timed critical pairs, are encoded in SMT as a logic formula

interpreted in linear arithmetic theory whose satisfiability witnesses bounded non-diagnosability. We have not only theoretically proved the correctness of our approach but also demonstrated its feasibility by applying it on different versions of two benchmarks selected from literature.

There are several perspectives for our approach. One ongoing work is to investigate the conditions under which the bounded non-diagnosability implies also non-diagnosability. Another perspective is to extend the current framework to distributed one by considering the communication between the subsystems. Furthermore, it is interesting to study other related properties, such as predictability when taking into account time constraints.

REFERENCES

Alur, R. and Dill, D.L. (1994). A theory of timed automata. *Theor. Comput. Sci.*, 126(2), 183–235.

Badban, B. and Lange, M. (2011). Exact incremental analysis of timed automata with an smt-solver. In *Proceedings of International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS'11)*, volume 6919 of *Lecture Notes in Computer Science*. Springer.

Bengtsson, J. and Yi, W. (2003). On clock difference constraints and termination in reachability analysis of timed automata. In *Proceedings of the 5th International Conference on Formal Engineering Methods (ICFEM'2003)*, volume 2885 of *Lecture Notes in Computer Science*, 491–503. Springer.

Bertrand, N., Fabre, E., Haar, S., Haddad, S., and Hélouët, L. (2014). Active diagnosis for probabilistic systems. In A. Muscholl (ed.), *Proceedings of the 17th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'14)*, volume 8412 of *Lecture Notes in Computer Science*, 29–42. Springer, Grenoble, France.

Bittner, B., Bozzano, M., and Cimatti, A. (2016a). Automated synthesis of timed failure propagation graphs. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, 972–978.

Bittner, B., Bozzano, M., Cimatti, A., and Zampedri, G. (2016b). Automated verification and tightening of failure propagation models. In *Proceedings of the 30th Conference on Artificial Intelligence (AAAI'16)*, 907–913.

Bouyer, P. (2003). Untameable timed automata. In *Proceedings of the Annual Symposium on Theoretical Aspects of Computer Science (STACS'03)*, volume 2607 of *Lecture Notes in Computer Science*, 620–631. Springer.

Bouyer, P., Chevalier, F., and D'Souza, D. (2005). Fault diagnosis using timed automata. In *Proceedings of International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'05)*, Lecture Notes in Computer Science. Springer.

Bozzano, M., Cimatti, A., Gario, M., and Micheli, A. (2015). Smt-based validation of timed failure propagation graphs. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI'15)*, 3724 –3730.

Cataldo, A., Lee, E.A., Liu, X., Matsikoudis, E., and Zheng, H. (2006). A constructive fixed-point theorem and the feedback semantics of timed systems. In *Proceedings of the 8th International Workshop on Discrete Event Systems (WODES'06)*.

Cimatti, A., Pecheur, C., and Cavada, R. (2003). Formal Verification of Diagnosability via Symbolic Model Checking. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, 363–369. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Console, L., Picardi, C., and Dupré, D.T. (2007). A Framework for Decentralized Qualitative Model-based Diagnosis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 286–291. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Debouk, R., Malik, R., and Brandin, B. (2002). A Modular Architecture for Diagnosis of Discrete Event Systems. In *Proceedings of the 41st IEEE Conference on Decision and Control (CDC-02)*, volume 1, 417–422. IEEE.

Germanos, V., Haar, S., Khomenko, V., and Schwoon, S. (2014). Diagnosability under weak fairness. In *Proceedings of the 14th International Conference on Application of Concurrency to System Design (ACSD'14)*. IEEE Computer Society Press, Tunis, Tunisia.

Grastien, A., Anbulagan, J.R., Rintanen, J., and Kelareva, E. (2007). Diagnosis of Discrete-event Systems Using Satisfiability Algorithms. In *Proceedings of the 22th American National Conference on Artificial Intelligence (AAAI-07)*, 305–310. Menlo Park, Calif.: AAAI Press.

Haar, S., Haddad, S., Melliti, T., and Schwoon, S. (2013). Optimal constructions for active diagnosis. In A. Seth and N. Vishnoi (eds.), *Proceedings of the 33rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'13)*, volume 24 of *Leibniz International Proceedings in Informatics*, 527–539. Leibniz-Zentrum für Informatik, Guwahati, India.

Jiang, S., Huang, Z., Chandra, V., and Kumar, R. (2001). A Polynomial Time Algorithm for Testing Diagnosability of Discrete Event Systems. *Transactions on Automatic Control*, 46(8), 1318–1321.

Kindermann, R., Junttila, T., and Niemela, I. (2012). Beyond lassos: Complete smt-based bounded model checking for timed automata. In *Proceedings of Joint FMOODS 2012 and FORTE 2012*, volume 7273 of *Lecture Notes in Computer Science*. Springer.

Pencolé, Y. (2004). Diagnosability Analysis of Distributed Discrete Event Systems. In *Proceedings of the 16th European Conference on Articifial Intelligent (ECAI04)*, 43–47. Nieuwe Hemweg: IOS Press.

Pencolé, Y. and Cordier, M.O. (2005). A Formal Framework for the Decentralised Diagnosis of Large Scale Discrete Event Systems and Its Application to Telecommunication Networks. *Artificial Intelligence*, 164, 121–170.

Priesterjahn, C., Heinzemann, C., and Schafer, W. (2013). From timed automata to timed failure propagation graphs. In *Proceedings of 16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC'13)*.

Reiter, R. (1987). A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32(1), 57–95.

Rintanen, J. (2007). Diagnosers and Diagnosability of Succinct Transition Systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 538–544. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995). Diagnosability of Discrete Event System. *Transactions on Automatic Control*, 40(9), 1555–1575.

Schumann, A. and Huang, J. (2008). A Scalable Jointree Algorithm for Diagnosability. In *Proceedings of the 23rd American National Conference on Artificial Intelligence (AAAI-08)*, 535–540. Menlo Park, Calif.: AAAI Press.

Struss, P. (1997). Fundamentals of Model-based Diagnosis of Dynamic Systems. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, 480–485. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, Inc.

Tripakis, S. (2002). Fault diagnosis for timed automata. In *Proceedings of International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'02)*, Lecture Notes in Computer Science. Springer.

Ye, L. and Dague, P. (2010). Diagnosability Analysis of Discrete Event Systems with Autonomous Components. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-10)*, 105–110. Nieuwe Hemweg: IOS Press.