Verification of Hybrid Automata Diagnosability by Abstraction

Maria D. Di Benedetto, *Fellow, IEEE*, Stefano Di Gennaro, and Alessandro D'Innocenzo

***Abstract*—A notion of diagnosability for hybrid systems is defined, which generalizes the common notion of observability. We propose an abstraction procedure to translate a hybrid automaton into a timed automaton, in order to verify observability and diagnosability properties. We introduce a procedure to check diagnosability, and show that for the system class of our abstraction (namely for a subclass of timed automata: the durational graphs) the verification problem belongs to the complexity class P. We apply our procedure to an electromagnetic valve system for camless engines.**

***Index Terms*—Abstraction, automatic verification, diagnosability, hybrid systems, observability, timed automata.**

I. Introduction

THE increase of functionality offered by today's controllers based on *embedded systems* requires more effort to verify the controlled system, as a malfunction may yield catastrophic results. Since most of the plants of interest have continuous dynamics, the controlled system has a mix of discrete events and continuous dynamics, namely it is a hybrid system [1]. The generality of the hybrid system framework offers flexibility. On the other hand, this generality makes the development of a general analysis theory difficult. When analyzing a hybrid system, the dimension of the state space is often so large that formal verification is out of the question due to its computational complexity. An important technique used to cope with complexity is *abstraction*. By abstraction, we create a system with smaller state space (even finite) that preserves the properties that we want to verify in the original system.

In this paper, we are interested in the automatic verification of the observability and diagnosability properties for hybrid automata. Diagnosability corresponds to failure detection in finite time. Given a plant, a system is diagnosable if, within a finite time bound and only using the observable output of the plant, it is possible to detect that a fault has occurred. We say that a system has a fault if an execution visits a given *faulty* subset of the state space. We discuss in this paper diagnosability of hybrid automata, where the output is given by discrete symbols (possibly unobservable), associated to the discrete transitions. The concept of diagnosability is tightly related to observability: diagnosability generalizes observability. Diagnosability has many applications in several fields, e.g., the detection of an error in an air traffic management procedure [2], [3], of failures in automotive systems [4], in a component of an industrial plant, or in communication systems [5].

Given a plant and a set of faulty states, an important problem often addressed in the literature is to verify automatically whether the system is diagnosable. For the class of discrete event systems (DES), the diagnosability verification problem was treated in several papers by Lin [6], Frank [7], and Lafortune [8]–[10], and the diagnosability verification problem was shown to be polynomial. In these papers, since the concept of time flow is not present in DES, a plant is defined diagnosable if it is possible to detect a failure after a finite number of transitions since the fault has occurred, rather than after a time delay. For the class of timed automata, a definition of δ-diagnosability has been proposed by Tripakis [11]: a plant is δ-diagnosable if it is possible to detect a failure after a time delay bounded by $\delta \in \mathbb{N}$ since the fault has occurred. The diagnosability verification problem for timed automata was demonstrated to belong to PSPACE. Diagnosability of hybrid systems was considered by Fourlas in [4], where a notion of diagnosability was proposed for input–output automata, diagnosability conditions were stated, but no complexity analysis was performed. In [12], a hybrid diagnosis problem was formulated, and qualitative techniques for diagnosis of continuous systems were proposed. In [13], a diagnoser and a mode estimation algorithm for hybrid automata were presented. The diagnosability verification problem for general hybrid systems and its decidability and computational complexity have not been characterized yet by the scientific community.

In this paper, we tackle the diagnosability verification for hybrid automata using an abstraction technique. The first contribution of this paper is a procedure for constructing an abstraction of a hybrid automaton, which belongs to a subclass of timed automata called durational graph. A durational graph is similar to the *durational transition graph* defined in [14]: the main difference is that in a durational transition graph the invariant sets are not defined, and the guards are rectangular sets defined by limits in the set $\mathbb{Q}$ of relative numbers. Durational graphs are more general than discrete event systems (where diagnosability verification is polynomial) and less general than timed automata (where diagnosability verification belongs to PSPACE). Our abstraction procedure is not guaranteed to be polynomial, but we prove in the electromagnetic valve case study that it is polynomial in an interesting and nontrivial

Manuscript received December 07, 2009; revised August 02, 2010; accepted December 31, 2010. Date of publication January 13, 2011; date of current version September 08, 2011. This work was supported in part by the European Commission under Projects iFLY, and IST NoE HYCON contract 511368. Recommended by Associate Editor K. H. Johansson.

The authors are with the Department of Electrical and Information Engineering, Center of Excellence DEWS, University of L'Aquila, L'Aquila 67100, Italy (e-mail: mariadomenica.dibenedetto@univaq.it; stefano.digennaro@univaq.it; adinnoce@seas.upenn.edu; alessandro.dinnocenzo@ing.univaq.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2011.2105738

real case. Thus, our abstraction method allows efficient diagnosability verification of hybrid systems for real cases of interest in engineering applications.

The second and main contribution of the paper is proving that the δ-diagnosability verification problem for durational graphs belongs to the complexity class P, as for DESs. In the literature, the δ-diagnosability verification problem has been defined as follows: "Given a system T and a fixed value δ, verify whether T is δ-diagnosable." The δ-diagnosability verification problem we solve in our paper is more general than the one defined above, and can be defined as follows: "Given a system T, compute the minimum value $\delta_{\min}$ such that T is $\delta_{\min}$-diagnosable." It is in fact extremely important to compute the worst case delay for fault detection. We believe that direct computation of $\delta_{\min}$ in polynomial time is a novel and strong result. To give an example, in Lemma 6 of [11], the minimum value δ for diagnosability of timed automata is computed by trying out different values for δ using a binary search, and verifying δ-diagnosability at each step of the search. This method can be applied to durational graphs, but it is not efficient since it depends on a binary search, while our algorithm computes $\delta_{\min}$ in just one step and in polynomial time with respect to the cardinality of the state space.

Our verification algorithm provides a novel, constructive and nontrivial proof that the computation of $\delta_{\min}$ reduces to a reachability problem over **a set** of durational graphs, constructed in polynomial time starting from the original durational graph. For this reason, tools like KRONOS [15] and UPPAAL [16] (for timed automata) and HyTech [17] (for hybrid systems) cannot be used to compute $\delta_{\min}$, and are not comparable to our algorithm. The advantage of our method is a polynomial algorithm that can be used to compute $\delta_{\min}$ for durational graphs, while existing algorithms and tools are not able to do so. The verification algorithm can be divided in two parts. The first part handles unobservable symbols and represents the core of the procedure: the main issue is dealing with cycles of edges associated with unobservable edges. The second part deals with systems with no unobservable transitions, and makes use of product automata algorithms similarly to [11].

We apply our theoretical results to verify diagnosability of an electromagnetic valve system for camless engines, a device of interest in automotive applications [18]. The observable output we use to perform the diagnosis is only given by the hitting times of the valve with the electromagnets: this corresponds to the use of low-cost sensors.

The paper is organized as follows. In Section II, we define a class of non deterministic hybrid automata, we define hybrid executions, and we introduce notations. In Section III, a definition of diagnosability for hybrid automata is given, which generalizes the notion of discrete state observability in [19]. The main contributions of the paper are as follows.

- In Section IV, we propose a procedure to construct an abstraction of a hybrid automaton, which belongs to a subclass of timed automata called durational graph. We prove that our abstraction preserves diagnosability.
- In Section V, we prove that the diagnosability verification problem for durational graphs belongs to the complexity class P.

In Section VI, we apply our verification procedure to an electromagnetic valve system for camless engines. Concluding remarks are offered in Section VII.

II. Basic Definitions and Notations

In this section, we introduce a class of non deterministic hybrid automata, and define hybrid executions. Then, we introduce a formalism to define the set of executions by means of a formal timed language. Finally, we define a subclass of hybrid automata, the durational graph.

Systems that have both discrete and continuous aspects in their dynamics are called hybrid systems. One prominent theoretical framework that is used to model hybrid systems is proposed in [20], where the discrete part consists of a labeled oriented graph, and the continuous part is described by a dynamical continuous system associated to each discrete state. The interaction between the continuous and discrete part is described by invariant, guard, and reset conditions. We consider here hybrid automata, that are hybrid systems with autonomous dynamics. The observable output is only given by discrete output symbols (possibly unobservable) associated to the discrete transitions, and the delay between the observed output symbols.

Definition 1 (Hybrid Automaton): A hybrid automaton is a tuple $\mathcal{H} = (Q \times X, Q_0 \times X_0, \mathcal{E}, E, \Psi, \eta, Inv, G, R)$ where:

- $Q \times X$ is the hybrid state space, where Q is a finite set of discrete states and $X \subseteq \mathbb{R}^n$ is the continuous state space.
- $Q_0 \times X_0 \subseteq Q \times X$ is the set of initial conditions.
- $\{\mathcal{E}_q\}_{q \in Q}$ associates to each discrete state the autonomous continuous time-invariant dynamics $\mathcal{E}_q : \dot{x} = f_q(x)$. Given an initial condition x_0, we define the solution at time t according to f_q by $x(t) = x_{f_q}(t, x_0)$. The solution is unique with the assumption that f_q is Lipschitz continuous.
- $E \subseteq Q \times Q$ is a collection of edges, where each edge $e \in E$ is an ordered pair of discrete states: the first component is the source and is denoted by $s(e)$, while the second component is the target and is denoted by $t(e)$.
- Ψ is the finite set of discrete output symbols $\{\varepsilon, \psi_1, \psi_2, \ldots, \psi_r\}$, where ε is the unobservable output, that corresponds to the empty string. $\eta : E \to \Psi$ is the output function, that associates to each edge a discrete output symbol.
- $\{Inv_q\}_{q \in Q}$ associates to each discrete state an invariant set $Inv_q \subseteq X$, $\{G_e\}_{e \in E}$ associates to each edge a guard set $G_e \subseteq Inv_{s(e)}$, and $\{R_e\}_{e \in E}$ associates to each edge a reset map $R_e : Inv_{s(e)} \to 2^{Inv_{t(e)}}$.

□

This class of hybrid automata is in general nondeterministic. The continuous state evolves following deterministic dynamics, and the discrete state evolution depends only on the continuous state according to guards, possibly with nondeterministic behaviors in the discrete transitions. We denote $inc(q) \triangleq \{e \in E : t(e) = q\}$ the set of the incoming edges in q, and $out(q) \triangleq \{e \in E : s(e) = q\}$ the set of the outgoing edges from q. We call $e \in E$ an ε-edge if $\eta(e) = \varepsilon$, and use the classical definition $cl_\varepsilon(q)$ of ε-closure of a discrete state q [21], as the set of discrete states that can be reached from q via a path of ε-edges. Notice that $q \in cl_\varepsilon(q)$.

Referring to [20], we recall the definitions of hybrid time basis and hybrid execution of a hybrid system. A *hybrid time basis* $\tau \triangleq \{I_k\}_{k\geq 0}$ is a finite or infinite sequence of intervals $I_k = [t_k, t'_k]$, we refer to [20] for the properties such intervals have to satisfy. The number of intervals is the cardinality $|\tau|$ of the time basis. A *hybrid execution* is a triple $\chi = (\tau, q, x)$, where τ is a hybrid time basis, and q, x describe the evolution of the discrete and continuous state by means of functions $q : \tau \to Q$ piecewise continuous, and $x : \tau \to X$. Functions[1] q, x are defined on the hybrid time basis τ, take values on the hybrid state space, and satisfy the continuous and discrete dynamics and their interactions (invariant, guard and reset). In this paper, we consider *non blocking* hybrid automata, i.e., systems such that all hybrid executions are defined for all time instants. Moreover, we exclude that infinite transitions might occur in a finite time interval. More precisely, we do not allow *Zeno executions*, since they are generally due to a modeling error or to an inadequacy of the model.

We now define the set of executions of the discrete state of a hybrid automaton (and the corresponding observations) by means of formal timed languages. Let $\mathcal{X}$ be the set of all executions $\chi = (\tau, q, x)$ of $\mathcal{H}$. We associate to each execution $\chi \in \mathcal{X}$ a unique timed string $\rho(\chi)$ as a sequence of pairs $\{(q(I_k), t'_k - t_k)\}_{k=0}^{|\tau|}$ with cardinality $|\rho(\chi)| = |\tau|$, where we write without loss of generality $q(I_k) = q_k \in Q$ and $t'_k - t_k = \Delta_k \in \mathbb{R}_+ \cup \{0, \infty\}$. Namely, $\rho(\chi)$ represents an execution of the discrete state of $\mathcal{H}$, where q_k denotes the discrete state in the time interval I_k and Δ_k denotes the dwell time in q_k. For this reason, we will call $\rho(\chi)$ an execution, although it is a timed string. Given an execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|}$, we introduce the following notations:

- $\rho|_i = q_i$ is the discrete state in the time interval I_i of the execution associated to ρ;
- $\rho|_i^j = q_i, \Delta_i, \ldots, q_j, \Delta_j$ is the substring of ρ from index i to j;
- $time(\rho) = \sum_{k=0}^{|\rho|} \Delta_k$ is the time duration of ρ.

Definition 2 (Formal Language of Executions): The timed language of executions of the discrete state of $\mathcal{H}$ is given by

$$\mathcal{L} \triangleq \{\rho(\chi) : \chi \in \mathcal{X}\}.$$

□

Given a subset of discrete states $Q^* \subseteq Q$, we define $\mathcal{L}_{Q^*}$ the language of executions with finite cardinality, such that the last visited discrete state belongs to Q^*:

$$\mathcal{L}_{Q^*} \triangleq \left\{\rho \in \mathcal{L} : |\rho| < \infty, \rho|_{|\rho|} \in Q^*\right\}.$$

Given an execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|}$, we define the associated output string as

$$\Delta_0, \eta((q_0, q_1)), \Delta_1, \eta((q_1, q_2)), \Delta_2, \ldots.$$

The associated *observation* $P(\rho)$ is obtained from the output by erasing all ε (unobservable) symbols and by adding up the adjacent time delays. For instance, an output string 3, ψ_1, 4, ε, 5, ψ_2, 2 is observed as 3, ψ_1, 9, ψ_2, 2.

[1]We abuse notation by using the same symbol q or x for an element of Q or X, and here to denote a function.

A *timed automaton* [22] is a hybrid automaton where the dynamics of the continuous variables have constant slope 1 for each discrete location (each variable is a clock), the initial continuous state is a singleton set for each discrete location, the guards are rectangular sets,[2] and the reset map for each variable is either the identity or zero. We call *durational graph* a timed automaton characterized by only one clock that is reset to 0 for all edges.

Definition 3 (Durational Graph): A durational graph is a hybrid automaton $(Q \times X, Q_0 \times X_0, \mathcal{E}, E, \Psi, \eta, Inv, G, R)$ such that:

- $X = \mathbb{R}_+ \cup \{0\}$ is the continuous state space;
- for each $q_0 \in Q_0$, the initial condition is given by $(q_0, 0)$;
- for each $q \in Q$, the continuous dynamics are defined by $\mathcal{E}_q : \dot{x} = 1$ and the set Inv_q is a rectangular set;
- for each $e \in E$, the set G_e is a rectangular set and $R_e(x) = \{0\}$.

□

This definition implies that a durational graph is uniquely identified by a tuple $\mathcal{G} = (Q, Q_0, E, \Psi, \eta, Inv, G)$. Definitions of formal languages of executions and observations also hold for durational graphs. The notations used in this paper are summarized in Table II.

III. Diagnosability Definition

Given a hybrid automaton $\mathcal{H}$, let $Q_c \subset Q$ be a set of discrete states that model a failure in $\mathcal{H}$: Q_c is called *faulty set*. A δ-faulty execution is a trajectory that enters the faulty set at a certain time instant, and then continues flowing for a time duration δ.

Definition 4 (δ-Faulty Execution): An execution $\rho \in \mathcal{L}$ is δ-faulty if there exists a finite index $k_c \geq 0, k_c \leq |\rho|$ if $|\rho| < \infty$, such that

$$(1)\, \forall k < k_c,\ \rho|_k \notin Q_c; (2)\ \rho|_{k_c} \in Q_c; (3)\ time\left(\rho|_{k_c}^{|\rho|}\right) = \delta.$$

□

For any faulty execution ρ, we use the notation $\rho|_{k_c}$ to denote the first faulty state visited by ρ. We define $\mathcal{F}_\delta$ the set of all δ-faulty executions, and $\mathcal{F} = \bigcup_{\delta \geq 0} \mathcal{F}_\delta \subseteq \mathcal{L}$ the set of all faulty executions. We say that a set Q_c is δ-diagnosable for a system $\mathcal{H}$ if it is possible to detect within a delay upper bounded by δ whether an execution has visited the faulty set, only using the observable output. More precisely:

Definition 5: A set Q_c is δ-diagnosable for $\mathcal{H}$ if and only if

$$\forall \rho \in \bigcup_{\delta^* \geq \delta} \mathcal{F}_{\delta^*},\ \forall \rho' \in \mathcal{L} \setminus \bigcup_{\delta^* \geq \delta} \mathcal{F}_{\delta^*},\ P(\rho) \neq P(\rho').$$

The notion of δ-diagnosability is more general than discrete state observability as defined in [19].

Definition 6: [19] A set Q_c is observable for $\mathcal{H}$ if and only if

$$\forall \rho \in \mathcal{L}_{Q_c},\ \forall \rho' \in \mathcal{L}_{Q \setminus Q_c},\ P(\rho) \neq P(\rho')$$

□

[2]A rectangular set in $\mathbb{R}^n$ is any subset that can be defined by a finite union of cartesian products of intervals.

Proposition 1: Q_c is observable if and only if Q_c is 0-diagnosable.

Proof: By Definition 5 and the fact that $\mathcal{F}_0 \subseteq \mathcal{L}_{Q_c}$, the result follows. ∎

As a consequence, the results given in this paper also apply to observability of the discrete state.

Remark 1: If a faulty set Q_c is δ-diagnosable, it is possible to detect that the faulty set has been visited, but it is not possible to distinguish which faulty state has been visited. To determine which faulty state has been visited, we need that for each $q_c \in Q_c$ the singleton set $\{q_c\}$ is δ-diagnosable. □

If a system is diagnosable for some finite δ, the following property shows that it is very interesting to compute the minimum value δ_m for which $\mathcal{H}$ is δ_m-diagnosable.

Proposition 2: Given $\mathcal{H}$, the following statements hold:

1) if Q_c is δ-diagnosable, then it is δ^*-diagnosable for all $\delta^* \geq \delta$;
2) if Q_c is not δ-diagnosable, then it is not δ^*-diagnosable for all $\delta^* \leq \delta$.

Proof: Straightforward by Definition 5. ∎

IV. Translating Hybrid Automata Into Timed Automata

In this section, we propose an abstraction procedure that can be used to verify δ-diagnosability of hybrid automata. This procedure can also be useful to verify more general properties that are not easy to check or are even undecidable for the general hybrid model, e.g., temporal properties [23]. The abstracting system is a durational graph as defined in Section II. We propose an algorithm to construct a durational graph $\mathcal{G}$ from a given hybrid automaton $\mathcal{H}$, and show that $\mathcal{G}$ preserves diagnosability properties.

Consider a hybrid automaton $\mathcal{H}$. We define the following sets.

- $X_0(q_0) \triangleq \{x_0 \in X_0 : (x_0, q_0) \in X_0 \times Q_0\}$ is the set of initial continuous conditions associated to the initial discrete state $q_0 \in Q_0$. The non blocking assumption implies that $X_0(q_0) \subseteq Inv_{q_0}$.
- $Range(R_e) \triangleq \{x \in X : \exists x' \in G_e, x \in R_e(x')\}$ is the range of the reset associated to edge $e \in E$. The non blocking assumption implies that $\forall q \in Q$, $\forall e \in inc(q)$, $Range(R_e) \subseteq Inv_q$: i.e., the reset "lands" in the invariant of the target location.

We construct $\mathcal{G}$, that is an abstraction of $\mathcal{H}$, as follows. Define a relation $\gamma \subseteq Q \times (Q \times (E \cup Q_0))$, where $l \in E \cup Q_0$

$$\gamma \triangleq \{(q,(q,l)) \text{ such that } q \in Q, \text{ and either } l \in inc(q) \text{or } l = q \in Q_0\}.$$

Namely, we relate to each state $q \in Q$ of $\mathcal{H}$ one state of $\mathcal{G}$ for each incoming edge $l \in inc(q)$, and an additional state if $q \in Q_0$, as illustrated in Fig. 1. In the following, we use the notation

$$\Re_l \triangleq \begin{cases} X_0(q_0), & \text{if } l \in Q_0 \\ Range(R_e), & \text{if } l \in E \end{cases}$$

Algorithm 1: Let a hybrid automaton $\mathcal{H} = (Q \times X, Q_0 \times X_0, \mathcal{E}, E, \Psi, \eta, Inv, G, R)$ be given. We define a durational graph $\mathcal{G} = (Q', Q'_0, E', \Psi', \eta', Inv', G')$ as follows:

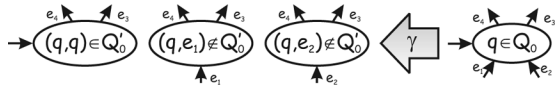

Fig. 1. Split induced by the relation γ. The horizontal arrows indicate an initial state.

1) $Q' \triangleq \{(q,l) : (q,(q,l)) \in \gamma\}$;
2) $Q'_0 \triangleq \{(q,q) \in Q' : q \in Q_0\}$;
3) $E' \triangleq \{((q_1,l_1),(q_2,l_2)) : l_2 = (q_1,q_2)$, and either $l_1 \in inc(q_1)$ or $l_1 = q_1 \in Q_0\}$;
4) $\forall e' = ((q_1,l_1),(q_2,l_2)) \in E'$, define $\eta'(e') \triangleq \eta(l_2)$;
5) $\forall q' = (q,l) \in Q'$, define

$$Inv'_{q'} \triangleq \{t \in \mathbb{R}_+ \cup \{0,\infty\} : \exists x_0 \in \Re_l, x_{f_q}(\tau, x_0) \in Inv_q, \forall \tau \in [0,t]\}. \quad (1)$$

The set $Inv'_{q'}$ consists of all time instants t such that there exists an execution of the continuous state, according to the dynamics f_q and with initial condition in $\Re_l$ at time 0, that remains in the invariant set Inv_q during the time interval $[0,t]$. Notice that the origin $t = 0$ always belongs to $Inv'_{q'}$, since we have assumed that $\forall q \in Q, \forall e \in inc(q)$, $Range(R_e) \subseteq Inv_q$, and $\forall q \in Q_0$, $X_0(q) \subseteq Inv_q$.

6) $\forall e' = ((q_1,l_1),(q_2,l_2)) \in E'$, define

$$G'_{e'} \triangleq \{t \in \mathbb{R}_+ \cup \{0,\infty\} : \exists x_0 \in \Re_{l_1}, x_{f_{q_1}}(t, x_0) \in G_{l_2}\}. \quad (2)$$

The set $G'_{e'}$ consists of all time instants t such that there exists an execution of the continuous state, according to the dynamics f_{q_1} and with initial condition in $\Re_{l_1}$ at time 0, that enables the transition l_2 at time t.

□

The intuition behind the algorithm is the following: we split each discrete state according to the relation γ, depending on the number of incoming edges and initial conditions (Fig. 1). This split ensures that any discrete state of $\mathcal{G}$ has only one incoming edge. The main issue is the computation of the invariant (1) and guard sets (2) by means of dwell time of the hybrid automaton in each discrete state. If the continuous dynamics are linear, the exact computation is possible when the system has a particular structure [24], [25]. In general, we can compute an over approximation of

$$Inv'_{q'} \supseteq \{t \in \mathbb{R}_+ \cup \{0,\infty\} : \exists x_0 \in \Re_l \; x_{f_q}(\tau, x_0) \in Inv_q, \forall \tau \in [0,t]\} \quad (3)$$

$$G'_{e'} \supseteq \{t \in \mathbb{R}_+ \cup \{0\} : \exists x_0 \in \Re_{l_1}, x_{f_{q_1}}(t, x_0) \in G_{l_2}\} \quad (4)$$

using approaches that can be found in the literature. The literature on computational approaches to reach set computation for hybrid systems is quite rich. If the dynamics are linear but the computation of the reach set cannot be solved in closed form, one can obtain approximations of the reach set by resorting to a number of approaches. The authors in [26] propose a procedure for automatic verification of safety properties of hybrid systems with linear continuous dynamics and uncertain bounded input. The procedure proposed by [27] works for high dimensional continuous state spaces, but it is not possible to quantify the

over-approximation error. [28] introduces a procedure to compute a sequence of polytopes (a *flow pipe*), that are over approximations of the reach sets within specific time intervals. By refining these intervals it is possible to determine approximations with arbitrary precision. The weak point of this approach is the increase in the computation time. Another procedure similar to [28] is presented in [29], where an algorithm to compute a sequence of zonotopes[3] is proposed. This algorithm is attractive, because the computation of the *flow pipe* is considerably faster for zonotopes than for polytopes [30]. Furthermore, the algorithm can be extended to compute the reach set of systems with a bounded control input. For the same class of models, [31] proposes a formal approach to compute over-approximating reachable sets with ellipsoidal shapes. Related results are [32]–[34]. If the dynamics are non linear, one can use the approach developed in [28]. This intensive research has given birth to several verification tools for reachability analysis on hybrid systems, such as *d/dt*, *MATISSE*, *CheckMate*, the *Ellipsoidal Toolbox*, and many others. For a more exhaustive review, the reader is directed to [35]. In particular, we will use the algorithms we proposed in [36], [37] to apply our abstraction procedure to the case study of Section VI.

We now focus on the properties of $\mathcal{G}$. In the following, if not clear from the context, we will use a superscript to refer to system $\mathcal{H}$ or to system $\mathcal{G}$. We show now that the behavior of $\mathcal{G}$ embeds the behavior of $\mathcal{H}$.

Proposition 3: Given $\mathcal{H}$ and $\mathcal{G}$, for each execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|} \in \mathcal{L}^{\mathcal{H}}$, there exists an execution $\rho' = \{(q'_k, \Delta'_k)\}_{k=0}^{|\rho'|} \in \mathcal{L}^{\mathcal{G}}$ such that $|\rho| = |\rho'|$, $(q_k, q'_k) \in \gamma$, $\Delta_k = \Delta'_k$, $\forall k = 0, \ldots, |\rho|$, and $\eta((q_{k-1}, q_k)) = \eta\left((q'_{k-1}, q'_k)\right)$, $\forall k = 1, \ldots, |\rho|$.

Proof: Consider the execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|} \in \mathcal{L}^{\mathcal{H}}$. Let $q'_0 = (q_0, q_0) \in Q'_0$: by construction of $\mathcal{G}$, $(q_0, q'_0) \in \gamma$. Let $e_1 = (q_0, q_1) \in E$: there exists $q'_1 = (q_1, e_1)$ with $(q_1, q'_1) \in \gamma$, such that $e'_1 = (q'_0, q'_1)$ and $\eta'(e'_0) = \eta(e_0)$. The guard G'_{e_1} computed as an over approximation (4), and the invariant $Inv'_{q'_0}$ computed as an over approximation (3), imply that $\Delta_0 \in G'_{e_1} \cap Inv'_{q'_0}$, and $[0, \Delta_0] \subseteq Inv'_{q'_0}$. By iteration, we construct an execution $\rho' = \{(q'_k, \Delta'_k)\}_{k=0}^{|\rho'|} \in \mathcal{L}^{\mathcal{G}}$ such that $|\rho| = |\rho'|$, $(q_k, q'_k) \in \gamma$, $\Delta_k = \Delta'_k$, $\forall k = 0, \ldots, |\rho|$, and $\eta((q_{k-1}, q_k)) = \eta\left((q'_{k-1}, q'_k)\right)$, $\forall k = 1, \ldots, |\rho|$. ■

Consider now the following assumption.

Assumption 1: Suppose that the guard and invariant sets of $\mathcal{G}$ are computed exactly, and the reset functions of the system $\mathcal{H}$ are *memoryless* (the system "forgets" its continuous state when a transition occurs):

$$\forall e \in E, \forall x \in G_e, R_e(x) = Range(R_e).$$

□

If Assumption 1 holds, not only the behavior of $\mathcal{G}$ embeds the behavior of $\mathcal{H}$ but also vice versa.

Proposition 4: Given $\mathcal{H}$ and $\mathcal{G}$, let Assumption 1 hold. Then for each execution $\rho' = \{(q'_k, \Delta'_k)\}_{k=0}^{|\rho'|} \in \mathcal{L}^{\mathcal{G}}$, there exists an execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|} \in \mathcal{L}^{\mathcal{H}}$ such that $|\rho| = |\rho'|$, $(q_k, q'_k) \in \gamma$, $\Delta_k = \Delta'_k$, $\forall k = 0, \ldots, |\rho|$, and $\eta((q_{k-1}, q_k)) = \eta\left((q'_{k-1}, q'_k)\right)$, $\forall k = 1, \ldots, |\rho|$.

Proof: Consider the execution $\rho' = \{(q'_k, \Delta'_k)\}_{k=0}^{|\rho'|} \in \mathcal{L}^{\mathcal{G}}$. By construction of $\mathcal{G}$, $q'_0 = (q_0, q_0)$ for some $q_0 \in Q_0$, with $(q_0, q'_0) \in \gamma$. Moreover, $q'_1 = (q_1, e)$ for some q_1 successor of q_0 and $e = (q_0, q_1)$, with $(q_1, q'_1) \in \gamma$. The guard G'_{e_1} computed exactly (2), and the invariant $Inv'_{q'_0}$ computed exactly (1), imply that there exists $x_0 \in X_0(q_0) : x_{f_{q_0}}(\Delta'_0, x_0) \in G_e$ and $x_{f_{q_0}}(\tau, x_0) \in Inv_{q_0}$, $\forall \tau \in [0, \Delta'_0]$. Under the assumption that the reset functions are memoryless, we construct by iteration an execution $\rho = \{(q_k, \Delta_k)\}_{k=0}^{|\rho|} \in \mathcal{L}^{\mathcal{H}}$ such that $|\rho| = |\rho'|$, $(q_k, q'_k) \in \gamma$, $\Delta_k = \Delta'_k$, $\forall k = 0, \ldots, |\rho|$, and $\eta((q_{k-1}, q_k)) = \eta\left((q'_{k-1}, q'_k)\right)$, $\forall k = 1, \ldots, |\rho|$. ■

We show now that our abstraction preserves diagnosability (and thus observability). We recall that Algorithm 1 takes as input a hybrid automaton $\mathcal{H}$, and produces as output a durational graph $\mathcal{G}$ and a relation $\gamma \subseteq Q^{\mathcal{H}} \times Q^{\mathcal{G}}$. Given $\mathcal{H}$, $\mathcal{G}$, and a faulty set $Q_c^{\mathcal{H}}$, define:

$$Q_c^{\mathcal{G}} \triangleq \bigcup_{q \in Q_c^{\mathcal{H}}} \{q' \in Q^{\mathcal{G}} : (q, q') \in \gamma\}.$$

Proposition 5: Given $\mathcal{H}$ and $\mathcal{G}$, a faulty set $Q_c^{\mathcal{H}}$ is δ-diagnosable for $\mathcal{H}$ **if** $Q_c^{\mathcal{G}}$ is δ-diagnosable for $\mathcal{G}$.

Proof: By contradiction, assume that the hypothesis is true and that $Q_c^{\mathcal{H}}$ is not δ-diagnosable for $\mathcal{H}$. This implies by Definition 5 that there exists $\delta^* \geq \delta$, $\rho \in \mathcal{F}_{\delta^*}^{\mathcal{H}}$, and $\rho' \in \mathcal{L}^{\mathcal{H}} \setminus \mathcal{F}^{\mathcal{H}}$ such that $P(\rho) = P(\rho')$. Let $\rho|_{k_c}$ be the first faulty state of ρ. By Proposition 3, there exist $\tilde{\rho}, \tilde{\rho}' \in \mathcal{L}^{\mathcal{G}}$ such that $P(\rho) = P(\tilde{\rho})$ and $P(\rho') = P(\tilde{\rho}')$. Furthermore, $\rho|_{k_c} \in Q_c^{\mathcal{G}}$ and $(\rho|_{k_c}, \tilde{\rho}|_{k_c}) \in \gamma$, thus $\tilde{\rho} \in \mathcal{F}_{\delta^*}^{\mathcal{G}}$. For the same reasoning, it is clear that $\tilde{\rho}' \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$. This implies that $Q_c^{\mathcal{G}}$ is not δ-diagnosable for $\mathcal{G}$, that is a contradiction. ■

When Assumption 1 holds, Proposition 4 holds. As a consequence, Proposition 5 becomes a necessary and sufficient condition:

Proposition 6: Given $\mathcal{H}$ and $\mathcal{G}$, let Assumption 1 hold. Then a faulty set $Q_c^{\mathcal{H}}$ is δ-diagnosable for $\mathcal{H}$ **if and only if** $Q_c^{\mathcal{G}}$ is δ-diagnosable for $\mathcal{G}$. ■

Proof: Straightforward inverting the reasoning in the proof of Proposition 5. ■

V. Diagnosability Verification

In this section, we focus on the diagnosability problem for durational graphs. We propose a verification procedure and determine its computational complexity. The verification algorithm consists of two parts. In the first part, the tricky one, we deal with edges associated to an unobservable output symbol: we propose an algorithm to construct a durational graph without unobservable outputs, which preserves diagnosability. In the second part, we propose a verification algorithm for systems that do not generate unobservable outputs.

Removal of ε-edges has been discussed for discrete event systems [38] to study observability properties, and for timed automata [39] to preserve simulation relations. We proposed in [19] a procedure to erase ε-edges from a durational graph, while preserving observability properties. However, this procedure does not preserve diagnosability since, if a faulty state q_c

[3]A zonotope is a centrally symmetric polytope, defined by the Minkowski sum of its line segment generators $s_1, \ldots, s_n \in \mathbb{R}^m$.

has at least one outgoing or incoming ε-edge, such state is not observable, while it might be δ-diagnosable for some $\delta > 0$. For this reason, in order to preserve δ-diagnosability, it is necessary to be careful when erasing ε-edges that have a faulty state as endpoint.

Let a durational graph $\mathcal{G} = (Q, \{q_s\}, E, \Psi, \eta, Inv, G)$ and a destination discrete state $q_d \in Q$ be given. Define the set $\Lambda(q_d)$ of all time instants t such that there exists an execution of $\mathcal{G}$ with time duration t that terminates in q_d:

$$\Lambda(q_d) \triangleq \left\{ t \in \mathbb{R}_+ \cup \{0\} : \exists \rho \in \mathcal{L}^{\mathcal{G}}_{\{q_d\}}, time(\rho) = t \right\}.$$

We also write $\{\infty\} \in \Lambda(q_d)$ if there exists an execution $\rho \in \mathcal{L}^{\mathcal{G}}$ such that $time(\rho) = \infty$ and $\forall i \geq 0$, $\rho|_i \neq q_d$.[4]

Proposition 7: Given a durational graph $\mathcal{G}$ and a discrete state $q_d \in Q$, then $\Lambda(q_d)$ can be computed in polynomial time. Moreover, $\Lambda(q_d)$ is a rectangular set if and only if all paths of $\mathcal{G}$ that connect q_s to q_d do not contain cycles, or there exists at least one path with cycles where at least one edge is associated to a non-singleton rectangular interval.

Proof: Given $\mathcal{G}$ and q_d, consider a *non deterministic finite automaton*[5] $\mathcal{N}$ with set of states Q, initial state $\{q_s\}$, and final state $\{q_d\}$. The alphabet of $\mathcal{N}$ is defined as a finite set of rectangular intervals $\Sigma \triangleq \{\sigma_e : e \in E, \sigma_e = G_e \cap Inv_{s(e)}\}$. Namely, it is the collection of all the guards G_e intersected with the corresponding invariant $Inv_{s(e)}$. The transition relation $\delta : Q \times \Sigma \to 2^Q$ of $\mathcal{N}$ is defined as $\delta(q, \sigma_e) = \{q' \in Q : e = (q, q')\}$.

It is easy to show that $\Lambda(q_d)$ can be computed from the regular expression associated to $\mathcal{N}$, by applying the following three rules, where $\sigma, \sigma' \in \Sigma$. Equation (1): Replace *alternation* $\sigma \mid \sigma'$ by $\sigma \cup \sigma'$ (finite union of rectangular intervals). That is, the alternation of symbols corresponds to the union of the crossing times. Equation (2): Replace *concatenation* $\sigma \cdot \sigma'$ by $\sigma + \sigma'$ (finite sum of rectangular intervals). That is, the concatenation of symbols corresponds to the sum of the crossing times. Equation (3): Replace *Kleene star* σ^* by $\bigcup_{n \geq 0} n\sigma$ (infinite union of rectangular intervals). That is, the repetition of symbols corresponds to the union of n crossing times of a cycle, for all $n \geq 0$.

We need to prove that $\Lambda(q_d)$ is a rectangular interval if the hypothesis of this proposition holds. Clearly, elements 1 and 2 of the list always generate rectangular intervals, since a finite union or sum of rectangular intervals is a rectangular interval, and each element of Σ is a rectangular interval. We prove now that element 3 generates a rectangular interval if and only if σ^* is not generated by all singleton intervals, that is equivalent to the hypothesis of this proposition. Let $\sigma = [t_1, t'_1] \cup \cdots \cup [t_r, t'_r], r \in \mathbb{N}$, then

$$\sigma^* = \bigcup_{n \geq 0} n\sigma = \bigcup_{n \geq 0} \left(n[t_1, t'_1] \cup \cdots \cup n[t_r, t'_r] \right).$$

We first prove that there exists a finite value $N_i \in \mathbb{N}$ for each $i \in \{1, \ldots, r\}$, such that the following holds:

$$\forall n \geq N_i, nt_i < (n+1)t_i \leq nt'_i < (n+1)t'_i.$$

The strict inequalities are clearly true for any value of i, n. Since $\forall i \in \{1, \ldots, r\}$, $t'_i \geq t_i$, then

$$(n+1)t_i \leq nt'_i \Longleftrightarrow n \geq \frac{t_i}{t'_i - t_i}.$$

Let $N_i \triangleq \{t_i/t'_i - t_i\} \in \mathbb{N}$, $\forall i \in \{1, \ldots, r\}$: define $t_\infty \in \mathbb{R}_+ \cup \{0, \infty\}$ by $t_\infty \triangleq \min_i \{N_i t_i\}$. It is clear that t_∞ is finite if and only if $\exists i : t'_i > t_i$. It is also clear that for $n \geq N_k, k = argmin\{N_i t_i\}$, all consecutive time intervals $n[t_k, t'_k]$ and $(n+1)[t_k, t'_k]$ overlap; thus, their infinite union $\bigcup_{n \geq N_k} n[t_k, t'_k]$ generates the set $[t_\infty, +\infty)$. Let N_∞ be the minimum value of n such that $nt_i \geq t_\infty, \forall i \in \{1, \ldots, r\}$. Thus, σ^* can be defined by a finite union of rectangular intervals:

$$\sigma^* = \left\{ \bigcup_{n=0}^{N_\infty} n\sigma \right\} \cup [t_\infty, +\infty).$$

■

The following algorithm takes as input a durational graph $\tilde{\mathcal{G}}$ and a faulty set $\tilde{Q}_c$ of $\tilde{\mathcal{G}}$, and produces as output a durational graph $\mathcal{G}$ without ε-edges, a faulty set Q_c of $\mathcal{G}$, and a *faulty function* $\lambda : Q_c \to 2^{\mathbb{R}_+ \cup \{0,\infty\}}$, that associates to each faulty state of $\mathcal{G}$ a rectangular time interval.

Algorithm 2: Given a durational graph $\tilde{\mathcal{G}} = (\tilde{Q}, \tilde{Q}_0, \tilde{E}, \tilde{\Psi}, \tilde{\eta}, \widetilde{Inv}, \tilde{G})$ and a faulty set $\tilde{Q}_c$, initialize $\mathcal{G} \triangleq \tilde{\mathcal{G}}$, $\Psi \triangleq \tilde{\Psi} \setminus \{\varepsilon\}$, $Q_c \triangleq \tilde{Q}_c$, and $\lambda(q) \triangleq \{0\}, \forall q \in Q_c$.

For each unvisited state $q_s \in Q$ such that $\exists e \in inc(q_s), \eta(e) \neq \varepsilon$, and for each $q_d \in Q, \psi \in \Psi$ such that $\exists q^* \in cl_\varepsilon(q_s) \setminus \{q_s\}, \exists e = (q^*, q_d) \in E, \eta(e) = \psi$, define $Q^*_c \triangleq cl_\varepsilon(q_s) \setminus \{q_s\} \cap Q_c$ and proceed as follows.

1) If $Q^*_c \neq \emptyset$, then create a new state q^c_s to Q and Q_c while keeping incoming and outgoing edges of q_s.[6] For each $q_c \in Q^*_c$, compute $\Lambda_1(q_c)$ from the durational graph $\mathcal{G}_1$ induced[7] on $\mathcal{G}$ by the set of states $cl_\varepsilon(q_s)$ and the set of edges $\{e \in E : \eta(e) = \varepsilon\}$, with initial state q_s. Compute $\Lambda_2(q_d)$ from the durational graph $\mathcal{G}_2$ induced on $\mathcal{G}$ by the set of states $cl_\varepsilon(q_s) \cup \{q_d\}$ and the set of edges $\{e \in E : \eta(e) = \varepsilon \text{or } s(e) \in cl_\varepsilon(q_s) \setminus \{q_s\}, t(e) = q_d, \eta(e) = \psi\}$, with initial state q_c. Set $\lambda(q^c_s) \triangleq \lambda(q_s) \cup \bigcup_{q_c \in Q^*_c} \Lambda_1(q_c)$. If $e = (q^c_s, q_d) \notin E$, then add e to E and set $\eta(e) \triangleq \psi, G_e \triangleq \emptyset$. Set $G_e \triangleq G_e \cup \bigcup_{q_c \in Q^*_c} (\Lambda_1(q_c) + \Lambda_2(q_d))$. Mark q^c_s as visited.
2) Compute $\Lambda_3(q_d)$ from the durational graph $\mathcal{G}_3$ induced on $\mathcal{G}$ by the set of states $(cl_\varepsilon(q_s) \setminus Q^*_c) \cup \{q_d\}$ and the set of edges $\{e \in E : \eta(e) = \varepsilon \text{or } s(e) \in cl_\varepsilon(q_s) \setminus \{q_s\}, t(e) = q_d, \eta(e) = \psi\}$, with initial state q_s. If $e = (q_s, q_d) \notin E$, then add e to E and set $\eta(e) \triangleq \psi, G_e \triangleq \emptyset$. Set $G_e \triangleq G_e \cup \Lambda_3(q_d)$. Mark q_s as visited.

Finally, set $Inv(q) \triangleq [0, \max_{e \in out(q)} \{\sup\{G_e\}\}]$ for each $q \in Q$, and erase all states whose incoming edges are all ε-edges, then erase all hanging and unobservable edges. □

[4]This happens when there exists an execution that never reaches q_d.

[5]For the classical definition of non-deterministic finite automata and regular expressions, we refer to [21].

[6]Namely we duplicate the state q^c_s.

[7]The durational graph induced on a durational graph $\mathcal{G}$ by the set of states X and the set of edges Y, is a durational graph where the set of states is X, the set of edges is Y, and outputs, guards, invariants and resets are the same of $\mathcal{G}$ restricted to the sets X, Y

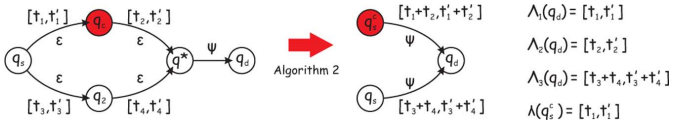

Fig. 2. Example of the split procedure operated by Algorithm 2.

The idea of the algorithm is to preserve all executions merging the states connected by ε-edges. The main issue is that it is sometimes needed to merge together faulty and not faulty states: in this case, as illustrated in Fig. 2, we consider all paths that visit the faulty states, and create a new faulty state q_s^c. We set $\lambda(q_s^c)$ with the set of all time instants, such that a fault can occur starting from q_s and generating only unobservable outputs. We set the guards of the outgoing edges from q_s as the set of time instants such that an observable output is generated, without visiting a faulty state; and we set the guards of the outgoing edges from q_s^c as the set of time instants such that an observable output is generated, visiting a faulty state. We assume that the guards of $\mathcal{G}$ are not singleton sets;[8] thus, Proposition 7 implies that the algorithm above constructs a durational graph, since every edge is associated to a guard that is a rectangular set. This is true also if there exist cycles of unobservable edges. In the following result, we formalize some properties that hold by construction of Algorithm 2.

Proposition 8: Given $\tilde{\mathcal{G}}$ and $\mathcal{G}$:

1) For each execution $\tilde{\rho} \in \mathcal{F}_{\delta}^{\tilde{\mathcal{G}}}$ and for each $\delta \geq 0$, there exist an execution $\rho \in \mathcal{F}_{\delta+\lambda^*}^{\mathcal{G}}$ and $\lambda^* \in \lambda(\rho|_{k_c})$, such that $P(\tilde{\rho}) = P(\rho)$.
2) For each execution $\tilde{\rho} \in \mathcal{L}^{\tilde{\mathcal{G}}} \setminus \mathcal{F}^{\tilde{\mathcal{G}}}$, there exists an execution $\rho \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$ such that $P(\tilde{\rho}) = P(\rho)$.
3) For each execution $\rho \in \mathcal{F}_{\delta}^{\mathcal{G}}$, for each $\lambda^* \in \lambda(\rho|_{k_c})$ and for each $\delta \geq \lambda^*$, there exists an execution $\tilde{\rho} \in \mathcal{F}_{\delta-\lambda^*}^{\tilde{\mathcal{G}}}$ such that $P(\rho) = P(\tilde{\rho})$.
4) For each execution $\rho \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$, there exists an execution $\rho \in \mathcal{L}^{\tilde{\mathcal{G}}} \setminus \mathcal{F}^{\tilde{\mathcal{G}}}$ such that $P(\rho) = P(\tilde{\rho})$.

We define now the second part of the algorithm, and assume absence of ε-edges. The idea for diagnosability verification is to construct a finite set of durational graphs that embed diagnosability properties of $\mathcal{G}$. We show how to check if there exists a (minimum) finite δ such that the starting system $\tilde{\mathcal{G}}$ (with ε-edges) is δ-diagnosable, by stating conditions on the structure of the constructed durational graphs. Given $\mathcal{G}$, we first define a constructive procedure of a product durational graph $\mathcal{C}_{q_0',q_0''}$ for each pair $(q_0', q_0'') \in Q_0 \times Q_0$. The following algorithm is similar to the classical product automaton construction: the difference is that we stop the exploration when we discover a faulty state in the first component, and we do not explore pairs of states with a faulty state in the second component.

Algorithm 3: Given durational graphs $\mathcal{G}_{q_0'} = (Q, \{q_0'\}, E, \Psi, \eta, G, Inv)$ and $\mathcal{G}_{q_0''} = (Q, \{q_0''\}, E, \Psi, \eta, G, Inv)$, construct the product durational graph $\mathcal{C}_{q_0',q_0''} = (\tilde{Q} \subseteq Q \times Q, \{\tilde{q}_0\}, \tilde{E}, \Psi, \tilde{\eta}, \tilde{G}, \widetilde{Inv})$ as follows:

Initialize $\tilde{Q} \triangleq \tilde{q}_0 \triangleq (q_0', q_0'')$, $\widetilde{Inv}((q_0', q_0'')) \triangleq Inv_{q_0'} \cap Inv_{q_0''}$, $\tilde{E} \triangleq \emptyset$;

For each unvisited state $(q', q'') \in \tilde{Q}$, do:

2.1) For each $e', e'' \in E : e' = (q', \bar{q}')$, $e'' = (q'', \bar{q}'')$, $\eta(e') = \eta(e'') = \psi \in \Psi$, $\bar{q}'' \notin Q_c$, set $\tilde{Q} \triangleq \tilde{Q} \cup (\bar{q}', \bar{q}'')$, $\widetilde{Inv}((\bar{q}', \bar{q}'')) \triangleq Inv_{\bar{q}'} \cap Inv_{\bar{q}''}$, $e \triangleq ((q', q''), (\bar{q}', \bar{q}''))$, $\tilde{E} \triangleq \tilde{E} \cup \tilde{e}$, $\tilde{\eta}(\tilde{e}) \triangleq \psi$, $\tilde{G}_{\tilde{e}} \triangleq G_{e'} \cap G_{e''}$. Mark (q', q'') as visited;

2.2) If $\bar{q}' \in Q_c$, then mark $(\bar{q}', \bar{q}'')$ as visited. □

Given $\mathcal{G}_{q_0'}$, $\mathcal{G}_{q_0''}$, $\mathcal{C}_{q_0',q_0''}$, and any pair (q_c, q) in the state space $\tilde{Q}$ of $\mathcal{C}_{q_0',q_0''}$, then there exist two executions ρ' of $\mathcal{G}_{q_0'}$ and ρ'' of $\mathcal{G}_{q_0''}$ with the same timed observation, where the last visited state of ρ' is q_c, while the last visited state of ρ'' is q. This property is a direct consequence of the classical product automaton construction, and can be formalized by the following proposition.

Proposition 9: Given $\mathcal{G}$ and $\mathcal{C}_{q_0',q_0''}$:

1) For each pair of executions $\rho' = \{(q_k', \Delta_k')\}_{k=1}^{|\rho'|} \in \mathcal{F}_0^{\mathcal{G}}$, $\rho'' = \{(q_k'', \Delta_k'')\}_{k=1}^{|\rho''|} \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$ with $P(\rho') = P(\rho'')$, there exists an execution $\rho = \{(q_k, \Delta_k)\}_{k=1}^{|\rho|} \in \mathcal{L}^{\mathcal{C}_{q_1',q_1''}}$ such that $(q_k, \Delta_k) = ((q_k', q_k''), \Delta_k' = \Delta_k'')$, $\forall k = 1, \ldots, |\rho|$ and $P(\rho) = P(\rho') = P(\rho'')$.
2) For each execution $\rho = \{(q_k, \Delta_k)\}_{k=1\cdots|\rho|} \in \mathcal{L}^{\mathcal{C}_{q_1',q_1''}}_{\{(q_c,q)\}}$, $(q_c, q) \in Q_c \times (Q \setminus Q_c)$, $\Delta_{|\rho|} = 0$, there exists a pair of executions $\rho' = \{(q_k', \Delta_k')\}_{k=1\cdots|\rho'|} \in \mathcal{F}_0^{\mathcal{T}}$, $\rho'' = \{(q_k'', \Delta_k'')\}_{k=1\cdots|\rho''|} \in \mathcal{L}^{\mathcal{T}} \setminus \mathcal{F}^{\mathcal{T}}$ with $P(\rho') = P(\rho'')$ such that $(q_k, \Delta_k) = ((q_k', q_k''), \Delta_k' = \Delta_k'')$, $\forall k = 1 \cdots |\rho|$ and $P(\rho) = P(\rho') = P(\rho'')$.

We define the set $Q_0^{\mathcal{C}} \triangleq \left\{(q_c, q) \in Q_c \times (Q \setminus Q_c) : \exists (q_0', q_0'') \in Q_0 \times Q_0, (q_c, q) \in Q^{\mathcal{C}_{q_0',q_0''}}\right\}$. Notice that $Q_0^{\mathcal{C}}$ is the set of pairs $(q_c, q) \in Q_c \times (Q \setminus Q_c)$ such that there exist two executions ρ', ρ'' with the same timed observation, where the last visited state of ρ' is q_c, while the last visited state of ρ'' is q. Consider now, for each pair $(q_c, q) \in Q_0^{\mathcal{C}}$, the product durational graph $\mathcal{D}_{q_c,q}$ obtained using a version of Algorithm 3 where line 2.2 is deleted. By construction, an execution of $\mathcal{D}_{q_c,q}$ models suffixes of two parallel executions of $\mathcal{G}$ having the same observation. The first one is a suffix starting from q_c of a faulty execution of $\mathcal{G}$, and the second one is a suffix starting from q of a non faulty execution of $\mathcal{G}$. This property is a direct consequence of the classical product automaton construction, and can be formalized by the following proposition.

Proposition 10: Given $\mathcal{G}$ and $\mathcal{D}_{q_c,q}$:

1) For each $\delta > 0$ and pair of executions $\rho' = \{(q_k', \Delta_k')\}_{k=1}^{|\rho'|} \in \mathcal{F}_{\delta}^{\mathcal{G}}$ and $\rho'' = \{(q_k'', \Delta_k'')\}_{k=1}^{|\rho''|} \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$ with $P(\rho') = P(\rho'')$, k_c the index of the first faulty state of ρ', $\rho'_{k_c} = q_c \in Q_c$ and $\rho''_{k_c} = q \in Q \setminus Q_c$, there exists an execution $\rho = \{(q_k, \Delta_k)\}_{k=1}^{|\rho|} \in \mathcal{L}^{\mathcal{D}_{q_c,q}}$ such that $(q_k, \Delta_k) = ((q'_{k+k_c}, q''_{k+k_c}), \Delta'_{k+k_c} = \Delta''_{k+k_c})$, $\forall k = 1, \ldots, |\rho|$, $time(\rho) = \delta$, and $P(\rho) = P(\rho'|_{k_c}^{|\rho'|}) = P(\rho''|_{k_c}^{|\rho''|})$.
2) For each execution $\rho = \{(q_k, \Delta_k)\}_{k=1}^{|\rho|} \in \mathcal{L}^{\mathcal{D}_{q_c,q}}$, there exists a pair of executions $\rho' = \{(q_k', \Delta_k')\}_{k=1}^{|\rho'|} \in \mathcal{F}_{time(\rho)}^{\mathcal{G}}$, $\rho'' = \{(q_k'', \Delta_k'')\}_{k=1}^{|\rho''|} \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$ with $P(\rho') = P(\rho'')$, k_c the index of the first faulty state of ρ', $\rho'_{k_c} = q_c$ and $\rho''_{k_c} = q \in Q \setminus Q_c$, $(q_k, \Delta_k) = ((q'_{k+k_c}, q''_{k+k_c}), \Delta'_{k+k_c} = \Delta''_{k+k_c})$, $\forall k = 1, \ldots, |\rho|$ and $P(\rho) = P\left(\rho'|_{k_c}^{|\rho'|}\right) = P\left(\rho''|_{k_c}^{|\rho''|}\right)$.

□

[8]Actually, it is sufficient to assume that in the cycles of unobservable edges, at least one edge is associated to a guard that is not a singleton set.

On the basis of Propositions 8, 9, and 10, it is now possible to state necessary and sufficient conditions for δ-diagnosability of the original system $\tilde{\mathcal{G}}$ with ε-edges.

Theorem 1: Given $\tilde{\mathcal{G}}$, Q_c is δ-diagnosable if and only if

$$\forall (q_c,q)\in Q_0^{\mathcal{C}}, \forall \rho \in \mathcal{L}^{\mathcal{D}_{q_c,q}}, time(\rho) < \delta + \inf\{\lambda(q_c)\}. \quad (5)$$

Proof: ($\Rightarrow$) By contradiction, assume that the hypothesis is true and that there exists $(q_c,q)\in Q_0^{\mathcal{C}}$ and $\rho \in \mathcal{L}^{\mathcal{D}_{q_c,q}}$ such that $time(\rho) = \delta^* \geq \delta + \inf\{\lambda(q_c)\}$. This implies that, by Propositions 9 and 10, there exist $\rho' \in \mathcal{F}^{\mathcal{G}}_{\delta^*}$ with $\rho'_{k_c} = q_c$, and $\rho'' \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$ such that $P(\rho) = P(\rho') = P(\rho'')$. By Proposition 8, for all $\lambda^* \in \lambda(q_c)$ there exist $\tilde{\rho}' \in \mathcal{F}^{\tilde{\mathcal{G}}}_{\delta^*-\lambda^*}$ and $\tilde{\rho}'' \in \mathcal{L}^{\tilde{\mathcal{G}}} \setminus \mathcal{F}^{\tilde{\mathcal{G}}}$, such that $P(\tilde{\rho}') = P(\tilde{\rho}'')$. Let $\lambda^* = \inf\{\lambda(q_c)\}$, then $\delta^* - \lambda^* \geq \delta + \inf\{\lambda(q_c)\} - \inf\{\lambda(q_c)\} = \delta$, that is a contradiction.

($\Leftarrow$) By contradiction, assume that the hypothesis is true and that Q_c is not δ-diagnosable for a given $\delta \geq 0$. This implies that there exist $\delta^* \geq \delta$, $\tilde{\rho}' \in \mathcal{F}^{\tilde{\mathcal{G}}}_{\delta^*}$ and $\tilde{\rho}'' \in \mathcal{L}^{\tilde{\mathcal{G}}} \setminus \mathcal{F}^{\tilde{\mathcal{G}}}$ such that $P(\tilde{\rho}') = P(\tilde{\rho}'')$. By Proposition 8, there exist $\rho' \in \mathcal{F}^{\mathcal{G}}_{\delta^*+\lambda^*}, \lambda^* \in \lambda(\rho|_{k_c})$ and $\rho'' \in \mathcal{L}^{\mathcal{G}} \setminus \mathcal{F}^{\mathcal{G}}$, such that $P(\rho') = P(\rho'')$. By Propositions 9 and 10 there exists $q \in Q \setminus Q_c$ such that $(\rho'_{k_c}, q) \in Q_0^{\mathcal{C}}$. Furthermore, there exists $\rho \in \mathcal{L}^{\mathcal{D}_{\rho'_{k_c},q}}$ such that $time(\rho) = \delta^* + \lambda^* \geq \delta + \inf\{\lambda(q_c)\}$, that is a contradiction. ∎

As a first step of the diagnosability verification, we can check if Q_c may be δ-diagnosable for a finite value of δ.

Proposition 11: Given $\mathcal{G}$, Q_c is δ-diagnosable for some finite δ only if, for all (q_c,q), $\mathcal{D}_{q_c,q}$ has the following properties:

1) no edges are associated to a guard set $[a,+\infty)$, $a\in\mathbb{R}_+$;
2) every edge e belonging to a cycle is associated to a guard set $G_e = \{0\}$.

Proof: For each $\mathcal{D}_{q_c,q}$, an execution $\rho \in \mathcal{L}^{\mathcal{D}_{q_c,q}}$ can have an infinite duration $time(\rho) = \infty$ only if either a discrete state is visited forever (Condition 1 is not satisfied), or if there exists a cycle of edges that can be crossed in a nonzero amount of time (Condition 2 is not satisfied). ∎

As discussed above, it is interesting to compute the minimum value δ_m for which Q_c is δ_m-diagnosable. The verification procedure we propose in this paper does not only allow to verify δ-diagnosability for a given δ but, much more important, also to directly compute in polynomial time δ_m for a given durational graph $\mathcal{G}$.

Theorem 2: Given $\mathcal{G}$ such that Q_c is δ-diagnosable for some $\delta < \infty$, the minimum value δ_m such that Q_c is δ_m-diagnosable is given by

$$\delta_m = \max_{\substack{\rho\in\mathcal{L}^{\mathcal{D}_{q_c,q}} \\ (q_c,q)\in Q_0^{\mathcal{C}}}} \{time(\rho) - \inf\{\lambda(q_c)\}\} \quad (6)$$

and can be computed in polynomial time.

Proof: Condition (6) clearly holds by Theorem 1. Computing the maximum duration among all executions for each system $\mathcal{D}_{q_c,q}$ is solvable as follows: let N be the cardinality of the discrete state space of $\mathcal{D}_{q_c,q}$. First construct the set of all paths that contain no cycles: notice that it is bounded by N^2. For each path $q_1,\ldots,q_s$, compute the maximum duration by $\sum_{k=1}^{s-1}\sup\{G_{(q_k,q_{k+1})}\}$. By Proposition 11, all cycles (if any) have time duration 0. This implies that the maximum duration among all executions that contain no cycles is also the maximum duration among all executions in $\mathcal{L}^{\mathcal{D}_{q_c,q}}$. ∎

As a consequence of the proposed verification algorithm, we obtain the following new result on verification of diagnosability of durational graphs.

Theorem 3: The δ-diagnosability verification problem for the class of durational graphs belongs to the complexity class P. □

VI. Diagnosability Verification in an Electromagnetic Valve System for Camless Engines

We apply in this section the proposed abstraction and diagnosability verification algorithms to a simple case study, given by an Electromagnetic Valve System for Camless Engines. The mathematical model is at the same time simple enough to apply and test our results, but yet realistic and nontrivial. Camless electromagnetic valves are devices recently considered to decouple the camshaft and the valve lift dynamics, namely to command the opening and closing phases of the intake and exhaust valves. The main advantage of these devices is the possibility of obtaining the optimal engine efficiency in all operating conditions. One of the main open problems is the control of the impact velocities between the valve and the constraints (typically the valve seat), which should be sufficiently low in order to eliminate acoustic noises and avoid damages of the mechanical components. The problem is complicated by the short time (typically $\sim 3-5$ ms) available at high engine speed to make a transition between the two valve's terminal positions, and the constraint in terms of actuator cost and space limitations. These last aspects imply that one typical request is the absence of the valve position sensor. We consider a simplified model of the electromagnetic valve, represented in Fig. 3 (see [18], [40], [41] and references therein for details). We suppose here to supply only one electromagnet to complete the opening or closing phase. The correct behavior of the valve controlled system can be modeled by the hybrid automaton $\mathcal{H}^1$ shown in Fig. 4: q_1 corresponds to the closing phase, q_2 to the valve completely close, q_3 to the opening phase, and q_4 to the valve completely open. The continuous dynamics can be described by the following equations:

$$\dot{x}_v = v_v,\ \dot{v}_v = \frac{1}{M}(-kx_v - bv_v + F_m + F_d + F_c) \quad (7)$$

describing the motion of the valve and of the connected anchor, where M is the mass. The valve position x_v ranges from $-\varrho$ (open valve) to $+\varrho$ (closed valve). Moreover, an elastic force $-kx_v$, due to some springs and a torsion bar, and a viscous friction $-b\dot{x}_v$ act on the valve steam. Finally, F_d is a disturbance whose main contribution is due to the force of the exhaust gases exiting the cylinder, and $F_c(x_v)$ is the constrain force due to the valve seat and electromagnet surfaces, and is always zero except when $x_v = \pm\varrho$, when $F_c(\pm\varrho) = \pm k\varrho - F_m(\pm\varrho,\phi_m) - F_d$. The anchor is attracted by the supplied electromagnet to close and to open the valve by means of the electromagnetic force

$$F_m(x_v,\phi_m) = -\frac{1}{2}\mathcal{D}_m(x_v)\phi_m^2,$$
$$\mathcal{D}_m = a_m e^{-b_m x_v} + c_m \quad (8)$$

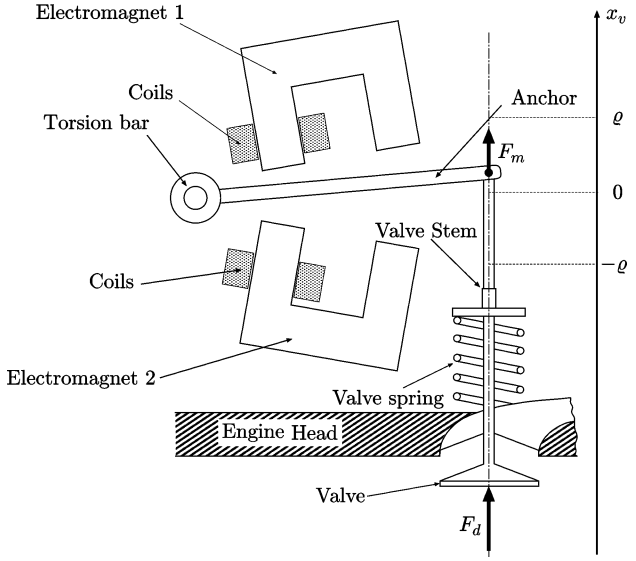

Fig. 3. Scheme of an electromagnetic valve system.

where a_m, b_m, c_m are some constants, and ϕ_m is the flux of the supplied electromagnet $m = 1, 2$. The dynamics of ϕ_m is here neglected for simplicity, since it is much faster than the mechanical ones. Thus, the squared flux can be considered as the control input of the system, i.e., $u = \phi_m^2$. The signs of the constants are such that $\mathcal{D}_1(x_v) > 0$ for the discrete states q_1, q_2, namely when the valve is closing and the electromagnet 1 is supplied, while $\mathcal{D}_2(x_v) < 0$ for the discrete states q_3, q_4, namely when the valve is opening and the electromagnet 2 is supplied. The discrete dynamics depend on the system state (x_v, v_v) and the control input u, according to the guard sets (defined on arrows in Fig. 4) and invariant sets (associated to the discrete states in Fig. 4). The reset functions are all identities. We assume without loss of generality that the initial hybrid state is $(q_1, (-\varrho, 0))$, but our results can be easily extended to more complex sets of initial states. The output of the system is a discrete symbol associated to the edges, i.e., ψ_1 or ψ_2 when the anchor hits respectively electromagnet 1 or electromagnet 2. It follows from [18] that the PD-like control

$$u = \frac{2}{\mathcal{D}_m(x_v)}(p_1(x_v - x_r) + p_2(v_v - v_r) + F_d - kx_r - bv_r - Ma_r).$$

$p_1, p_2 > 0$ ensures the correct behavior of the valve. Here $x_r = (-1)^{m+1}\varrho$, m =1, 2, $v_r = 0$, $a_r = 0$ are the reference values for appropriately operating the valve.

Setting $e = (\, x_v - x_r \quad v_v - v_r \,)^T$, when $F_c = 0$ and using the control above the error dynamics are given by $\dot{e} = A_c e$, where

$$A_c = \begin{pmatrix} 0 & 1 \\ -k_1 & -b_1 \end{pmatrix}, \quad k_1 = \frac{k + p_1}{M} > 0, \quad b_1 = \frac{b + p_2}{M} > 0. \tag{9}$$

We want to address here the diagnosability problem due to large parameter variations, which occur in faults of the device we are considering. In fact, the system parameters k, b are subject to abrupt changes due to possible malfunctions. Let k_0, b_0 be their

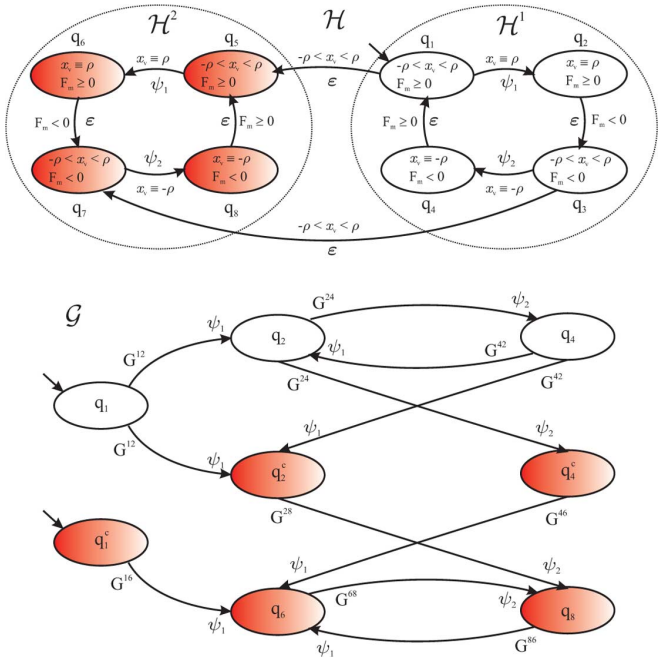

Fig. 4. Hybrid model of the Electromagnetic Valve System $\mathcal{H}$ and abstraction $\mathcal{G}$.

nominal values and $k = k_0 + \Delta k$, $b = b_0 + \Delta b$ the real ones. The controller parameters (p_1, p_2) must be chosen to satisfy the following constraints for the nominal values $k = k_0$ and $b = b_0$:

1) the tracking error goes asymptotically to zero;
2) the norm of the control input is bounded by $u_{\max}$;
3) the seating velocity, i.e., the velocity of the valve when approaching the mechanical constraints, is less than or equal to an appropriate value $v_{\max}$.

From the first assumption we obtain $p_1 > -k$, $p_2 > -b$. Setting $F_{d,\max} = \max_{t \geq 0} F_d(t)$ and $e_{v,\max}$ the maximum velocity error admissible, from the second we obtain

$$|u| \leq \frac{2}{\min_{x_v} \mathcal{D}_m(x_v)} (p_1 2\varrho + p_2 e_{v,\max} + F_{d,\max} + k\varrho) \leq u_{\max} \tag{10}$$

which can be translated as

$$a_1 p_1 + a_2 p_2 - a_3 \leq 0 \tag{11}$$

$a_1 = 1.6 \times 10^{-10}$, $a_2 = 2 \times 10^{-7}$, $a_3 \simeq 1 \times 10^6$, see Table I. For the third assumption, note that the raising time for the error dynamics is $t_r = \pi / \sqrt{4k_1 - b_1^2}$, where $4k_1 - b_1^2 > 0$ to obtain a fast response, namely

$$p_1 > \frac{1}{4M}(b + p_2)^2 - k. \tag{12}$$

Hence, the velocity error has to satisfy $|v_v - v_r|_{t=t_r} = 2\varrho e^{-b_1/2t_r} \leq v_{\max}$, thus obtaining

$$p_1 \leq \frac{1 + n^2}{4Mn^2}(b + p_2)^2 - k, \quad n = \frac{2}{\pi} \ln \frac{2\varrho}{v_{\max}}. \tag{13}$$

A solution to (12) and (13) exists since $1 + n^2/n^2 > 1$. Conditions (11), (12), and (13) for $k = k_0$, $b = b_0$ define the set of

TABLE I
ELECTROMAGNETIC VALVE SYSTEM PARAMETERS

$k_0 = 1.17 \times 10^5$ N/m	$b_0 = 6$ Ns/m	$M = 0.1054$ Kg
$F_{d,\max} = 3410$ N	$\varrho = 4 \times 10^{-3}$ m	$e_{v,\max} = 10$ m/s
$\min_{x_v} \mathcal{D}(x_v) = 1 \times 10^8$	$u_{\max} = 1 \times 10^6$ V	$v_{\max} = 0.05$ m/s

TABLE II
SUMMARY OF NOTATIONS

$s(e)$	Source state of the edge e
$t(e)$	Target state of the edge e
ε	Unobservable output
$inc(q)$	Set of incoming edges in the state q
$out(q)$	Set of outgoing edges from the state q
$cl_\varepsilon(q)$	ε-closure of the state q
$\mathbb{R}_+$	Positive reals
$\rho\|_i$	Discrete state at index i of the execution associated to ρ
$\rho\|_i^j$	Substring of ρ from index i to index j
$time(\rho)$	Time duration of the execution associated to ρ
$\mathcal{L}_{Q^*}$	Set of finite executions that terminate in Q^*
$P(\rho)$	Observation associated to the string ρ
$\mathcal{F}_\delta$	Set of δ faulty executions
$X_0(q_0)$	Set of initial conditions associated to the initial discrete state q_0
$Range(R_e)$	Range of the reset associated to the edge e
$\Gamma(q)$	Set of time instants t such that there exists an execution of duration t that terminates in the state q

controller parameters ensuring the correct behavior. A pair in this set is for instance $(p_1^*, p_2^*) = (2 \times 10^5, 315)$.

We assume that Δk can vary in the interval $[-k_0, k_0]$, where $-k_0$ corresponds to $k = 0$ (the springs are broken). Moreover, we assume that Δb can vary in the interval $[-0.9b_0, 2b_0]$: in other words, the viscous friction can increase up to 200% of the nominal value, and can decrease up to 90% of the nominal value. We define $P = [-k_0, k_0] \times [-0.9b_0, 2b_0]$.

When (k, b) change, the controller may not ensure the correct valve behavior. In fact, the variations $(\Delta k, \Delta b)$ are allowed to belong to a set $P_{safe} \subset P$, but when they exit this set a faulty behavior occurs. In order to determine P_{safe}, let us set $p_1 = p_1^*, p_2 = p_2^*$ in (10), (12), (13), with $k = k_0 + \Delta k, b = b_0 + \Delta b$:

$$\Delta k \leq \frac{a_3 - p_1^* a_1 - p_2^* a_2}{a_0}$$
$$\Delta k > \frac{1}{4M}(b_0 + \Delta b + p_2^*)^2 - p_1^* - k_0$$
$$\Delta k \leq \frac{1+n^2}{4Mn^2}(b_0 + \Delta b + p_2^*)^2 - p_1^* - k_0$$

with $a_0 = 8 \times 10^{-11}$.

We assume that $(\Delta k, \Delta b)$ may abruptly belong to a faulty value in $P_{\text{faulty}} = P \setminus P_{safe}$. In that case, the corresponding dynamics of the controlled system switch to the *faulty dynamics*, modeled in Fig. 4 by the hybrid automaton $\mathcal{H}^2$. The dynamics of each discrete state of $\mathcal{H}^2$ is the same as in $\mathcal{H}^1$, except for the value of the parameters (k, b). The sudden change of the system parameters to a faulty value may occur at any time instant from discrete states q_1, q_3, and is associated to an unobservable output. The overall model $\mathcal{H}$ takes into account the fault occurrence as illustrated in Fig. 4: we assume that the system does not return to a correct behavior once it switches to a faulty behavior. Although $\mathcal{H}$ is an autonomous hybrid system, it is slightly different from the model in Definition 1, since $(\Delta k, \Delta b)$ can be viewed as continuous disturbances that non deterministically assume values in P_{safe} or P_{faulty}, and determine a correct or faulty behavior. However, since the guards are 1-D and the dynamics linear, we can construct our durational graph abstraction by applying the Matlab algorithm we developed in [36], including the system parameters in the state space and considering P_{safe} and P_{faulty} as sets of initial conditions. Thus, erasing ε-edges by means of Algorithm 2, it is possible to construct the durational graph $\mathcal{G}$ shown in Fig. 4. In order to determine the guards G^{ij} of $\mathcal{G}$, we compute the minimum and maximum time for the anchor to touch electromagnet 1 starting from electromagnet 2 and vice versa, considering $(\Delta k, \Delta b) \in P_{safe}$ for the guards of the non-faulty states and $(\Delta k, \Delta b) \in P_{\text{faulty}}$ for the guards of the faulty states. The invariant sets can be defined by $Inv_{q_i} = [0, \max_j\{\sup\{G^{ij}\}\}]$ for each discrete state of $\mathcal{G}$. The construction of $\mathcal{G}$ yields the faulty set $Q_c = \{q_1^c, q_2^c, q_4^c, q_6, q_8\}$ and the faulty function defined as follows: $\lambda(q_1^c) = G^{12}$, $\lambda(q_2^c) = G^{24}$, $\lambda(q_4^c) = G^{42}$, $\lambda(q_6) = \lambda(q_8) = \{0\}$. It is easy to check that there exists a finite δ such that $\mathcal{H}$ is δ-diagnosable only if the following logical formula holds:

$$(G^{12} \cap G^{68} = \varnothing) \vee (G^{42} \cap G^{86} = \varnothing)$$
$$\wedge$$
$$(G^{12} \cap G^{28} = \varnothing) \vee (G^{12} \cap G^{68} = \varnothing) \vee (G^{42} \cap G^{86} = \varnothing)$$
$$\wedge$$
$$(G^{42} \cap G^{46} = \varnothing) \vee (G^{12} \cap G^{68} = \varnothing) \vee (G^{42} \cap G^{86} = \varnothing)$$

which is true if and only if $G^{12} \cap G^{68} = \varnothing$. Using the Matlab algorithm developed in [36] we compute $G^{12} = [1.81, 2.48]$ ms and $G^{68} = [1.71, 1.92] \cup [2.30, 3.42]$ ms, thus $G^{12} \cap G^{68} \neq \varnothing$. Namely, if the components of the valve system admit parameters uncertainty within the set P_{faulty}, then P_{faulty} is not δ-diagnosable for any δ, and it is not possible to detect faults in finite time. In other words, the diagnosability verification procedure has given a negative answer. However, there may exist a subset $P^*_{\text{faulty}} \subset P_{\text{faulty}}$ which is diagnosable. In this case, one can re-design the control system (e.g., by replacing some mechanical components of the valve) so that the parameter uncertainty lies in P^*_{faulty}. A control system designed like this is guaranteed to be diagnosable.

For this reason, searching for a *diagnosable* subset of faulty behaviors $P^*_{\text{faulty}} \subset P_{\text{faulty}}$ can be viewed as a support to the design of the control system. In particular, the search for a diagnosable set P^*_{faulty} (e.g., by a trial-and-error process), may yield to the maximal faulty set (or an approximation of it) such that the system is diagnosable.

For our valve system, we defined $P^*_{\text{faulty}} \triangleq [-1.17 \times 10^5, -0.7 \times 10^5) \times [-5.4, +12]$, as illustrated in Fig. 5. According to the faulty set P^*_{faulty}, we compute $G^{68} = (2.48, 3.42]$ ms: thus, $G^{12} \cap G^{68} = \varnothing$ and P^*_{faulty} is δ-diagnosable for some finite δ. Using Theorem 2, one obtains that the minimum value δ_m such that the set P^*_{faulty} is δ_m-diagnosable is $\delta_m = 2 \cdot \sup\{G^{12}\} = 4.96$ ms.

We can conclude that if the control system is designed so that parameters uncertainty lies in the set P_{faulty}, then faults cannot be diagnosed in finite time. However, if the control system is

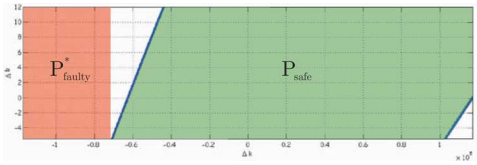

Fig. 5. Faulty sets.

designed so that parameters uncertainty lies in the set P^*_{faulty}, then faults can be diagnosed within 4.96 ms.

VII. Conclusion

We proposed a novel verification procedure for checking diagnosability for a hybrid automaton whose output is a timed string taking values on a finite set. We proposed a definition of δ-diagnosability that generalizes the notion of observability. To verify this property, which is not easy to check and may be even undecidable for a general hybrid model, we proposed an abstraction procedure. The abstracting system belongs to a subclass of timed automata, which is called durational graph. We also proposed a novel algorithm to construct a durational graph $\mathcal{G}$, from a given hybrid automaton $\mathcal{H}$, and showed that it preserves diagnosability. We proposed a novel algorithm to check diagnosability on durational graphs and to directly compute the minimum value δ_m for which a system is δ_m-diagnosable, and proved that the verification problem belongs to the complexity class P. Theoretical results were applied to an electromagnetic valve system for camless engines.

Our current work is concerned with the extension of our diagnosability definition to hybrid (continuous and discrete) faulty sets, in order to model more complex faults, and with the characterization of the distance between trajectories of the original hybrid system and trajectories of the durational graph abstraction, using the notion of approximate bisimulation.

References

[1] 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," *Theoret. Comput. Sci.*, vol. 138, pp. 3–34, 1995.

[2] M. D. Di Benedetto, S. Di Gennaro, and A. D'Innocenzo, "Error detection within a specific time horizon and application to air traffic management," in *Proc. Joint 44th IEEE Conf. Decision Control and Eur. Control Conf. (CDC-ECC'05)*, Seville, Spain, Dec. 2005, pp. 7472–7477.

[3] M. D. Di Benedetto, S. Di Gennaro, and A. D'Innocenzo, "Critical states detection with bounded probability of false alarm and application to air traffic management," in *Proc. 2nd IFAC Conf. Anal. Design of Hybrid Syst. (ADHS)*, Alghero, Sardinia, Italy, Jun. 7–9, 2006, pp. 24–29.

[4] G. K. Fourlas, K. J. Kyriakopoulos, and N. J. Krikelis, "Diagnosability of hybrid systems," in *Proc. 10th Mediterranean Conf. Control Autom. (MED2002)*, Lisbon, Portugal, Jul. 9–12, 2002, pp. 3994–3999.

[5] A. Sheth, C. Hartung, and R. Han, "A decentralized fault diagnosis system for wireless sensor networks," in *Proc. 2nd IEEE Int. Conf. Mobile Ad-Hoc and Sens. Syst. (MASS) 2005*, 1999, pp. 192–194.

[6] F. Lin, "Diagnosability of discrete event systems and its applications," *J. Discrete Event Dyn. Syst.*, vol. 4, no. 1, pp. 197–212, May 1994.

[7] P. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[8] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.

[9] T. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially-observed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 47, no. 9, pp. 1491–1495, Sep. 2002.

[10] A. Paoli and S. Lafortune, "Safe diagnosability for fault tolerant supervision of discrete event systems," *Automatica*, vol. 41, no. 8, 2005.

[11] S. Tripakis, *Fault Diagnosis for Timed Automata*, W. Damm and E. R. Olderog, Eds. New York: Springer-Verlag, 2002, vol. 2469, Lecture Notes in Computer Science, pp. 205–221.

[12] S. McIlraith, G. Biswas, D. Clancy, and V. Gupta, "Hybrid systems diagnosis," in *Hybrid Systems: Computation and Control*, N. Lynch and B. Krogh, Eds. New York: Springer, 2000, vol. 1790, Lecture Notes in Computer Science, pp. 282–295.

[13] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Trans. Syst., Man, Cybern. I—Part B*, vol. 35, no. 6, pp. 1225–1240, Dec. 2005.

[14] F. Laroussinie, N. Markey, and P. Schnoebelen, "Efficient timed model checking for discrete-time systems," *Theoret. Comput. Sci.*, vol. 353, no. 1–3, pp. 249–271, Mar. 2006.

[15] S. Yovine, "Kronos: A verification tool for real-time systems," *Int. J. Softw. Tools Technol. Transf.*, vol. 1, no. 1, pp. 123–133, Oct. 1997.

[16] K. G. Larsen, P. Pettersson, and W. Yi, "UPPAAL in a nutshell," *Int. J. Softw. Tools Technol. Transf.*, vol. 1, no. 1, pp. 134–152, Dec. 1997.

[17] T. Henzinger, P.-H. Ho, and H. Wong.-Toi, "Hytech: A model checker for hybrid systems," *Softw. Tools Technol. Transf.*, vol. 1, pp. 110–122, 1997.

[18] S. Di Gennaro, B. C. Toledo, and M. D. Di Benedetto, "Non-linear control of electromagnetic valves for camless engines," *Int. J. Control: Special Iss. Automotive Control*, vol. 80, no. 11, pp. 1796–1813, 2007.

[19] A. D'Innocenzo, M. D. Di Benedetto, and S. Di Gennaro, "Observability of hybrid automata by abstraction," in *Hybrid Systems: Computation and Control*, J. Hespanha and A. Tiwari, Eds. New York: Springer-Verlag, 2006, vol. 3927, Lecture Notes in Computer Science, pp. 169–183.

[20] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica, Special Iss. Hybrid Syst.*, vol. 35, pp. 349–370, 1999.

[21] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages and Computation*. Boston, MA: Addison-Wesley, 1979.

[22] R. Alur and D. Dill, "A theory of timed automata," *Theoret. Comput. Sci.*, vol. 126, pp. 183–235, 1994.

[23] E. Clarke, O. Grumberg, and D. Peled, *Model Checking*. Cambridge, MA: MIT Press, 2002.

[24] H. Anai and V. Weispfenning, "Reach set computations using real quantifier elimination," in *Hybrid Systems: Computation and Control*, M. D. Di Benedetto and A. Sangiovanni-Vincentelli, Eds. New York: Springer-Verlag, 2001, vol. 2034, Lecture Notes in Computer Science, pp. 103–117.

[25] G. Lafferriere, G. J. Pappas, and S. Yovine, "Symbolic reachability computations for families of linear vector fields," *J. Symbol. Comput.*, vol. 32, no. 3, pp. 231–253, Sep. 2001.

[26] E. Asarin, O. Bournez, T. Dang, and O. Maler, "Approximate reachability analysis of piecewise linear dynamical systems," in *Hybrid Systems: Computation and Control*. Pittsburgh, PA: Springer-Verlag, 2000, Lecture Notes in Computer Science.

[27] H. Yazarel and G. J. Pappas, "Geometric programming relaxations for linear system reachability," in *Proc. Amer. Control Conf.*, Boston, MA, Jun. 2004, pp. 553–559.

[28] A. Chutinan and B. Krogh, "Computing polyhedral approximations to flow pipes for dynamic systems," in *Proc. 37th IEEE Conf. Decision Control*, Tampa, FL, Dec. 1998, pp. 2089–2094.

[29] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *Hybrid Systems: Computation and Control*, M. Morari and L. Thiele, Eds. New York: Springer-Verlag, 2005, vol. 3414, Lecture Notes in Computer Science, pp. 291–305.

[30] L. Guibas, A. Nguyen, and L. Zhang, "Zonotopes as bounding volumes," in *Proc. 14th Annu. ACM-SIAM Symp. Discrete Algorithms.*, Baltimore, MD, 2003, pp. 803–812.

[31] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in *Hybrid Systems: Computation and Control*, N. Lynch and B. Krogh, Eds. New York: Springer-Verlag, 2000, Lecture Notes in Computer Science 1790, pp. 202–214.

[32] A. B. Kurzhanski and P. Varaiya, Ellipsoidal toolbox Elect. Eng. Comput. Sci., UC Berkeley, 2006, Tech. Rep..

[33] Z. Han and B. H. Krogh, "Reachability analysis of large—Scale affine systems using low—Dimensional polytopes," in *Hybrid Syst.: Comput. Control*, J. Hespanha and A. Tiwari, Eds. New York: Springer-Verlag, 2006, vol. 3927, Lecture Notes in Computer Science, pp. 287–301.

[34] A. A. Julius, G. Fainekos, M. Anand, I. Lee, and G. J. Pappas, "Robust test generation and coverage for hybrid systems," in *Hybrid Systems: Computation and Control, To Appear*. New York: Springer-Verlag, 2007, Lecture Notes in Computer Science.

[35] [Online]. Available: http://wiki.grasp.upenn.edu/hst/index.php?n=main.homepage

[36] A. D'Innocenzo, A. A. Julius, G. J. Pappas, M. D. Di Benedetto, and S. Di Gennaro, "Verification of temporal properties on hybrid automata by simulation relations," in *Proc. 46th IEEE Conf. Decision Control*, New Orleans, LA, Dec. 12–14, 2007.

[37] A. D'Innocenzo, A. A. Julius, M. D. Di Benedetto, and G. J. Pappas, "Approximate timed abstractions of hybrid automata," in *Proc. 46th IEEE Conf. Decision Control*, New Orleans, LA, Dec. 12–14, 2007.

[38] C. Ozveren and A. Willsky, "Observability of discrete event dynamic systems," *IEEE Trans. Autom. Control*, vol. 35, pp. 797–806, 1990.

[39] C. Choffrut and M. Goldwurm, "Timed automata with periodic clock constraints," *J. Automata, Lang. Combinatorics*, vol. 5, pp. 371–404, 2000.

[40] C. A. Lua, B. C. Toledo, M. D. Di Benedetto, and S. D. Gennaro, "Feedback regulation of electromagnetic valves for camless engines," in *Proc. Eur. Control Conf.—ECC'07*, Kos, Greece, Jul. 2–5, 2007, pp. 4103–4110.

[41] C. A. Lua, B. C. Toledo, M. D. Di Benedetto, and S. D. Gennaro, "Output feedback regulation of electromagnetic valves for camless engines," in *Proc. Amer. Control Conf.*, New York, Jul. 11–13, 2007, pp. 2967–2972.

Maria D. Di Benedetto (M'89–SM'93–F'02) received the "Dr. Ing." degree (*summa cum laude*) in electrical engineering and computer science from the University of Roma "La Sapienza," Rome, Italy, in 1976 and the degree "Docteur-Ingenieur" and the degree "Doctorat d'Etat Sciences," Université de Paris-Sud, Orsay, France, in 1981 and 1987, respectively.

Since 1994, she has been Professor of Control Theory at the University of L'Aquila, L'Aquila, Italy. She is the PI and Director of the Center of Excellence for Research DEWS "Architectures and Design Methodologies for Embedded Controllers, Wireless Interconnects, and System-on-Chip." Since 1995, she has been a Member of the Scientific Committee of the Center of Excellence for Research CETEMPS, University of L'Aquila. She is cofounder and member of the Governing Board of WEST Aquila S.r.L.. She has been a Member of the Board of Fondazione MIRROR since 2008. Her research interests revolve around nonlinear control and hybrid systems, with applications to automotive and air traffic control.

Stefano Di Gennaro received the degree in nuclear engineering in 1987 (*summa cum laude*), and the Ph.D. degree in system engineering in 1992, both from the University of Rome "La Sapienza," Rome, Italy.

In October 1990, he joined the Department of Electrical Engineering, University of L'Aquila, as Assistant Professor of automatic control. Since 2001, he has been Associate Professor of automatic control at the University of L'Aquila, L'Aquila, Italy. He holds courses on automatic control and nonlinear control. In 1986, he was Visiting Scientist at the Nuclear Research Center ENEA—Casaccia. He has been a Visiting Professor at the Laboratoire des Signaux et Systemes, CNRS-Paris, of the Department of Electrical Engineering, Princeton University, Princeton, NJ, of the Department of Electrical Engineering and Computer Science, University of California, Berkeley, and of the Centro de Investigacion y Estudios Avanzados del IPN, Unidad Ciudad de Mexico and Unidad Guadalajara, Mexico. He is working in the area of hybrid systems, regulation theory, and applications of nonlinear control.

Alessandro D'Innocenzo received the Laurea degree (*summa cum laude*) in electrical engineering from the University of L'Aquila, L'Aquila, Italy, in 2000 and the Ph.D. degree from the Department of Electrical and Information Engineering, University of L'Aquila, in 2007 with a thesis entitled "Observability and Temporal Properties of Hybrid Systems: Analysis and Verification." He accomplished the International Curriculum Option of Doctoral Studies in Hybrid Control for Complex, Distributed and Heterogeneous Embedded Systems in 2007.

He has been a Postdoctoral Researcher in the Department of Electrical and Information Engineering, University of L'Aquila, from 2007 to 2009, and in the Department of Electrical and Systems Engineering of University of Pennsylvania in 2008. Since January 2010, he has been an Assistant Professor in the Department of Electrical and Information Engineering, University of L'Aquila.

Dr. D'Innocenzo was a recipient of Fondazione Filauro award for Ph.D. students in 2005.