Verification of Hybrid Automata Diagnosability with Measurement Uncertainty

Yi Deng, Alessandro D’Innocenzo, Maria D. Di Benedetto, Stefano Di Gennaro, and A. Agung Julius

Abstract—The problem of system diagnosability verification is concerned with whether a fault in the system operation can be diagnosed by using the system model and observations of the system output. In this paper, we investigate the $(\delta_d, \delta_m)$-diagnosability of hybrid automata, which characterizes the maximum delay for diagnosing faults since their first occurrence, given the measurement uncertainty in observing the system output.

We present a methodology that analyzes the $(\delta_d, \delta_m)$-diagnosability of hybrid automata. Due to the complex dynamics, the hybrid system diagnosability is often difficult to analyze directly. We thus propose an approach of constructing an abstraction using the trajectories of the original system. Their $(\delta_d, \delta_m)$-diagnosability properties are proved to be quantitatively related to each other. The abstraction has only finitely many trajectories that extend to the end of the time horizon of interest, so its diagnosability can be easily calculated, and then used to derive the diagnosability of the original system. We illustrate this procedure with an example.

I. INTRODUCTION

For safety and economic considerations, diagnosing faulty states of system operation plays an important role in industry. For an aircraft, for example, we want the system to be able to locate an abrupt engine malfunction in time to take remedial action so that a catastrophic accident is avoided; or for a household air purifier, we want its filter screen degradation to be tracked and indicated to the user for a replacement; etc. This paper presents fault diagnosis for systems that are modeled as hybrid automata, by only observing ordered discrete events with their timing, which is assumed to be inaccurate to some extent. We reduce the problem to a finite one, and practically compute the diagnosability as well as the diagnoser.

Fault diagnosis involves three tasks: detection (whether something goes wrong), isolation (where it goes wrong), identification (what size the fault has), while the former two deserve primary emphasis [1]. Thus the essential task consists in discriminating a fault from normal system behaviors as well as from other faults. Based on the system model, this can be fulfilled by comparing available measurements with information analytically derived from the mathematical model of systems. As modeled, a fault has some particular pattern of anticipated measurements as its symptoms, which generate the fault signature [2]. Intuitively, fault diagnosability is determined by the discriminability of fault signatures.

Efficient fault diagnosis requires fault signatures that make good use of the measurement and computation resources to increase the signature discriminability. Indeed, when a system operates, only a limited source of information is available for measurement. Such considerations for hybrid systems motivates our research work in this paper. By definition, a hybrid system has interacting discrete and continuous dynamics. The system state can flow continuously, and can also jump by triggering an event. We want to diagnose faults for hybrid systems based on the discrete event sequences for the following reasons: First, without tracking discrete events, the system model and measurement information may be underutilized. Second, for a hybrid system where multiple modes (discrete states, also called locations) exist, if one does not track events, a mechanism that works in one location to diagnose faults may be invalid when the location changes. Third, it is presented in this paper that observing the timed event sequences can be sufficient for diagnosing failures. Consider a bouncing ball for instance. Suppose the only sensor is human ears that hear the sound when the ball hits the ground. By just tracking a sequence of sounds, one can tell whether the ball leaks, or is subjected to a sudden impact, etc. The time intervals between the sounds here are informative for fault diagnosis.

To study the fault diagnosability analysis problem for hybrid systems, we use some ideas on fault detection and isolation from the FDI (fault detection and isolation) community. There, information from measurement and model is synthesized in so-called residuals [3], which serve as the basis for diagnosis. A fault corresponds to a particular pattern of residuals (fault signature). So the diagnostic problem comprises of residual generation and decision making stages [4], see Fig.1 from [5]. This approach has wide application in continuous-time and discretized dynamical system models [6], [7]; but for hybrid systems, diagnosing faults in this way does not work efficiently due to the difficulty in capturing event sequences.

Fig. 1. Fault diagnosis in FDI community [5].

On the other hand, it is presented in [8], [9] that a hybrid system can be cast into the discrete event system (DES) framework so that approaches for DES diagnosis based on the order of events [10] can be drawn on. The present paper also reports on the hybrid system diagnosability problem relying on ordered events, but besides the event sequences we also want to make use of another property of the traces of the system that is closely related to the continuous dynamics, i.e., the time intervals between observed events. With timed behaviors considered, better diagnosability can be achieved due to the

0018-9286 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
discriminability increased by the temporal properties; relevant works can be found in [11], [12].

Due to the limited measurement and computation resource, not every event is observable, and moreover the measurement uncertainty can be introduced thereby. It is clear that the inaccuracy of actual measurements can impair the discriminability of observed symptoms. Fault diagnosability is then affected. We thus formally formulate the fault diagnosability analysis problem in this paper for hybrid automata as follows: Suppose for the system, one can only observe events with their timing. Moreover, only a subset of the events are modeled as observable, and the timing has limited measurement accuracy. Any event that causes the system state to enter a faulty set is called a faulty event. We want to analyze whether the occurrence of any faulty event can be deduced within a limited time.

With this setting, sequences of timed observable events serve as the information available for fault diagnosis. In other words, symptoms are given by them; but due to the intrinsic complexity of hybrid dynamics, it can be rather difficult to analyze their discriminability directly. As illustrated in later sections, such analysis requires a pairwise comparison of all the possible observed event sequences, including comparing the events themselves and their timing. In general this is impossible for hybrid systems. We thus present a methodology that solves the problem in an indirect way: In Section II, we propose to construct for the hybrid automaton a system abstraction, whose diagnosability is very easy to analyze; in Section III, a quantitative relation between the diagnosability of a hybrid automaton and its system abstraction is proved, which allows us to derive the former from the later. An example is presented to illustrate the methodology.

The present paper uses $(\delta_d,\delta_\alpha)$-diagnosability definition for hybrid automata that characterizes the limitation on the time delay for a fault to be diagnosed, as well as the uncertainty in measuring time intervals (see Section III-B). The $(\delta_d,\delta_\alpha)$-diagnosability definition is initially proposed in [13], which we extend from faulty detectability to both detectability and isolability. The quantitative relation between the diagnosability of a hybrid automaton and its system abstraction can be found in Theorem 1. Another significant contribution of this paper is that we demonstrate a method to reduce a hybrid automaton to an abstraction of finitely many trajectories that extend to the end of the time horizon of interest (see Section II-C). This, in turn, allows us to practically diagnose faults that appear in the time horizon of interest. System abstraction has been studied previously for reachability analysis [14], [15], [16], [17], [18] and controller synthesis [19], [20], [21], [22], [23]. Compared to these literature, our work is different in the sense that: (i) We construct the abstraction for finite-time fault diagnosis; (ii) Our abstraction of hybrid automata does not grid the state space, so the computational cost does not necessarily grow exponentially as the state space dimension increases. The abstraction consists of only finitely many simulated trajectories. Its diagnosability can be easily calculated, and then used to analyze the diagnosability of the original hybrid automaton and construct the diagnoser.

II. HYBRID AUTOMATA ABSTRACTION

A. Definition of Hybrid Automata

Hybrid automata are autonomous systems with interacting continuous and discrete dynamics, where the interaction between the continuous and discrete part can be described by events and invariant sets. We use the definition proposed by [24]:

Definition 1 (Hybrid Automaton) A hybrid automaton $H = (L \times X, L^0 \times X^0, D, E, Inv)$ is a tuple that consists of:

- A possibly infinite set $L \times X$ of hybrid states $(\ell, x)$, where $\ell \in L$ is the discrete state, and $x \in X$ is the continuous state. Discrete states are also called locations.
- A possibly infinite set $L^0 \times X^0$ of initial states.
- $D$ associates with each location $\ell \in L$ the autonomous continuous time-invariant dynamics, $D_\ell : \tilde{x} = D_\ell(x)$. We assume that this differential equation admits a unique global solution $\xi_\ell(t, x^0_\ell)$, where $\xi_\ell(0, x^0_\ell) = x^0_\ell$ is the initial condition in $\ell$, and the function $\xi_\ell$ satisfies $\frac{d\xi_\ell(t, x^0_\ell)}{dt} = D_\ell(\xi_\ell(t, x^0_\ell))$.
- $Inv : L \rightarrow X$ associates with each location an invariant set $Inv(\ell) \subset X$. Only if the continuous state satisfies $x \in Inv(\ell)$, can the discrete state be at the location $\ell$.
- $E$ is a set of events. In each location $\ell$, the system evolves continuously according to $D_\ell$ until an event $e := (\ell, \ell', g, r, e) \in E$, $e \in E$ occurs. The event is guarded by $g \subset Inv(\ell)$. Namely, a necessary condition for the occurrence of $e \in g \subset E$.
- After the event, the discrete state changes from the source location $\ell$ to the target location $\ell'$, and the continuous state is reset according to the reset map $r : Inv(\ell) \rightarrow Inv(\ell')$. Let $(\ell, x)$ denote the system state that triggers $e = (\ell, \ell', g, r)$. Then the reset state is $\ell' (r(x))$.

Let $G_\ell$ denotes the set union of guards such that the associated events all have $\ell$ as the source location. Let $\partial Inv(\ell)_{out}$ denote part of the boundary $\partial Inv(\ell)$ where the continuous state is evolving outward $Inv(\ell)$, i.e., given $g_\ell(\tau, x_\ell^0) \in \partial Inv(\ell)_{out}$, for any $t > 0$, there exists $t_0 \in (0, t)$ such that $g_\ell(\tau, x_\ell^0) \notin Inv(\ell)$. We adopt the following assumptions:

1) Non-deadlocking. For every location $\ell$, we require $\partial Inv(\ell)_{out} \subset G_\ell$ to avoid deadlocking. Namely, whenever the continuous state is evolving outside $Inv(\ell)$, a jump must be specified.

2) Non-determinism. When the continuous state reaches a guard, an event may or may not occur (unless it is at $\partial Inv(\ell)_{out}$, where an event is forced to occur). Moreover, guards associated with different events can overlap, where any of the events may occur.

3) Well-posedness. The differential equation $\dot{x} = D_\ell(x)$ admits a unique solution, namely, it satisfies the Lipschitz condition.

4) The system does not have Zeno behavior [25].

5) All the reset maps are continuous functions.

6) The initial set is compact.

When the hybrid system operates, a sequence of events can be triggered. Some of the events are observable, while the rest are not. We associate each observable event with an observable output symbol $\psi \in \Psi_o$, and the unobservable events with the empty output symbol $\psi = \emptyset$. The initial state is not a reset state of an event $e \in E$, but it is assumed that we know when the system starts to operate, i.e., when to start the timer in the fault diagnosis. For convenience, we define an initialization event $e^0 \notin E$ associated with the output symbol $\ell$ (starting signal). Then a trajectory of the hybrid system can be defined as a sequence:

Definition 2 (Trajectory). Given $H = (L \times X, L^0 \times X^0, D, E, Inv)$, a trajectory of $H$ is

$$\rho = (e^0, \ell^0, x^0, x^0, e^1, \ell^1, x^1, x^1, \ell^2, x^2, x^2, \cdots) = \{(e^i, \ell^i, x^i, x^i)\}_{i=0}^{N}$$

such that:

- for all $i \geq 0$, $(\ell^i, x^i) \in L \times X$, and $(\ell^i, x^i) \in L^0 \times X^0$;
- for all $i \geq 0$, $\tau^i \in \mathbb{R}_{\geq 0}$, and $\xi_{\ell^i}(t, x^i) \in Inv(\ell^i)$ for all $t \in [0, \tau^i]$;
- for all $i \geq 1$, $e^i = (\ell^{i-1}, \ell^i, g^i, r^i, \ell^i)$ in $E$, $\ell_{\xi_{\ell^i-1}}(\tau^{i-1}, x^{i-1}) \in g^i$, and $x^i = r^i(\ell_{\xi_{\ell^i-1}}(\tau^{i-1}, x^{i-1}))$, i.e., $(\ell^i, x^i)$ is the reset state of $(\ell^i-1, x^{i-1})$.
state.

Suppose there is a trajectory \( \rho' = \{(e'_i, \ell'_i, x'_i, \tau'_i)\}_{i=0}^{N'} \) such that \( N' \leq N, (e'_i, \ell'_i, x'_i, \tau'_i) = \ldots \{\text{Robust}(\hat{\ell}_0, \hat{x}_0)| (\hat{\ell}_0, \hat{x}_0) \in L_0 \times X_0\} \) are not bounded from below by a positive number. Consequently, randomly generating

\[ \text{Definition 4 (Label Sequence). Given a trajectory } \rho = \{(e_i, \ell_i, x_i, \tau_i)\}_{i=0}^{N}, \text{ the sequence of timed output symbols produced by } \rho \text{ is } s = (\Delta^0, \psi_0), (\Delta^1, \psi^1) \ldots = (\{\Delta^i, \psi^i\}_{i=0})_{i=1}, \]

where \( (\Delta^0, \psi_0) = (0, 1) \), and for all \( i \geq 1, \Delta^i = \tau^i-1, \psi^i \in \Psi_0 \cup \{0\} \) is the output symbol associated with \( e \in E \). For convenience, we define a set of labels \( \Sigma := R_\infty \times (\Psi_0 \cup \{0, 1\}) \), and refer to a sequence of timed output symbols as a label sequence.

Clearly, each timed event sequence comprises of a time sequence and an event sequence, and each label sequence comprises of a time sequence and an output symbol sequence.

**B. Timed Language**

Over the set of events \( E \), a set of timed event sequences is called a timed language. Let \( \chi(E) \) denote the set of all the timed event sequences that can be generated over \( R_\infty \times E \). We define a metric on \( \chi(E) \) in such a way that the distance only depends on the time sequences if the event sequences of two timed event sequences are the same, and raised to infinity otherwise.

**Definition 5 (Timed Event Sequence Metric). Given } p_0 = \{e_1, \tau_1\}_{i=0}^{N_0}, p_2 = \{e_2, \tau_2\}_{i=0}^{N_2} \in \chi(E),

\[ d_\chi(p_1, p_2) = \begin{cases} 
\sup_i d_\Theta(t_i, \tau_i) & \text{if } N_1 = N_2, \\
\infty & \text{otherwise,}
\end{cases} \]

where \( d_\theta(t, \tau) = \begin{cases} 
0 & \text{if } t = \tau, \\
|t - \tau| & \text{if } t \neq \tau
\end{cases} \).

**Definition 6 (Relative Timed Metric).**

\[ d_\Phi(t_1, t_2) = \begin{cases} 
\frac{|t_1 - t_2|}{t_1 + t_2} & \text{if } t_1 = t_2 = 0, \\
\frac{|t_1 - t_2|}{t_1 + t_2} & \text{if } t_1 = t_2.
\end{cases} \]

By Definition 2, when \( \rho \) is a sub-trajectory of \( \rho' \), \( N \leq N' \), \( \{e'_i, \ell'_i, x'_i, \tau'_i\} = \{e^i, \ell^i, x^i, \tau^i\} \) for all \( i \in [0, N-1], \chi(N, x_N) = (\chi(N, x_N), x_N) \leq \tau_N, \) and thus \( \chi(N, x_N) \).

Note that if the trajectories initiated from \( (\tilde{\rho}^0, z^0) \) trigger multiple timed event sequences (see the non-determinism assumption in Section II-A), such robust neighborhood might not exist. We thus need to use the adapted approach presented in [27] to compute robust neighborhoods (called safe neighborhoods in [27] because of the application to system safety verification) around such an initial state. Instead of a single trajectory \( \rho \), a finite set of representative trajectories \( \rho \) are simulated from \( (\tilde{\rho}^0, z^0) \) (triggering multiple timed event sequences), and used to compute \( \text{Robust}(\tilde{\rho}^0, z^0) \). As a result, for any \( (\tilde{\rho}^0, x^0) \in \text{Robust}(\tilde{\rho}^0, z^0) \),warf any trajectory \( \rho' = \{e'_i, \ell'_i, x'_i, \tau'_i\}_{i=0}^{N'} \) is initiated from \( (\tilde{\rho}^0, x^0) \), there exists \( \hat{\rho} = \{e'_i, \ell'_i, x'_i, \tau'_i\}_{i=0}^{N'} \) such that \( \hat{\rho} \) is a sub-trajectory of \( \rho' \).
robust neighborhoods and covering $L^0 \times X^0$ may never terminate with 100% coverage.

This problem is fixed by setting the threshold value $d_{thr}$ in [27] to a positive number instead of 0. Then for each $(\hat{p}^0, \hat{x}^0)$ we obtain $[\hat{p}]^e$, a finite set of representative (virtual) trajectories simulated from $(\hat{p}^0, \hat{x}^0)$. In the process of simulating $[\hat{p}]^e$, an event can be triggered even if the corresponding guard is not reached by the continuous system state (but the system state should be within $d_{thr}$ distance to the guard). Therefore, some trajectories non-existent according to the original discrete dynamics are included in $[\hat{p}]^e$. These are called virtual trajectories [27].

**Definition 8 (Virtual Trajectory).** Given $H = (L \times X, L^0 \times X^0, D, E, Inv)$, a metric $\phi : X \times X \to \mathbb{R}_{\geq 0}$, and $d_{thr} \geq 0$, a virtual trajectory of $H$ is

$$\rho = ((e^i, \hat{e}^0, x^0, \tau_i^0), (e^1, \hat{e}^1, x^1, \tau_i^1), \ldots) = \{(e^i, \hat{e}^i, x^i, \tau_i^i)\}_{i=0}^N,$$

such that

- for all $i \geq 0$, $(\hat{e}^i, x^i) \in L \times X$, and $(\hat{e}^0, x^0) \in L^0 \times X^0$;
- for all $i \geq 0$, $\tau_i^i \in \mathbb{R}_{\geq 0}$, and $\xi_i(x^i) \in Inv(\hat{e}^i)$ for all $i \in [0, \tau_i^i]$;
- for all $i \geq 1$, $e^i = (\hat{e}^{i-1}, \hat{e}^i, g^i, r^i) \in E$, $(e^i, x^i)$ is the reset state for $e^i$, where $x^i = r^i(y^i)$ for some $y^i \in g^i$, and $\phi(\xi_i-1, \tau_i^{i-1}, x_i^{i-1}, y_i^i) \leq d_{thr}$.

Virtual trajectories can be defined in the same manner as Definition 2.

With $[\hat{p}]^e$ simulated, the same algorithm [27] can be used to compute the so-called enlarged neighborhoods $Robust^e(\hat{p}^0, \hat{x}^0)$.

As a result, for any $(\hat{e}^0, x^0) \in Robust^e(\hat{p}^0, \hat{x}^0)$, for any trajectory $\rho^i = ((e^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ initiated from $(\hat{e}^0, x^0)$, there exist $\rho^i = ((\hat{e}^0, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N \in [\hat{p}]^e$, and a trajectory $\rho = ((e^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ such that the properties listed in Prop. 2 hold. Moreover, the radii of $Robust^e(\hat{p}^0, \hat{x}^0)$ are bounded from below by a positive number [27]. Therefore, $L^0 \times X^0$ can be fully covered by finitely many neighborhoods that are randomly generated. We denote the simulated initial states as $L^0 \times X^0 \subseteq L^0 \times X^0$, whose enlarged robust neighborhoods (or robust neighborhoods if possible) cover $L^0 \times X^0$.

The system abstraction is constructed with the finitely many (virtual) trajectories simulated in the construction of $L^0 \times X^0$. We use $\mathcal{J}(H)$ to denote the set of sub-trajectories of these simulated (virtual) trajectories such that the trajectory horizon does not exceed the specified horizon. By definition, sub-trajectories include the trajectory itself. Depending on the whether the horizon is specified by the total dwell time $t_{max}$ or the number of events triggered $N_{max}$, $\mathcal{J}(H)$ can be $\mathcal{J}(t_{max}, H)$ or $\mathcal{J}(N_{max}, H)$.

- $\mathcal{J}(t_{max}, H)$ is defined to be the set of all the sub-trajectories $\hat{\rho} = ((\hat{e}^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ of the simulated (virtual) trajectories such that $\sum_{i=0}^N \tau_i^i \leq t_{max}$;
- $\mathcal{J}(N_{max}, H)$ is defined to be the set of all the sub-trajectories $\hat{\rho} = ((\hat{e}^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ of the simulated (virtual) trajectories such that $N \leq N_{max}$.

In what follows, we use $\mathcal{J}(H)$ to denote the trajectories of $H$ that are of interest in the finite-horizon fault diagnosis. $\mathcal{J}(H)$ can be specified as $\mathcal{J}(t_{max}, H)$ or $\mathcal{J}(N_{max}, H)$:

- $\mathcal{J}(t_{max}, H)$ is defined to be the set of all the trajectories $\rho = ((e^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ of $H$ such that $\sum_{i=0}^N \tau_i^i \leq t_{max}$;
- $\mathcal{J}(N_{max}, H)$ is defined to be the set of all the trajectories $\rho = ((e^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ of $H$ such that $N \leq N_{max}$.

Let $\mathcal{P}(t_{max}, H), \mathcal{P}(N_{max}, H), \mathcal{P}(t_{max}, H), \mathcal{P}(N_{max}, H)$ be respectively the set of timed event sequences produced by $\mathcal{J}(t_{max}, H), \mathcal{J}(N_{max}, H), \mathcal{J}(t_{max}, H), \mathcal{J}(N_{max}, H)$. Then the following proposition holds:

**Proposition 3.** Given the simulation horizon $[0, t_{end}]$ (or $N$) in the construction of $L^0 \times X^0$,

1) if $L^0 \times X^0$ is constructed from robust neighborhoods, then we have

$$h(\mathcal{P}(t_{end}, H), \mathcal{P}(t_{end}, H)) \leq \varepsilon, \quad (1)$$

$$h(\mathcal{P}(N(N-1)), \mathcal{P}(N(N-1), H)) \leq \varepsilon, \quad (2)$$

$$h(\mathcal{P}(t_{end}, H), \mathcal{P}(t_{end}, H)) = 0, \quad (3)$$

$$h(\mathcal{P}(N(N-1)), \mathcal{P}(N(N-1), H)) = 0; \quad (4)$$

2) if $L^0 \times X^0$ is constructed from enlarged robust neighborhoods, then we have Eq. (1)(2).

**Proof:** Suppose the simulation horizon for $L^0 \times X^0$ is specified by the total dwell time $t_{end}$. For any timed event sequence $\hat{p}^i \in \mathcal{P}(t_{end}, H)$, by definition, there is a trajectory $\rho^i = ((\hat{e}^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ initiated from $(\hat{e}^0, x^0) \in L^0 \times X^0$ that produces $\rho^i$ and satisfies $\sum_{i=0}^N \tau_i^i \leq t_{end}$. Let $(\hat{e}^0, x^0) \in L^0 \times X^0$ be a simulated initial state whose (enlarged) robust neighborhood covers $(\hat{e}^0, x^0)$. By the property of (enlarged) robust neighborhood, there exist $\rho = ((\hat{e}^i, \hat{e}^i, x^i, \tau_i^i))_{i=0}^N$ such that $N = N^e = \varepsilon^e = d_{\rho}(\tau_i, \varepsilon) \leq \varepsilon$ for all $i \in [0, N]$, where $\rho$ is simulated from $(\hat{e}^0, x^0)$ for the specified horizon, and $\rho$ is a sub-trajectory of $\rho^i$ or the other way around.

By Proposition 1, $\tau_i^i \in [\tau_i^i, \tau_i^i + \varepsilon]$ for all $i \in [0, N]$, which implies

$$t_{end} \leq t_{end} \leq \varepsilon.$$
• for all \((\hat{\ell}^0, \hat{x}^0) \in \hat{L}^0 \times \hat{X}^0, \) for all \(\hat{p} \in [\hat{p}]_e, \) \(H\) has a trajectory \(\rho\) such that \(d_{\chi}(\hat{p}, p) \leq \epsilon.\)

Then instead of Eq. (3)(4), the following holds:

\[
\begin{align*}
\hat{h}(\hat{P}_R(t_{\text{end}}, H), \hat{P}_N(t_{\text{end}})) & \leq 1 + \epsilon \quad \text{if} \quad \epsilon, H \\
\hat{h}(\hat{P}_N(\hat{N} - 1, H), \hat{P}_N(\hat{N} - 1, H)) & \leq \epsilon.
\end{align*}
\]

In this case, we still obtain approximate timed language equivalence. It can be proved that when the positive threshold value \(d_{\chi}, \) in the algorithm of [27] is sufficiently small, such condition can be satisfied; but for the present work to be focused, we do not discuss the proof.

The abstraction has finitely many trajectories that extend to the end of the simulation horizon. Its diagnosability can be easily analyzed and used to derive the diagnosability of \(H.\) We see this in the next section.

III. DIAGNOSABILITY WITH MEASUREMENT UNCERTAINTY

A. Projected Label Sequences

In Section III, we investigate the problem of diagnosing faults for hybrid automata without directly observing the trajectories.

During the system operation, one can only observe a sequence of labels, which is a sequence of timed output symbols; but due to the unobservable events, this observed label sequence may be different from the original one produced by the trajectory. For example, listening to different sounds with time intervals generated to the unobservable events, this observed label sequence may be a sequence of timed output symbols; but due to the measurement uncertainty of time intervals.

Definition 9 (Projected Label Sequence). Let \(S^*\) denote the set of all the label sequences generated over \(S.\) Let \(s = \{(\Delta^i, \psi^i)\}_{i=0}^{N_f} \in S^*\) be a label sequence, and \(\Pi : S^* \rightarrow S^*\) be a single-valued projection map. Then \(\pi := \Pi(s)\) is called the projected label sequence of \(s\) through the map \(\Pi.\)

We define the projection map \(\Pi\) that absorbs all the symbols in the unobservable output symbol \(\emptyset\) into the first observable label that follows, while leaves the rest of labels unchanged. That is, it projects \(s\) to \(\pi\) in the sense that every unobservable output symbol \(\emptyset\) is erased, whose dwell time gets added into the dwell time of the next label that has an observable output symbol. For instance, \((\Delta^0, \psi^0), (\Delta^1, \emptyset), (\Delta^2, \psi^2)\) is projected to \((\Delta^0, \psi^0), (\Delta^1 + \Delta^2, \psi^2).\)

If a trajectory has a consecutive unobservable output symbol at its end, then the unobservable end is abandoned in the projected label sequence. Projected label sequences are the only accessible information for system diagnosis. They contain two aspects of information as below, where \((\Delta^i, \psi^i)\) are the labels after projection:

1) Before the starting signal \(\psi^0 = \epsilon, \) the system operation is clear of faults.

2) Each observable symbol \(\psi^i, i \geq 1\) should be output \(\Delta^i\) time units later than the preceding one; in the meantime, no observable symbol can be output.

In what follows, a metric on \(S^*\) is defined in such a way that the distance only depends on the time sequences if the output symbol sequences of two label sequences are the same, and raised to infinity otherwise. This is motivated by the application of diagnosing faults by observing the projected label sequences. Unlike dwell time, which may be measured with uncertainty, different output symbols are assumed to be readily differentiable from each other.

Definition 10 (Label Sequence Metric). Given \(s_1 = \{\Delta^i_1, \psi^i_1\}_{i=0}^{N_1}, s_2 = \{\Delta^i_2, \psi^i_2\}_{i=0}^{N_2} \in S^*,\)

\[
d_{\Sigma^*}(s_1, s_2) \triangleq \begin{cases} 
\sup_i d_{\chi}(\Delta^i_1, \Delta^i_2) & \text{if } N_1 = N_2, \\
\infty & \text{otherwise;}
\end{cases}
\]

where \(d_{\chi}\) is the relative time metric defined in Section II-B.

B. Definition of \((\delta_d, \delta_m)-\text{Diagnosability}\)

In fault diagnosis two main tasks are fault detection and isolation. Detecting a fault requires its manifestation being discriminable from that of normal system behaviors, while isolating requires further discriminability from symptoms of other faults. To investigate the diagnosability we start from defining the discriminability of a fault from normal behaviors and other faults based on the system model.

Consider a hybrid automaton \(H = (L \times X, L^0 \times X^0, D, E, Inv)\), and the projection map \(\Pi\) defined in Section III-A. Let \(L_f \subset L\) be the set of locations that model a failure: \(L_f\) is called the faulty set, whose elements can be partitioned into \(M\) disjoint subsets \(\bigcup_{j=1}^M L^j_f = L_f.\)

Each fault subset \(L^j_f\) corresponds to a type of fault \(F_j \in F := \{F_1, \ldots, F_M\}.\) The discriminability of a fault \(F_j\) is proposed below, which takes into account the delay in diagnosing a fault since it first occurs, as well as the measurement uncertainty of time intervals.

Definition 11 \((F_j, \delta)-\text{Faulty Trajectory}.\) A trajectory \(\rho = \{((\ell^i, \ell^i, \tau^i, \pi^i))_{i=0}^{N_f}\} \in F_j\)-faulty if and only if there exists a finite index \(i^f \in [0, N_f]\) such that:

1) \(\forall i < i^f, \ell^i \not\in L^j_1;\)

2) \(\ell^i \in L^j_f;\)

3) \(\sum_{i=i^f}^{N_f} \tau^i \geq \delta.\)

In the special case where \(\delta = 0,\) i.e., only the occurrence of \(F_j\) is required, the trajectory \(F_j\)-faulty is used. If a trajectory is free of any fault, then it is called normal.

Definition 12 \((\delta_d, \delta_m)-\text{Diagnosability}.\) Let \(J\) be a set of trajectories, \(\rho_1, \rho_2\) be two arbitrary trajectories in \(J,\) which produce the label sequences \(s_1, s_2,\) and \(\pi_1 = \Pi(s_1), \pi_2 = \Pi(s_2).\)

1) \(F_j\) is called \((\delta_d, \delta_m)-\text{discriminable from normal (with respect to } J)\) if and only if the following is satisfied: if \(\rho_1\) is a \(F_d\)-faulty trajectory, then either \(\rho_2\) is not normal or \(d_{\Sigma^*}(\pi_1, \pi_2) > \delta_m.\)

2) Given \(F_j \in F = \{F_1, \ldots, F_M\}, \) \(F_j\) is called \((\delta_d, \delta_m)-\text{discriminable from } F_i,\) with respect to \(J\) if and only if the following is satisfied: if \(\rho_1\) is a \(F_d\)-faulty trajectory, then either \(\rho_j\) is not \(F_i\)-faulty or \(d_{\Sigma^*}(\pi_1, \pi_2) > \delta_m.\)

In what follows, we define the diagnosability based on the \((\delta_d, \delta_m)-\text{discriminability of faults. We want to perform finite-horizon diagnosability analysis of } H.\) To that end, we need to consider the discriminability of faults with respect to \(J(H).\) As mentioned before, \(J(H)\) can be specified as \(J_R(\tilde{t}_{\text{max}}(H))\) (or \(J_N(\tilde{t}_{\text{max}}(H))\), i.e., the set of all the possible trajectories of \(H\) such that the trajectory horizon does not exceed \(t_{\text{max}}\) or \(N_{\text{max}}\)). We also need to consider the diagnosability of faults with respect to \(J_R(\epsilon_{\text{max}}(H))\) (or \(J_N(\epsilon_{\text{max}}(H))\)), since the diagnosability analysis of the original system has to be performed indirectly from the system abstraction.

Essentially, \(d_{\Sigma^*}(\pi_1, \pi_2) > \delta_m\) requires that a pair of projected label sequences coming from a \(F_d\)-faulty trajectory and a normal/F_i-faulty trajectory can be distinguished from each other.
given the measurement uncertainty $\delta_m$ for time intervals. Thus, $(\delta_d, \delta_m)$-discriminability from normal amounts to the detectability of $F_j$ with the delay $\delta_d$ and measurement uncertainty $\delta_m$. As for isolability, things are more complicated, since $F_j$ has to be $(\delta_d, \delta_m)$-discriminable from all other faults, namely all $F_i \in \mathbb{F} = \{F_1, \ldots, F_M\}$ such that $F_i \neq F_j$. The semantics of isolating $F_j$ from $F_i$ is to exclude the occurrence of $F_i$ as a cause to alarm, relying on the possibly delayed symptoms of $F_j$.

Another remark about the $(\delta_d, \delta_m)$-discriminability of $F_j$ from $F_i$ is on asymmetry. To see this, suppose $p_1, p_2$ are too close to discriminate in the sense that $d_{\delta_d}(\pi_1, \pi_2) \leq \delta_m$. If $p_1$ is $F_j$-faulty and $p_2$ is $F_k$-faulty with $\delta_d < \delta_m$, then $F_j$ is not $(\delta_d, \delta_m)$-discriminable from $F_i$, but $F_k$ can still be $(\delta_d, \delta_m)$-discriminable from $F_j$.

**Definition 13** ($(\delta_d, \delta_m)$-Diagnosability). Given a set of trajectories $\mathcal{J}$ and a faulty set $L^f = \{L_1^f, \ldots, L_M^f\}$, we have the following definitions with respect to $\mathcal{J}$:

- $F_j \in \mathbb{F} = \{F_1, \ldots, F_M\}$ is $(\delta_d, \delta_m)$-detectable if and only if it is $(\delta_d, \delta_m)$-discriminable from normal;
- $F_j$ is $(\delta_d, \delta_m)$-isolable if and only if it is $(\delta_d, \delta_m)$-discriminable from all other faults $F_i \in \mathbb{F}, F_i \neq F_j$;
- $F_j$ is $(\delta_d, \delta_m)$-diagnosable if and only if it is $(\delta_d, \delta_m)$-detectable and isolable.

The set of trajectories $\mathcal{J}$ is said to be $(\delta_d, \delta_m)$-diagnosable if and only if all the faults in $\mathbb{F}$ are $(\delta_d, \delta_m)$-diagnosable with respect to $\mathcal{J}$. In particular, if $\mathcal{J}(H)$ is $(\delta_d, \delta_m)$-diagnosable, we also say that the system $H$ is $(\delta_d, \delta_m)$-diagnosable (for the considered horizon).

In discussions above the discriminability of faults under consideration is confined to single fault. This can be easily extended to multiple-fault cases, since the discriminability is defined based on the $F_j$-faulty trajectory definition, which is readily modifiable by using and operators. For example, given $\mathbb{F} = \{F_1, F_2, F_3\}$ and a combination of faults $f = \{F_1, F_2\} \in 2^\mathbb{F}$, define $f$-faulty trajectories as both $F_1$-faulty and $F_2$-faulty, i.e., the intersection of the sets of $F_1$-faulty and $F_2$-faulty trajectories forms the set of $f$-faulty trajectories. Then multiple-fault discriminability can be defined and analyzed. For simplicity we consider the single-fault case.

**Proposition 4.** Given $H$ and $L^f$, the following statements hold:

1. If $H$ is $(\delta_d, \delta_m)$-diagnosable, then it is $(\delta_d^*, \delta_m^*)$-diagnosable for all $\delta_d^* \geq \delta_d$.
2. If $H$ is not $(\delta_d, \delta_m)$-diagnosable, then it is not $(\delta_d^*, \delta_m^*)$-diagnosable for all $\delta_d^* \leq \delta_d$.
3. If $H$ is $(\delta_d, \delta_m)$-diagnosable, then it is $(\delta_d^*, \delta_m^*)$-diagnosable for all $\delta_m^* \leq \delta_m$.
4. If $H$ is not $(\delta_d, \delta_m)$-diagnosable, then it is not $(\delta_d^*, \delta_m^*)$-diagnosable for all $\delta_m^* \geq \delta_m$.

Proof: Straightforward by Definition 13.

**C. Comparison of Diagnosability Notions**

The classical DES diagnosability notion [10] relies on event sequences. Extended to a multiple-fault version in [28], it states that the DES is diagnosable if and only if for all faults $F_j \in \mathbb{F}$, all $F_j$-faulty trajectories can be determined via the projections of prefixes whose length are uniformly bounded, that is, discriminated from trajectories that are not $F_j$-faulty. This uniform bound on event numbers gives the maximum delay for diagnosing $F_j$ since it first occurs, but the number is not prescribed explicitly in the definition.

The diagnosability definition for timed automata proposed by [12] is different from the DES diagnosability in some ways: Dense time instead of discrete time (counting events) is considered. The $F_j$-faulty and non-$F_j$-faulty trajectories (which refer to trajectories that are not $F_j$-faulty) are also required to be discriminable within a maximum delay by projections; but the maximum delay is explicitly prescribed in the notion of $\Delta$-diagnosability. So the fault $F_j$ has to be determined at most $\Delta$ (integral) time units later than its first occurrence. If the system is $\Delta$-diagnosable for some natural number $\Delta$, then it is called diagnosable.

Similar to the $\Delta$-diagnosability for timed automata, the $(\delta_d, \delta_m)$-diagnosability for hybrid automata prescribes a maximum delay $\delta_d$ in (real) time units as the diagnosis time window. So the delays $\Delta$ and $\delta_d$ are basically the same. The $(\delta_d, \delta_m)$-diagnosability additionally conveys the uncertainty $\delta_m$ in symptom measurements. In specific, a metric $d_{\delta_d}$ has been defined to compute the distance between the projected label sequences, and a distance greater than the measurement uncertainty $\delta_m$ ensures the diagnosability. By contrast, direct discrimination of different projected label sequences as in the DES and timed automata diagnosability corresponds to the case $\delta_m = 0$.

Moreover, the $(\delta_d, \delta_m)$-diagnosability differs in that it requires the discriminability of $F_j$-faulty trajectories from normal trajectories and other faulty trajectories rather than from non-$F_j$-faulty trajectories. Note that discriminability from non-$F_j$-faulty trajectories does not exclude other faults as the causes to alarms (for instance, a trajectory is both $F_j$-faulty and Faulty-faulty), although it implicitly excludes normality. As for the $(\delta_d, \delta_m)$-diagnosability, both detectability and isolability are involved.

**D. Bridging The Timed Language to $(\delta_d, \delta_m)$-Diagnosability**

The projected label sequences are the information available for fault diagnosis, while properties of the timed languages are what we can get from system abstraction. Thus a relation should be established between the metrics for projected label sequences and timed event sequences.

**Proposition 5.** Given $s_1, s_2 \in \Sigma^*$ as the label sequences produced by the timed event sequences $p_1, p_2 \in \chi(E)$, and $\pi_1 = \Pi(s_1), \pi_2 = \Pi(s_2)$, we have $d_{\delta_d}(\pi_1, \pi_2) \leq d_{\delta_d}(p_1, p_2)$.

Proof: If $d_{\delta_d}(p_1, p_2)$ is finite, then clearly $d_{\delta_d}(s_1, s_2) = d_{\delta_d}(p_1, p_2)$, and $s_1, s_2$ have the same output symbol sequence. According to the definition of $\Pi$, $\pi_1, \pi_2$ also must have the same output symbol sequence. Moreover, for any $\Delta_1^f, \Delta_1^s, \Delta_2^f, \Delta_2^s > 0$, $|\Delta_1^f| + |\Delta_2^f| - |\Delta_1^s| - |\Delta_2^s| \leq |\Delta_1^s - \Delta_2^s| + |\Delta_2^f - \Delta_1^f|$, therefore, by combining some labels’ dwell time distance after the projection, the supremum of dwell time distance over all the labels has become smaller. Thus, we obtain $d_{\delta_d}(\pi_1, \pi_2) \leq d_{\delta_d}(s_1, s_2)$.

For two sets of trajectories $\mathcal{J}(H), \hat{\mathcal{J}}(H)$ satisfying $\hat{h}(p(H), \hat{p}(H)) \leq \epsilon$, we now show there is a relation between their $(\delta_d, \delta_m)$-diagnosability.

**Lemma 1.** Given a faulty set $L^f$, and $\mathcal{J}(H), \hat{\mathcal{J}}(H)$ that satisfy $\hat{h}(p(H), \hat{p}(H)) \leq \epsilon$, if $\mathcal{J}(H)$ is not $(\delta_d, \delta_m)$-diagnosable, then $\hat{\mathcal{J}}(H)$ is not $(\delta_d^*, \delta_m^*)$-diagnosable.

Proof: For convenience, let $F_j(H) \subset \mathcal{J}(H), \hat{F}_j(H) \subset \hat{\mathcal{J}}(H)$ denote the set of $F_j$-faulty trajectories, and $F_j^s(H) \subset \hat{F}_j(H), \hat{F}_j^s(H) \subset \hat{F}_j(H)$ denote the set of $F_j$-faulty trajectories. Suppose $\mathcal{J}(H)$ is not $(\delta_d, \delta_m)$-diagnosable. This implies, by Definition 12 and 13, the existence of a fault $F_j$ and two trajectories $p_1, p_2 \in \mathcal{J}(H)$.

$$p_1 = (e_0, e_1, e_2, \ldots, e_i, e_i', e_i^f, t_i^f, \ldots, e_j, x_2, t_j),$$

$$p_2 = (e_0', e_1', e_2', \ldots, e_i', e_i^f, x_2', t_j'),$$
such that $\rho_1 \in F^d_\delta(H)$, $\rho_2$ is either normal or contained in $F_\rho(H)$ for some fault $F_i \neq F_j$, and their projected label sequences satisfy $d_\star((\pi_1, \tau_1)) \leq \delta_m$.

The timed event sequences produced by $\rho_1, \rho_2$ are in $P(H)$ by definition. Denote them as $\rho_1 = \{(\ell_1^i, \tilde{\tau}_1^i)\}_{i=0}^{N_1}, \rho_2 = \{(\ell_2^j, \tilde{\tau}_2^j)\}_{i=0}^{N_2}$ respectively. Since $\rho_1 \in F^d_\delta(H)$, then $\sum_{i=1}^{N_1} \tilde{\tau}_1^i \geq \delta_d$.

Since $\tilde{h}(P(H), \tilde{H}(H)) \leq \epsilon$, there exist two trajectories $\rho_1, \rho_2$ in $\tilde{F}_\delta(H)$, $\rho_2$ is either normal or contained in $F_i(H)$ for some fault $F_i \neq F_j$, and their projected label sequences satisfy $d_\star((\pi_1, \tau_1)) \leq \delta_m + 2\epsilon$.

By definition, $\rho_1 \in F^d_\delta(H)$ with $\delta = \delta_d + 2\epsilon$, and thus $\tilde{H}(H)$ is not $(\delta_d, \delta_m)$-diagnosable.

**Lemma 2.** Given a faulty set $L^f$ and $\tilde{H}(H), \tilde{F}(H)$ that satisfy $\tilde{h}(P(H), \tilde{P}(H)) \leq \epsilon$, if $\tilde{H}(H)$ is $(\delta_d, \delta_m)$-diagnosable, and $\delta_m \geq 2\epsilon$, then $\tilde{F}(H)$ is $(\delta_d, \delta_m)$-diagnosable.

Proof: Directly follow from Lemma 1.

**Theorem 1.** Given a faulty set $L^f$ and $\tilde{H}(H), \tilde{F}(H)$ that satisfy $\tilde{h}(P(H), \tilde{P}(H)) \leq \epsilon$, if $\tilde{H}(H)$ is not $(\delta_d, \delta_m)$-diagnosable, then $\tilde{F}(H)$ is not $(\delta_d, \delta_m)$-diagnosable; if $\tilde{F}(H)$ is $(\delta_d, \delta_m)$-diagnosable, and $\delta_m \geq 2\epsilon$, then $\tilde{H}(H)$ is $(\delta_d, \delta_m)$-diagnosable.

Proof: Directly follow from Lemma 1 and 2.

### E. Diagnosability Analysis Algorithm

To analyze whether $H$ is $(\delta_d, \delta_m)$-diagnosable, the following steps can be used:

1. Randomly simulate initial states for the horizon $[0, t_{end}]$ (or $N$) and compute the enlarged robust neighborhoods (or robust neighborhoods if possible) around them to fully cover $L^0 \times X^0$; then construct the system abstraction (see Section II-C).
2. It is guaranteed that $\tilde{h}(P(H), \tilde{P}(H)) \leq \epsilon$, where $\tilde{P}(H) = \tilde{P}(t_{end})$, and $P(H) = \tilde{P}(H)$. By the triangle inequality, $\tilde{H}(H)$ is $(\delta_d, \delta_m)$-diagnosable for finite horizon.
3. If $\tilde{h}(P(H), \tilde{P}(H)) \leq \epsilon$ holds as well, where $P(H) = \tilde{P}(t_{end})$, and $\tilde{P}(H) = \tilde{P}(t_{end}, H)$. We can use Algorithm 1 to verify the $(\delta_d, \delta_m)$-diagnosability of $H$ for finite horizon.

**Algorithm 1** Given $L^f$ and $\tilde{H}(H)$, verify if $\tilde{H}(H)$ is $(\delta_d, \delta_m)$-diagnosable.

1. **procedure** **VERIFICATION$(\delta_d, \delta_m, \epsilon)$**
2. compute $\delta_d \leftarrow \delta_d + \epsilon$, $\delta_m \leftarrow \delta_m + 2\epsilon$
3. **for** $j \leftarrow 1$ to $M$ **do**
4. compute $\tilde{F}_j(H)$, the set of $F_j$-faulty trajectories
5. compute $\tilde{P}_j(H)$, the set of $F_j$-faulty trajectories
6. **end for**
7. compute $\tilde{F}_0(H)$, the set of normal trajectories
8. **for** $j \leftarrow 1$ to $M$ **do**
9. for $i \leftarrow 0$ to $M$ **do**
10. if $i \neq j$ then
11. compute $\tilde{\delta}_{i,j} \leftarrow \inf_{\rho_1 \in \tilde{F}^d_\delta(H) \rho_2 \in \tilde{F}_j(H)} d_\star((\pi_1, \tau_2), (\pi_1, \tau_2))$
12. **end if**
13. **end for**
14. **end for**
15. compute $\tilde{\delta}_m \leftarrow \inf_{1 \leq j \leq M} \tilde{\delta}_{i,j}$
16. if $\tilde{\delta}_m < \delta_m$ then
17. $\tilde{H}(H)$ is $(\delta_d, \delta_m)$-diagnosable, $\tilde{H}(H)$ is $(\delta_d, \delta_m)$-diagnosable.
18. **end if**
19. **end procedure**

**F. Diagnosers**

The system abstraction constructed in Section II-C allows us to build a diagnoser for the system. The proposed approach to system abstraction and diagnoser construction is trajectory-based, which is disparate from other approaches in the literature. For instance, the work [29] proposed to construct a durational graph as an abstraction of the original system in order to analyze the diagnosability, which is fast but conservative, and the constructed durational graph requires further analysis of the timed language. In contrast, the trajectory-based abstraction in the present work can be arbitrarily precise, and the constructed diagnoser works in the following transparent way: It stores a finite list of candidate trajectories whose location sequences can be reached by $H$, and keeps narrowing down the list by observing the timed event sequences of $H$ till a decision is made.

We construct the diagnoser as a hybrid automaton $H_d = (L_d \times X_d, L_d \times X_d P_d, D_d, E_d, I_{N_d})$:

- Let $\{\tilde{\rho}_k\}^N_{k=1} \subset H(H)$ denote the set of (virtual) trajectories that extend to the end of the time horizon, where $\tilde{\rho}_k = \{(\ell_k, \tilde{\tau}_k, \tilde{\tau}_k, \tilde{\tau}_k)\}_{i=0}^{N_1}$. The state space of the diagnoser is defined as $L_d := \{0, 1, 2, \ldots\}$, $X_d := \mathbb{R}$. 0018-9286 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Example. A location $\ell_d \in L_d$ can be $({\{1,6\}},2)$. The diagnoser being at $\ell_d$ means that the location sequence reached by $\bar{\Delta}$ is on the guard of the first event triggered by the diagnoser. We write this event as $e_{d1,j} = (\ell_d,\ell_{d1,j},I_{d1,j},r)$.

Then we can model events $e_{d1,j} = (\ell_d,\ell_{d1,j},I_{d1,j},r)$ for all $j \leq z_1$ and $I_{d1,j}$, not empty, where $r(x) = 0$ for any $x \in R$.

By using the same way, events in $E_d$ can be modeled for other $\psi \in \{\psi(1),\ldots,\psi(o)\}$, and for other locations $\ell_d \in L_d$.

If and only if $H_d$ reaches a location $\ell_d$ such that $w_{(k,n)} = (F_j)$ for all $w_{(k,n)} \in w_{|\ell_d}$, the diagnoser raises an alarm for the fault $F_j$, which means $F_j$ is detected and isolated.

The diagnoser $H_d$ starts operating at the same time as $H$. The continuous system state $x_{d}$ is the time $d$. The discrete system state $\ell_d$ updates according to the triggered events $e_d \in E_d$. An event $e_d$ is triggered if and only if an output symbol $\psi$ with accumulated dwell time $\bar{\Delta}$ (since the preceding observed output symbol) is observed from $H$, such that $\psi$ is identical to the symbol that models $e_d$, and also $\bar{\Delta}$ satisfies the guard condition of $e_d$.

We assume the abstraction $\hat{J}(H)$ is $(\hat{\delta}_d,\hat{\delta}_m)$-diagnosable, $\hat{h}(\mathcal{P}(H),\mathcal{P}(H)) \leq \epsilon$, and $\delta_m \geq 2\epsilon$. By Theorem 1, $H$ is $(\hat{\delta}_{d+1},\delta_m - 2\epsilon)$-diagnosable. We thus assume that the measurement uncertainty of time intervals for $H$ is at most $(\delta_m - 2\epsilon)$. Specifically, any measured time interval must lie inside a ball of radius $(\frac{1}{2}\delta_m - \epsilon)$ centered at the true value.

Proposition 6. If the trajectory of $H$, $\rho \in \hat{J}(H)$, is $F_j\delta$-faulty, then the F1 fault alarm is already raised by the diagnoser $H_d$.

Proof: For any $\rho \in \hat{J}(H)$, there exists $\hat{\rho} \in \hat{J}(H)$ such that the produced timed event sequences satisfy $d_{\rho}(\rho,\hat{\rho}) \leq \epsilon$.

Hence, $\rho$ is $F_j\delta$-faulty implies that $\hat{\rho}$ is $F_j\delta$-faulty. By Prop. 5, we have $d_{\rho}(\pi,\hat{\rho}) \leq \epsilon$, where $\pi,\hat{\rho}$ are the projected label sequences produced by $\rho,\hat{\rho}$. Let $\bar{\rho}$ be the measurement of $\pi$. Then $d_{\rho}(\bar{\rho}) \leq d_{\rho}(\pi,\hat{\rho}) + d_{\rho}(\bar{\rho},\hat{\rho}) \leq \frac{3}{2}\delta_m$.

Since $\hat{J}(H)$ is $(\hat{\delta}_{d+1},\delta_m)$-diagnosable, for any $\bar{\rho}$ that is normal or $F_j$-faulty ($F_j \neq F_j$), the projected label sequences produced by $\hat{\rho}$ must satisfy $d_{\rho}(\pi,\hat{\rho}) > \frac{2}{3}\delta_m$. Thus $d_{\rho}(\bar{\rho}) \leq \frac{1}{3}\delta_m$. Namely, the measurement of the projected label sequence of $\rho$ must be at least $\frac{1}{3}\delta_m$ away from the projected label sequence of any normal or $F_j$-faulty trajectory.

Suppose $w_{(k,n)} \neq \{F_j\}$ for some $(k,n) \in \ell_d$, where $\ell_d = (k,n)$ is the last discrete state of the diagnoser updated according to $\pi$. By the construction of $H_d$, there must exist $\rho$ as a sub-trajectory of $\rho_k$, such that $\rho$ is normal or $F_j$-faulty ($F_j \neq F_j$), and its projected label sequence satisfies $d_{\rho}(\pi,\hat{\rho}) \leq \frac{1}{3}\delta_m$.

By contradiction, the fault alarm for $F_j$ should be raised no later than the last discrete state reached by the diagnoser according to $\pi$.

Proposition 7. If the F1 fault alarm is raised by the diagnoser for the trajectory $\rho \in \hat{J}(H)$, then $\rho$ must be $F_j$-faulty.

Proof: For any $\rho \in \hat{J}(H)$, there exists $\hat{\rho} \in \hat{J}(H)$ such that the produced timed event sequences satisfy $d_{\rho}(\rho,\hat{\rho}) \leq \epsilon$. Let $\rho_k \in \hat{J}(H)$ be a (virtual) trajectory extending to the end of the time horizon such that $\rho$ is a sub-trajectory of $\rho_k$. The label sequence produced by $\rho_k$ is $s_k = \{\{\hat{\Delta}_k,\psi_k\}_{k=0}^{N_k}\}$. The diagnoser operates according to the measurement $\bar{\rho}$ of the projected label sequence $\pi$ produced by $\rho$. By the construction of $H_d$, clearly $(k,n) \in \ell_d$, where $\ell_d = (k,n)$ denotes the $(n+1)$th location reached by $H_d$ during its operation. Let $\ell_{k,n}$ be the first observable event in $E$ triggered by $\rho_k$ after the starting signal, whose associated output symbol is $\psi_k \in \Psi_o$, and the accumulated dwell time since the initialization is $\bar{\Delta} := \cup_{k=0}^{N_k} \Delta_k$.

The projected label sequence $\pi$ produced by $\rho_k$ as a sequence $(0,\ell_{1,1},\{(\Delta,\psi)\}_{(2,\ell_{2,2})})$, and the measurement $\pi$ of $\tau$ as $(0,\ell_{1,1},\{(\Delta,\psi)\},\pi)$, then $\psi = \psi_k$, and $\Delta_k$ is on the guard of the first event triggered by the diagnoser. We write this event as $e_{d1,j} = (\ell_{d1,j},I_{d1,j},r)$.
like in the construction of \( E_d \). Accordingly, \( \psi_d(1) \) corresponds to \( \psi(1) \) in the construction of \( e_{d1,j} \) and \( \ell_d = \ell_d^p = \ell_d^f = \ell_d^0 \). According to the second property of guards in the construction of \( e_{d1,j} \), if \( \Delta \in [\Delta]_1 \setminus [\Delta]_j \), then \( \Delta \notin B(\Delta, \frac{1}{3} \bar{d}_m) \). We already have \( \Delta = \sum_i \lambda_i \Delta_i \in [\Delta]_1 \) by definition. Moreover, it follows from \( d_{\ell^*}(\pi, \pi) \leq \frac{2}{3} \bar{d}_m \) (see the proof of Prop. 6) that \( \Delta \in B(\Delta, \frac{1}{3} \bar{d}_m) \). Therefore, \( \Delta \in [\Delta]_1 \), \( (k, 1) \in \ell_{d1,j} = \ell_d^0 \). With similar argument, it can be proved that for any \( (\Delta, \psi) \) output by the \( n^{th} \) observable event of \( \rho \) triggered after the starting signal, the updated location of the diagnoser must contain \( (k, n) \). If \( H_d \) reaches a location whose fault labels are all \( \{ F_1 \} \), which implies \( \psi(k, n) = \{ F_1 \} \), then clearly \( \hat{\rho} \) is \( F_1 \)-faulty, and thus \( \rho \) is \( F_1 \)-faulty.

G. Numerical Example of Diagnosability Verification

Consider a 2-dimensional hybrid automaton \( H \) with 9 locations. The invariant sets and dynamics are visualized in Figure 2. Guards are boundaries of the invariant sets. Reset map for continuous state is the identity matrix. Some trajectories initiated from the set \( \{(x_1, x_2) | 2.2 \leq x_1 \leq 2.3, 2.2 \leq x_2 \leq 2.3\} \) are simulated for the time horizon \([0, 1.5]\) as shown in Figure 3. All the events in \( E \) has the unified output symbol \( \alpha \). The location \( \ell_8 \) is designated as faulty. The initial set leads to both normal and faulty trajectories.

We construct the system abstraction as in Section II-C by covering the initial set with the enlarged robust neighborhoods computed for 11 simulated initial states, where the parameter \( \epsilon \) in relative time metric is set to 0.1 (see Figure 4).

Since \( \hat{\mathcal{J}}(H) \) contains a finite number of simulated trajectories, it is easy to verify that 5 trajectories are normal (reaching \( \ell_7, \ell_4, \ell_1 \)), while 6 trajectories are faulty (reaching \( \ell_7, \ell_8, \ell_9 \)). There are two simulated trajectories

\[
\rho_1 = (e^3, 7, [2.2062, 2.2166]^T, 0.1635), \quad (e^3, 4, [2, 2.0095]^T, 1.1552), \quad (e^3, 1, [1, 2.5095]^T, 0.1813),
\]

\[
\rho_2 = (e^3, 7, [2, 2.964, 2.2062]^T, 0.1635), \quad (e^3, 8, [2.0818, 2]^T, 0.3466), \quad (e^5, 9, [2.5630, 1]^T, 0.9899),
\]

respectively corresponding to the projected label sequences \( \pi_1 = (0, t), (0.1635, \alpha), (1.1552, \alpha) \) and \( \pi_2 = (0, t), (0.1635, \alpha), (0.3466, \alpha) \). Hence, although \( \rho_2 \) goes faulty to \( \ell_8 \), it is not possible to tell it immediately apart from the normal \( \rho_1 \) by monitoring only the first two output symbols and the associated time intervals. Instead, we need to wait longer. With Algorithm 1 it can be verified that \( \hat{\mathcal{J}}(H) \) is \( (0.3466, 0.5384) \)-diagnosable, and \( \mathcal{J}(H) \) is \( (0.4236, 0.3384) \)-diagnosable. Therefore, by observing the output symbols and measuring the time intervals in between the measurement uncertainty 0.3384 in relative time metric, any trajectory of \( H \) that enters \( \ell_8 \) as the second location at the time instant \( t^f \) can be diagnosed before the time window \([t^f, t^f + 0.4236]\) runs out.

IV. CONCLUSION

In this paper, we propose a diagnosability notion for hybrid automata that conveys the maximum delay in detecting and isolating faults under given measurement uncertainty in time intervals. We prove a quantitative relation on such \((\delta_d, \delta_m)\)-diagnosability of a hybrid automaton and its system abstraction, whose timed language approximately includes or is approximately equivalent to that of the original system. By analyzing the \((\delta_d, \delta_m)\)-diagnosability of the system abstraction especially when it has only finitely many trajectories extending to the time horizon of interest, diagnosability analysis and diagnoser construction of the original system can be...
simplified to a great extent. We present a method to construct such
system abstractions by using the robust test generation and coverage
idea, which can be implemented automatically with the Matlab
toolbox STRONG [30] for linear systems. An example is illustrated,
reducing the finite-horizon diagnosability analysis of a system with
infinitely many trajectories to finite.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>hybrid automaton</td>
</tr>
<tr>
<td>$L \times X$</td>
<td>state space</td>
</tr>
<tr>
<td>$D$</td>
<td>dynamics</td>
</tr>
<tr>
<td>$L^0 \times X^0$</td>
<td>initial set</td>
</tr>
<tr>
<td>$E$</td>
<td>events</td>
</tr>
<tr>
<td>$I_{nv}$</td>
<td>invariant sets</td>
</tr>
<tr>
<td>$\ell$</td>
<td>location (discrete state)</td>
</tr>
<tr>
<td>$x$</td>
<td>continuous state</td>
</tr>
<tr>
<td>$e$</td>
<td>event</td>
</tr>
<tr>
<td>$r$</td>
<td>reset map</td>
</tr>
<tr>
<td>$g$</td>
<td>guard</td>
</tr>
<tr>
<td>$G_\ell$</td>
<td>guards (in $\ell \in L$)</td>
</tr>
<tr>
<td>$\xi_\ell(t, x^0_\ell)$</td>
<td>dynamical system solution</td>
</tr>
<tr>
<td>$\rho$</td>
<td>trajectory</td>
</tr>
<tr>
<td>$J$</td>
<td>a set of trajectories</td>
</tr>
<tr>
<td>$p$</td>
<td>timed event sequence</td>
</tr>
<tr>
<td>$P$</td>
<td>timed language</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>dwell time</td>
</tr>
<tr>
<td>$\psi$</td>
<td>output symbol</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>real numbers</td>
</tr>
<tr>
<td>$\Psi_o$</td>
<td>observable output symbols</td>
</tr>
<tr>
<td>$\emptyset$</td>
<td>unobservable output symbol</td>
</tr>
<tr>
<td>$\iota$</td>
<td>starting signal</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>labels</td>
</tr>
<tr>
<td>$s$</td>
<td>label sequence</td>
</tr>
<tr>
<td>$J$</td>
<td>trajectories of interest</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>timed language produced by $J$</td>
</tr>
<tr>
<td>$\Pi$</td>
<td>projection map</td>
</tr>
<tr>
<td>$\pi$</td>
<td>projected label sequence</td>
</tr>
<tr>
<td>$L^f$</td>
<td>faulty set</td>
</tr>
<tr>
<td>$F_j$</td>
<td>a type of fault</td>
</tr>
<tr>
<td>$\delta_d$</td>
<td>delay parameter</td>
</tr>
<tr>
<td>$\delta_m$</td>
<td>measurement uncertainty</td>
</tr>
</tbody>
</table>
REFERENCES


Alessandro D’Innocenzo is Assistant Professor in the Department of Information Engineering, Computer Science and Mathematics at the University of L’Aquila. In 2005 he was recipient of Fondazione Filauro award for PhD students. In 2007 he obtained the PhD degree in Electrical and Information Engineering from University of L’Aquila, and accomplished the International Curriculum Option of Doctoral Studies in Hybrid Control for Complex, Distributed and Heterogeneous Embedded Systems. In 2007 and 2009 he has been Postdoctoral Researcher in the Department of Electrical and Information Engineering of University of L’Aquila. In 2008 he has been Postdoctoral Researcher in the Department of Electrical and Systems Engineering of University of Pennsylvania. His research focuses on control theory and in particular, hybrid systems, formal verification and networked control, with applications to air traffic management, building automation and communication systems.
Maria Domenica Di Benedetto obtained the "Dr. Ing." degree (summa cum laude) of Electrical Engineering and Computer Science, University of Roma "La Sapienza", in 1976. In 1981, she obtained the degree "Docteur-Ingenieur" and in 1987 the degree "Doctorat d'Etat Sciences", Université de Paris-Sud (Orsay, France). Since 1994, she has been Professor of Control Theory at University of l’Aquila. Since 2002, she has been IEEE Fellow. She is the PI and Director of the Center of Excellence for Research DEWS "Architectures and Design methodologies for Embedded controllers, Wireless interconnect and System-on-Chip". Since 1995, she is Member of the Scientific Committee of the Center of Excellence for Research CETEMPS, University of L’Aquila. She is co-founder and member of the Governing Board of WEST Aquila S.r.L.. She is 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 obtained 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. 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 visiting professor at the Laboratoire des Signaux et Systemes, CNRS–Paris, of the Department of Electrical Engineering of the Princeton University and of the Department of Electrical Engineering and Computer Science of the University of California at Berkeley, USA, 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.

A. Agung Julius received the Ph.D. degree in applied mathematics from the University of Twente, The Netherlands, in 2005. He joined the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, as an Assistant Professor in December 2008. From 2005 to 2008, he was a Postdoctoral Researcher at the University of Pennsylvania. His research interests include systems and control, systems biology, stochastic models in systems biology, control of biological systems, hybrid systems, and mathematical systems theory. Dr. Julius received an NSF CAREER award in 2010.