INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2009; **19**:1564–1580
Published online 27 March 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc.1432

Discrete state observability of hybrid systems

Maria D. Di Benedetto[‡], Stefano Di Gennaro[§] and Alessandro D'Innocenzo[*,†]

Department of Electrical and Information Engineering, Center of Excellence DEWS, University of L'Aquila,
Poggio di Roio, 67040 L'Aquila, Italy

SUMMARY

We propose a novel definition of observability, motivated by safety critical applications, given with respect to a subset of *critical* discrete states that model unsafe or unallowed behaviors. For the class of discrete event systems, we address the problem in the setting of formal (regular) languages and propose a novel observability verification algorithm. For the class of switching systems, we characterize the minimal set of extra output information to be provided by the continuous signals in order to satisfy observability conditions, and propose a milder observability notion that allows a bounded delay in state observation. For the class of hidden Markov models, we analyze decidability and complexity of the verification problem. Copyright © 2009 John Wiley & Sons, Ltd.

Received 21 December 2007; Revised 13 November 2008; Accepted 9 December 2008

KEY WORDS: observability; discrete event system; switching system; hidden Markov model; automatic verification; computational complexity

1. INTRODUCTION

In many safety critical applications, e.g. in air traffic management procedures [1–3], it is often required to detect if the current behavior of the system is associated with a dangerous or unallowed operation. Estimation methods and observer design techniques are essential in this regard, for the design of a control strategy for error propagation avoidance and/or error recovery. Discrete event and hybrid systems are a powerful tool for the analysis and control of multi-agent systems, since it is convenient to model undesired or dangerous behaviors by means of discrete states that we call *critical* states. Then, the possibility of detecting dangerous situations depends on the observability properties of the system with respect to the critical states.

[*] Correspondence to: Alessandro D'Innocenzo, Department of Electrical and Information Engineering, Center of Excellence DEWS, University of L'Aquila, Poggio di Roio, 67040 L'Aquila, Italy.

[†] E-mail: alessandro.dinnocenzo@ing.univaq.it

[‡] E-mail: mariadomenica.dibenedetto@univaq.it

[§] E-mail: stefano.digennaro@univaq.it

Contract/grant sponsor: European Commission under Project IST NoE HyCON; contract/grant number: 511368
Contract/grant sponsor: European Commission under Project iFly; contract/grant number: TREN/07/FP6AE/S07.71574/037180

Various notions of observability have been introduced in the literature for discrete event systems [4–8] and hybrid systems [2, 9–13]. We focus in this paper on the observability of the discrete state, and propose a definition of observability with respect to a subset of discrete *critical* states. We first formulate our problem in the setting of discrete event systems, then we extend our results to switching systems and hidden Markov models.

We first consider discrete event systems and propose our definition of discrete state observability. Observability conditions can be checked on the structure of the discrete state observer [2, 4, 5, 9], which can be constructed in exponential time with respect to the cardinality of the discrete state space: this implies that the complexity of the verification algorithm is exponential as well. We address the observability verification problem in the setting of formal (regular) languages [14], and propose a new verification algorithm, executable in *polynomial* time, which exploits properties of operations on regular languages. The main contribution with respect to the results of [9] consists in the analysis of the computational complexity for the observability verification. We prove that our observability conditions can be checked efficiently in polynomial time, instead of exponential time. Moreover, our algorithms provide (i) the minimum number of steps K after which the critical states can be *observed* and (ii) the minimum set of the extra signals needed to satisfy the observability conditions.

We then consider a subclass of hybrid systems, called switching systems, where a continuous dynamical system is associated with each discrete state. When the information given by the discrete output are not sufficient to build an observer, the continuous dynamics can be exploited as proposed in [9] to generate some discrete signals that provide additional information useful to discriminate the discrete states. This can be done by using fault detection techniques [15, 16], as for example in [9, 17] where a bank of Luenberger observers is used to identify the discrete state. However, the choice of the extra signals needed to satisfy the observability conditions is not unique. We propose an algorithm to compute the *minimum* extra information needed to achieve observability. Since the generation of these extra output symbols requires a nonzero generation time, a milder notion of observability, which allows a bounded delay in the observation, and a verification algorithm are proposed.

Finally, we consider hidden Markov models. We propose an observability definition similar to that given in [18] for the continuous states of jump linear systems, which allows a bound in the probability of estimation uncertainty. As one of the main results of the paper, we show that the addressed observability verification problem is decidable, and we characterize its computational complexity.

The organization of the paper is as follows. In Section 2, we analyze discrete state observability for discrete event systems. In Section 3, we extend our results to switching systems. In Section 4, we address the observability verification problem for hidden Markov models. In Section 5, an illustrative example is presented. Finally, in Section 6, we offer some concluding remarks.

2. DISCRETE EVENT SYSTEMS

In this section we propose a formal definition of observability of a subset of discrete states for discrete event systems. We analyze the verification problem using the discrete output of the system and propose a novel verification procedure that can be executed in polynomial time.

Definition 1 (*Discrete event system*)
A discrete event system is a tuple $\mathscr{D}=(Q,Q_0,E,\Psi,\eta)$ such that

- Q is a finite set of N discrete states.
- $Q_0\subseteq Q$ is the set of initial conditions.
- $E\subseteq Q\times Q$ is a collection of edges; each edge $e\in E$ is an ordered pair of discrete states, the first component of which is the source and is denoted by $s(e)$, while the second is the target and is denoted by $t(e)$.
- Ψ is the finite set of discrete output symbols. It includes the empty string ε that corresponds to unobservable output.
- $\eta:E\rightarrow\Psi$ is the output function, that associates with each edge a discrete output symbol.

The executions of discrete states of $\mathscr{D}$ are the sequences $\rho=\{q_k\}_{k=0}^{|\rho|}$ such that $q_0\in Q_0$, $(q_k,q_{k+1})\in E, k=0,1,\ldots,|\rho|-1$, with $|\rho|\geqslant 0$ the length of the execution.

From this definition, it is not possible that a system has two edges e_1, e_2 with the same source $s(e_1)=s(e_2)$ and target $t(e_1)=t(e_2)$. There is no loss of generality since it is always possible to construct an equivalent system that complies our model by splitting the source or the target state, where 'splitting' a state q_i means creating two states q_i', q_i'', keeping the incoming and outgoing edges.

Definition 2 (*Formal language of executions*)
The formal language of the executions of discrete states of $\mathscr{D}$ is given by

$$\mathscr{L}\triangleq\{\rho=\{q_k\}_{k=0}^{|\rho|}:q_0\in Q_0,(q_k,q_{k+1})\in E,$$

$$k=0,1,\ldots,|\rho|-1\}$$

Given a subset of discrete states $Q^*\subseteq Q$, we define

$$\mathscr{L}_{Q^*}\triangleq\{\rho\in\mathscr{L}:|\rho|<\infty,q_{|\rho|}\in Q^*\}$$

the language of executions with finite cardinality, such that the last visited discrete state belongs to Q^*. For $q\in Q$, we use for simplicity the notation $\mathscr{L}_q$ instead of $\mathscr{L}_{\{q\}}$. Given an execution $\rho=\{q_k\}_{k=0}^{|\rho|}$, the associated output string is $\{\eta((q_k,q_{k+1}))\}_{k=0}^{|\rho|-1}$. The associated *observation* $P(\rho)$ is obtained erasing all unobservable outputs from the output string.

Definition 3 (*Formal language of observations*)
The formal language of the observations of $\mathscr{D}$ is given by

$$\mathscr{P}\triangleq\{P(\rho):\rho\in\mathscr{L}\}$$

Given a subset of discrete states $Q^*\subseteq Q$, we define $\mathscr{P}_{Q^*}$ the language of the observations generated by executions whose last visited state belongs to Q^*

$$\mathscr{P}_{Q^*}\triangleq\{P(\rho):\rho\in\mathscr{L}_{Q^*}\}$$

Since two distinct executions can generate the same observation, the intersection set $\mathscr{P}_{Q_1}\cap\mathscr{P}_{Q_2}$ is not necessarily empty for $Q_1\cap Q_2=\emptyset$. This is a crucial issue for observability of the discrete state, as we will show in the following.

DOI: 10.1002/rnc

Let $Q_c \subset Q$ be the set of *critical states* of $\mathscr{D}$, i.e. the set of discrete states associated with unsafe or unallowed behaviors of $\mathscr{D}$. We say that Q_c is observable for $\mathscr{D}$ if it is possible to construct a system that, on the basis of the observations, is able to detect whether the current discrete state of $\mathscr{D}$ belongs to Q_c or not. A necessary and sufficient condition can be given in terms of observations.

Definition 4
Given a discrete event system $\mathscr{D}$, the set Q_c is observable if and only if

$$\mathscr{P}_{Q_c} \cap \mathscr{P}_{Q \setminus Q_c} = \emptyset \tag{1}$$

Intuitively, each observation can be generated either only by executions whose last visited state belongs to Q_c, or only by executions whose last visited state does not belong to Q_c.

In the following we address the observability verification problem in the setting of *regular languages* [14]. Given a discrete event system $\mathscr{D} = (Q, Q_0, E, \Psi, \eta)$, one of the algorithms proposed in [2, 4, 5, 9] can be used to construct the discrete state observer $\mathscr{O}_{Q_c} = (\hat{Q} \subseteq 2^Q, \hat{q}_0 = \{Q_0\}, \hat{Q}_c, \hat{E}, \hat{\Psi} = \Psi \setminus \{\varepsilon\}, \hat{\eta})$. $\mathscr{O}_{Q_c}$ is a deterministic finite automaton (DFA), where each discrete state $\hat{q} \in \hat{Q}$ is a subset of Q and the final set

$$\hat{Q}_c \triangleq \{\hat{q} \in \hat{Q} : \hat{q} \cap Q_c \neq \emptyset \wedge \hat{q} \cap Q \setminus Q_c \neq \emptyset\}$$

is induced by the critical set Q_c. The definitions of *nondeterministic finite automaton* (NFA), DFA, *regular language*, and an algorithm to construct the discrete state observer $\mathscr{O}_{Q_c}$ are recalled in the Appendix.

The DFA $\mathscr{O}_{Q_c}$ accepts the language $\mathscr{P}_{Q_c} \cap \mathscr{P}_{Q \setminus Q_c}$ and it is therefore possible to verify observability conditions directly on $\mathscr{O}_{Q_c}$ checking if the accepted language is empty, i.e. if $\hat{Q}_c = \emptyset$. Hence, the observability verification can be done in time exponential in $N = |Q|$ by constructing the observer. However, there exists an NFA having a discrete state space cardinality polynomial in N, which accepts the same language as $\mathscr{O}_{Q_c}$. This implies that it is possible to construct an observer that consists of a set of concurrent DFAs, and whose output is given by a logical operation on the outputs of the DFAs. We exploit this property of regular languages to define an observability verification procedure that can be executed in time polynomial in N, on a discrete event system $\mathscr{D}$. The main idea of the algorithm is to use operations on regular languages to check condition (1) without constructing the observer.

Algorithm 1
Given a discrete event system $\mathscr{D}$ and a critical set Q_c

1. Construct the NFA $\mathscr{N}_{Q_c}$ that accepts $\mathscr{P}_{Q_c}$.
2. Construct the NFA $\mathscr{N}_{Q \setminus Q_c}$ that accepts $\mathscr{P}_{Q \setminus Q_c}$.
3. Construct the NFA $\mathscr{N}_\cap$ that accepts $\mathscr{P}_{Q_c} \cap \mathscr{P}_{Q \setminus Q_c}$.
4. Q_c is observable for $\mathscr{D}$ if and only if the language accepted by $\mathscr{N}_\cap$ is empty.

Theorem 1
Algorithm 1 can be executed in $O(N^4)$.

Proof
The first and second steps require N^2 iterations each, since $\mathscr{P}_{Q_c}$, $\mathscr{P}_{Q \setminus Q_c}$ are finite unions of the regular languages $|Q_c|$, $|Q \setminus Q_c|$, respectively. The third step requires N^4 iterations, since the intersection of the two regular languages $\mathscr{P}_{Q_c}$, $\mathscr{P}_{Q \setminus Q_c}$ is accepted by a NFA with state space

DOI: 10.1002/rnc

cardinality $N^2 \times N^2$. The last step can be executed during step 3. Hence, the overall complexity is given by $2N^2 + N^4 \sim O(N^4)$. □

The previous result can be extended to the case of state observability after a transient of K transitions.

Definition 5
Given a discrete event system $\mathscr{D}$, the set Q_c is observable in K-steps if and only if

$$\forall \rho : P(\rho) \in \mathscr{P}_{Q_c} \cap \mathscr{P}_{Q \setminus Q_c}, \quad |\rho| < K \tag{2}$$

In order to verify condition (2), Algorithm 1 can be used with line 4 replaced by:

4′. Q_c is observable in K-steps for $\mathscr{D}$ if and only if the final states of $\mathscr{N}_\cap$ can only be reached by finite paths that contain less than K transitions.

The minimum value $K_{\min}$ such that Q_c is observable in $K_{\min}$-steps can be computed in polynomial time by searching for the maximum length of all paths that reach a final state of the system $\mathscr{N}_\cap$.

3. SWITCHING SYSTEMS

In this section we extend our results to a subclass of hybrid systems, called switching systems, where a continuous dynamical system is associated with each discrete state. When the information given by the discrete output are not sufficient to build an observer, we provide an algorithm to compute the minimum set of extra information we need in order to make the system observable. These extra information are determined from the continuous input and output signals and cannot be generated instantaneously. We propose an algorithm to construct an abstract model that formalizes the generation of extra information by means of discrete output symbols. We then introduce a milder observability definition that allows bounded delay in the observation of the current discrete state and give a procedure to verify this property on the abstract system.

Definition 6 (*Switching system*)
A switching system is a tuple $\mathscr{S} = (\mathscr{D}, X, X_0, U, Y, \mathscr{E})$ such that:

- $\mathscr{D} = (Q, Q_0, E, \Psi, \eta)$ is a discrete event system as in Definition 1.
- $X \subseteq \mathbb{R}^n$ is the continuous state space.
- $X_0 \subseteq X$ is the set of initial continuous conditions.
- $U \subseteq \mathbb{R}^m$, $Y \subseteq \mathbb{R}^p$ are the sets of continuous control input and observable output.
- $\{\mathscr{E}_q\}_{q \in Q}$ associates with each discrete state $q \in Q$ the continuous time-invariant dynamics

$$\mathscr{E}_q : \dot{x} = f_q(x, u) \tag{3}$$

with output $y = g_q(x)$.

It is worth noting that a solution of Equation (3) exists and is unique under the assumption that f_q is continuous with respect to time and Lipschitz continuous with respect to x, and the control input is piecewise continuous from the right and with left limit.

This class of switching systems is nondeterministic, in general. The continuous state evolves following deterministic dynamics, and the discrete state performs nondeterministic transitions. We recall in the Appendix, the definitions of a *hybrid time basis* $\tau \triangleq \{I_k\}_{0 \leqslant k \leqslant |\tau|}$ with cardinality

$|\tau|$ and of a *hybrid execution* $\chi=(\tau,q,x)$. Let $\mathscr{X}$ be the set of all executions χ of $\mathscr{S}$. In this paper, we consider *nonblocking* switching systems, i.e. systems such that all hybrid executions are defined for all time instants [19]. We say that a hybrid execution is *Zeno* if it is characterized by an infinite number of jumps in a finite time [20]. We consider switching systems that do not generate Zeno executions.

To each execution $\chi=(\tau,q,x)\in\mathscr{X}$ we associate a unique string $\rho(\chi)$ as a sequence $\{q(I_k)\}_{k=0}^{|\tau|}$ with cardinality $|\rho(\chi)|=|\tau|$. Namely, $\rho(\chi)$ represents an execution of the discrete state of $\mathscr{S}$, with $q(I_k)$ the discrete state in the time interval I_k.

Definition 7 (*Formal language of executions*)
The formal language of the executions of discrete states of $\mathscr{S}$ is given by

$$\mathscr{L}\triangleq\{\rho(\chi):\chi=(\tau,q,x)\in\mathscr{X}\}$$

According to Definition 7, we can use the same notions of language of observations given in Section 2, and all previous results hold.

Given a switching system $\mathscr{S}$ whose discrete layer $\mathscr{D}$ does not satisfy the observability condition (1), following [9] we exploit the knowledge coming from the continuous dynamics to generate additional discrete signals that provide extra information to discriminate the discrete states. We define a partial function $h:Q\rightarrow\Psi_e$ that associates to some states $q\in Q$ an additional discrete output symbol $h(q)\in\Psi_e$. We say that (Ψ_e,h) is a solution if Q_c is observable, according to condition (1), for the system $\mathscr{S}$ augmented with the additional output. An optimal solution (Ψ_e^*,h^*), which is not necessarily unique, is a solution that minimizes the number $|\Psi_e^*|$ of extra discrete outputs that are necessary to achieve observability. This optimal solution can be computed in exponential time (with respect to the cardinality $|Q|$ of the discrete state space) using the following algorithm.

Algorithm 2
Given a switching system $\mathscr{S}$ and a critical set Q_c

1. Compute $\mathscr{N}_\cap$ applying Algorithm 1 to system $\mathscr{S}$.
2. For each set $\bar{Q}\in 2^Q$, searching with increasing cardinality of $\bar{Q}$, delete from $\mathscr{N}_\cap$ the discrete states (q_1,q_2) such that q_1, $q_2\in\bar{Q}$. If the language accepted by $\mathscr{N}_\cap$ is empty, then define $\Psi_e^*\triangleq\{\psi_q:q\in\bar{Q}\}$, $h^*(q)\triangleq\psi_q$ and exit.

Proposition 1
Given a switching system $\mathscr{S}$ and a critical set Q_c, the output of Algorithm 2 (Ψ_e^*,h^*) is an optimal solution.

Proof
If a state (q_1,q_2) belongs to the state space of the system $\mathscr{N}_\cap$, then there exist two executions $\rho_1\in\mathscr{L}_{q_1}$, $\rho_2\in\mathscr{L}_{q_2}$ such that $P(\rho_1)=P(\rho_2)$. Given any $q_1,q_2\in\bar{Q}$ it is clear that, using the extra outputs $h(q_1)$, $h(q_2)$, $h(q_1)\neq h(q_2)$, we obtain $P(\rho_1)\neq P(\rho_2)$. Moreover, all executions ρ_1', $\rho_2\prime$ that have ρ_1, ρ_2 as prefixes will satisfy $P(\rho_1')\neq P(\rho_2')$. For this reason, all observations of $\mathscr{N}_\cap$ generated passing through the state (q_1,q_2) will not generate ambiguity, and thus (q_1,q_2) can be deleted. When all states (q_1,q_2) such that $q_1,q_2\in\bar{Q}$ are deleted from $\mathscr{N}_\cap$ and if the obtained system accepts the empty language, then it follows that using the extra output defined by $\Psi_e^*\triangleq\{\psi_q:q\in\bar{Q}\}$, $h^*(q)\triangleq\psi_q$ satisfies observability conditions. Since Algorithm 2 performs a search on all sets of extra output signals $\bar{Q}\in 2^Q$, with increasing cardinality of $\bar{Q}$, then the output of the algorithm is

an optimal solution. Since the number $|Q|$ of discrete states is finite, Algorithm 2 is guaranteed to converge. Moreover, a solution (Ψ_e^*, h^*) is always defined at the end of the algorithm, since $\Psi_e^* = Q$, $h^*(q) = q$ is always a solution. □

A nonoptimal solution $(\Psi_e^\sharp, h^\sharp)$ can be computed in polynomial time as follows.

Algorithm 3
Given a switching system $\mathscr{S}$ and a critical set Q_c

1. For all $q_\mathrm{c} \in Q_\mathrm{c}$, initialize $Q_{q_\mathrm{c}} \triangleq \emptyset$.
2. Compute $\mathscr{N}_\cap = (Q^\cap, q_0^\cap, Q_f^\cap, E^\cap, \Psi^\cap, \eta^\cap)$ applying Algorithm 1 to system $\mathscr{S}$.
3. Given $(q_1, q_2) \in Q_f^\cap$, by definition, either $q_1 \in Q_\mathrm{c}$, $q_2 \notin Q_\mathrm{c}$, or $q_2 \in Q_\mathrm{c}$, $q_1 \notin Q_\mathrm{c}$. In the former case, add q_2 to Q_{q_1}, and in the latter case, add q_1 to Q_{q_2}.
4. For any $q_\mathrm{c} \in Q_\mathrm{c}$ and $q \in Q_{q_\mathrm{c}}$, define $\Psi_e^\sharp = \{\psi_q : q \in Q_\mathrm{c} \text{ or } q \in \bigcup_{q_\mathrm{c} \in Q_\mathrm{c}} Q_{q_\mathrm{c}}\}$, $h^\sharp(q) \triangleq \psi_q$.

Proposition 2
Given a switching system $\mathscr{S}$ and a critical set Q_c, the output of Algorithm 3 $(\Psi_e^\sharp, h^\sharp)$ is a solution.

Proof
Algorithm 3 performs a search only on extra output signals in a subset $2^{\bar{Q}} \subseteq 2^Q$, $\bar{Q} \subseteq Q$, namely only within the set of states that are indistinguishable from critical states. More formally, $\bar{Q} = \{q : \exists p \in \mathscr{P}_{Q_\mathrm{c}} \cap \mathscr{P}_{Q \setminus Q_\mathrm{c}}, \exists \rho \in \mathscr{L}_q : P(\rho) = p\}$. Since the number $|Q|$ of discrete states is finite, Algorithm 3 is guaranteed to converge. Since Algorithm 2 performs a search on a subset of extra output signals, a solution is not always defined at the end of the algorithm. Even if Algorithm 3 fails to find a solution, a solution may exist. □

It can happen that a solution (Ψ_e, h) obtained using the algorithms above is not achievable, in the sense that we may not be able to generate the extra signals for all discrete states, or different discrete states may have 'similar' continuous dynamics (namely if $\mathscr{E}_{q_i} = \mathscr{E}_{q_j}$, $q_i \neq q_j$, then $h(q_i) = h(q_j)$). Moreover, even if the solution is achievable, we have to take into account the time needed for the generation of each signal $h(q)$ by using the continuous dynamics. For example, in [9] where a bank of Luenberger observers is used for the generation of extra outputs, this time depends on the gain matrices of the observers. If the generation time in state q, denoted $\delta_{h(q)}$, is nonzero (which is almost always the case), then Q_c may be not observable in the sense of Definition 4. As a consequence, we introduce a milder definition of observability that requires a bounded delay in the observation of a critical state.

Definition 8 (*Observer with bounded delay*)
Given a switching system $\mathscr{S}$, an observer with delay δ of the critical set Q_c is a system $\mathscr{O}_{Q_\mathrm{c}}^\delta$ whose input is the output of $\mathscr{S}$, and whose output $\hat{y}(t)$ is such that

$$\forall k \geqslant 0 \quad \forall t \in [t_k + \delta, t_k']$$
$$\hat{y}(t) = \begin{cases} 1 & \text{if } q(I_k) \in Q_\mathrm{c} \\ 0 & \text{if } q(I_k) \notin Q_\mathrm{c} \end{cases}$$

A set Q_c is said to be observable with delay δ for $\mathscr{S}$ if and only if an observer $\mathscr{O}_{Q_\mathrm{c}}^\delta$ exists.

If Q_c is observable with delay $\delta^* \geqslant 0$, then it is observable with delay δ for any $\delta > \delta^*$. We define $\delta_{\min}$ as the minimum value such that Q_c is observable with delay $\delta_{\min}$. It is clear that the value $\delta_{\min}$ depends on the structure of the system and on the generation times $\delta_{h(q)}$ of extra outputs.

In order to verify if the additional information obtained by (Ψ_e, h) are sufficient to satisfy the observability condition with delay, we give an algorithm for constructing a system $\tilde{\mathscr{S}}$ that formalizes the generation of extra discrete output symbols. This algorithm uses the notions of minimum $\Delta_m(q)$ and maximum $\Delta_M(q)$ dwell times, the definitions of which can be found in the Appendix, and is based on the following assumption.

Assumption 1
For each $q \in Q$, the generation time $\delta_{h(q)}$ is less than the minimum dwell time $\Delta_m(q)$, namely $h(q)$ is generated before any discrete transition from the discrete state q to a different discrete state takes place.

Algorithm 4
Given a switching system $\mathscr{S}$, construct a switching system $\tilde{\mathscr{S}}$ as follows. First, assign $\Psi \triangleq \Psi \cup \Psi_e$. Then, for each discrete state $q \in Q$ do

1.1. Replace each q by the discrete states q^1 and q^2.
1.2. For all $e \in E$ such that $t(e)=q$ assign $t(e) \triangleq q^1$, and for all $e \in E$ such that $s(e)=q$ assign $s(e) \triangleq q^2$.
1.3. Add $e_q \triangleq (q^1, q^2)$ to E and assign $\eta(e_q) \triangleq h(q)$.
1.4. Assign $\Delta_m(q^1) \triangleq \Delta_M(q^1) \triangleq \delta_{h(q)}$, $\Delta_m(q^2) \triangleq \Delta_m\ (q) - \delta_{h(q)}$ and $\Delta_M(q^2) \triangleq \Delta_M(q) - \delta_{h(q)}$.

The intuition of Algorithm 4 is illustrated in Figure 1. Assumption 1 implies that the executions of $\tilde{\mathscr{S}}$ are the same as those of $\mathscr{S}$, splitting the time bases intervals.

Proposition 3
For each execution $\chi = (\tau, q, x)$ of $\mathscr{S}$, there exists an execution $\tilde{\chi} = (\tilde{\tau}, \tilde{q}, \tilde{x})$ of $\tilde{\mathscr{S}}$ such that

1. let $\tau = \{I_k\}_{k=0}^{|\tau|}$, $I_k = [t_k, t'_k]$, then $\tilde{\tau} = \{I_k^1\}_{k=0}^{|\tau|} \cup \{I_k^2\}_{k=0}^{|\tau|}$, where $I_k^1 = [t_k, t_k + \delta_{h(q(I_k))}]$ and $I_k^2 = [t_k + \delta_{h(q(I_k))}, t'_k]$,
2. let $q(I_k) = q$, then $\tilde{q}(I_k^1) = q^1$, $\tilde{q}(I_k^2) = q^2$,
3. $x(t) = \tilde{x}(t)$, $\forall t \in \tau$,

and viceversa.

It is possible to verify observability with delay for $\mathscr{S}$ by checking the observability condition (1) for $\tilde{\mathscr{S}}$. Let Q and $\tilde{Q}$ be the discrete state spaces of $\mathscr{S}$ and $\tilde{\mathscr{S}}$, respectively, and let $\mathrm{suc}(q) \triangleq \{\bar{q} \in Q : \exists e \in E, s(e) = q, t(e) = \bar{q}\}$ be the set of successors of q.

Theorem 2
Given $\mathscr{S}$ and $\tilde{\mathscr{S}}$, Q_{c} is observable with delay δ for $\mathscr{S}$ if

1. The set $\tilde{Q}_{\mathrm{c}} \triangleq \bigcup_{q \in Q_{\mathrm{c}}} (q^2 \cup \mathrm{suc}(q^2))$ is observable for $\tilde{\mathscr{S}}$.
2. $\delta_{h(q)} \leqslant \delta$, $\forall q \in Q_{\mathrm{c}} \cup \mathrm{suc}(Q_{\mathrm{c}})$.

Proof
Define $\delta^* = \max_{q \in Q_{\mathrm{c}} \cup \mathrm{suc}(Q_{\mathrm{c}})} \delta_{h(q)}$, where $\delta^* \leqslant \delta$ by Condition 2. Condition 1 implies that there exists an observer $\tilde{\mathcal{O}}_{\tilde{Q}_{\mathrm{c}}}$ for $\tilde{\mathscr{S}}$ such that if $\tilde{q}(\tilde{I}_k) \in \tilde{Q}_{\mathrm{c}}$, then the observer's output $\hat{y}(t) = 1$ for all $t \in \tilde{I}_k$. By construction of $\tilde{\mathscr{S}}$ and by Proposition 3, there exists an observer for $\mathcal{O}_{Q_{\mathrm{c}}}$ such that

DOI: 10.1002/rnc

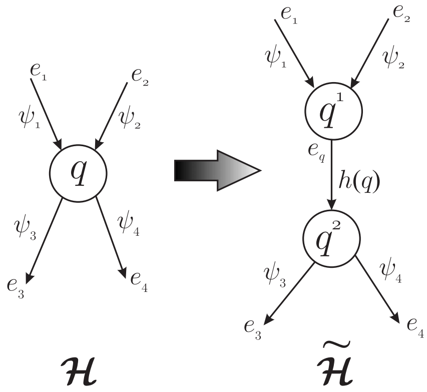

Figure 1. Discrete states of $\mathscr{S}$ are split by Algorithm 4, to consider the generation time of $h(q)$.

if $q(I_k)\in Q_c$, $I_k=[t_k,t'_k]$, then the observer's output $\hat{y}(t)=1$ for all $t\in\tilde{I}_k=[t_k+\delta_{h(q(I_k))},t'_k]\supseteq[t_k+\delta^*,t'_k]\supseteq[t_k+\delta,t'_k]$. Condition 1 also implies that there exists an observer $\tilde{\mathcal{O}}_{\tilde{Q}_c}$ for $\tilde{\mathscr{S}}$ such that if $\tilde{q}(\tilde{I}_k)\notin\tilde{Q}_c$, then the observer's output $\hat{y}(t)=0$ for all $t\in\tilde{I}_k$. By construction of $\tilde{\mathscr{S}}$ and by Proposition 3, there exists an observer for $\mathcal{O}_{Q_c}$ such that if $q(I_k)\notin Q_c$, $I_k=[t_k,t'_k]$, then the observer's output $\hat{y}(t)=0$ for all $t\in\tilde{I}_k=[t_k+\delta_{h(q(I_k))},t'_k]\supseteq[t_k+\delta^*,t'_k]\supseteq[t_k+\delta,t'_k]$. □

Given a solution (Ψ_e,h) obtained using Algorithms 2 and 3, if the first condition of Theorem 2 holds, then $\delta_{\min}=\delta^*$ as defined in the proof. The condition is only sufficient since $\tilde{\mathscr{S}}$ embeds continuous inputs and outputs of $\mathscr{S}$ by means of extra output discrete signals that are not unique. The condition becomes necessary and sufficient if these extra output signals represent all the available information.

4. HIDDEN MARKOV MODELS

In this section, we extend our results to the class of hidden Markov models [21]. We propose an observability definition similar to that given in [18] for the continuous states of jump linear systems, which allows a bound in the probability of estimation uncertainty. We show that the addressed observability verification problem is decidable, and we characterize the computational complexity.

Let $\mathbb{P}[q(k)=q]$ denote the probability that the discrete state is q at time k.

Definition 9 (*Hidden Markov model*)
A hidden Markov model is a triple $\mathscr{M}=(\mathscr{D},\Pi,\Pi_0)$, where

- $\mathscr{D}=(Q,Q_0,E,\Psi,\eta)$ is a discrete event system as in Definition 1, where the discrete output function is redefined as an output probability function $\eta: E\times\Psi\to[0,1]$, which associates with each edge $e\in E$ and output $\psi\in\Psi$ the probability that the discrete transition e generates the symbol ψ as output.
- Π is a $N\times N$ transition probability matrix defined by

$$\Pi_{ij}=\mathbb{P}[q(I_{k+1})=q_j|q(I_k)=q_i],\forall k\geqslant 0$$

with $\sum_{j=1}^{N}\Pi_{ij}=1$, $\forall i=1,\ldots,N$.

DOI: 10.1002/rnc

- Π_0 is a N-dimensional initial probability vector defined by

$$\Pi_{0i} = \mathbb{P}[q(I_0) = q_i]$$

with $\sum_{i=1}^{N} \Pi_{0i} = 1$.

We define execution and observation languages of hidden Markov models as in Section 2. When not clear from the context, a superscript will identify the system we refer to. Given $\mathscr{M} = (\mathscr{D}, \Pi, \Pi_0)$, the spaces of all executions of $\mathscr{M}$ and $\mathscr{D}$ coincide, i.e. $\mathscr{L}^{\mathscr{M}} = \mathscr{L}^{\mathscr{D}}$. However, while for $\mathscr{D}$ the discrete execution is generated by a nondeterministic algorithm, for $\mathscr{M}$ it is generated by a probability space, uniquely identified by the transition probability matrix Π and by the initial probability distribution Π_0. We use these additional information on the system execution to define an observability notion for hidden Markov models, which requires a bound in the estimation uncertainty probability. We then relate this notion to the one introduced in Section 2.

In the observability definition for a discrete event system $\mathscr{D}$, we required the existence of an observer able to detect whether the current discrete state is in the critical set or not, without error. Since we have defined on a hidden Markov model $\mathscr{M}$ a probability measure in the target and output of a discrete transition, one can use the discrete observations to compute (using the Viterbi algorithm [22, 23]) the conditional probability distribution of the current discrete state given the measured observation. We define the conditional probability that the final state of an execution ρ belongs to a critical set Q_{c} given an observation p by $\mathbb{P}[\rho \in \mathscr{L}_{Q_{\mathrm{c}}} | P(\rho) = p]$. Notice that the output function $P : \mathscr{L} \to \mathscr{P}$ defined in Section 2 is not invertible. Since more than one path can generate the same observation, it is possible to define for each observation $p \in \mathscr{P}$ generated by the system $\mathscr{M}$, the set of executions that generate p as observation by $P^{-1}(p) \triangleq \{\rho \in \mathscr{L} : P(\rho) = p\}$, where $P^{-1} : \mathscr{P} \to \mathscr{L}$ is a map. We assume in this section, the absence of edges whose output is unobservable: this implies that $\forall p \in \mathscr{P}$, $\forall \rho \in P^{-1}(p)$, $|p| = |\rho|$.

With the assumption that our observer generates as output the most likely current discrete state according to the Viterbi algorithm estimate, we formalize an observability definition that requires a bound in the probability of estimation error. More precisely, we require that the probability of an estimation error is always bounded by $(\lambda_m, \lambda_M) \in [0, 1] \times [0, 1]$, $1 - \lambda_m < \lambda_M$ as follows:

1. $1 - \lambda_m$ is the worst-case probability that the observer misses to detect that a critical state is currently active, when the current discrete state of $\mathscr{M}$ is in the critical set (*observation miss probability*).
2. $1 - \lambda_M$ is the worst-case probability that the observer detects that a critical state is currently active, when the current discrete state of $\mathscr{M}$ is not in the critical set (*false alarm probability*).

We can formalize the above properties as follows.

Definition 10 (*Observer with bounded reliability*)
Given a hidden Markov model $\mathscr{M}$, an observer of the critical set Q_{c} with reliability (λ_m, λ_M) is a function $\mathscr{O}_{Q_{\mathrm{c}}} : \mathscr{P} \to \{0, 1\}$ such that

$$\mathscr{O}_{Q_{\mathrm{c}}}(P(\rho)) = \begin{cases} 1 & \text{if } \mathbb{P}[\rho \in \mathscr{L}_{Q_{\mathrm{c}}} | P(\rho) = p] \in [\lambda_M, 1] \\ 0 & \text{if } \mathbb{P}[\rho \in \mathscr{L}_{Q \setminus Q_{\mathrm{c}}} | P(\rho) = p] \in [0, 1 - \lambda_m] \end{cases}$$

A set Q_{c} is said to be observable with reliability (λ_m, λ_M) for $\mathscr{M}$ if and only if an observer $\mathscr{O}_{Q_{\mathrm{c}}}$ with reliability (λ_m, λ_M) exists.

The definition above characterizes a structural property of the hidden Markov model $\mathscr{M}$. This condition guarantees that, if $\lambda_m \simeq 1$ and $\lambda_M \simeq 1$, then for each observation of $\mathscr{M}$ we are very confident either that the discrete state is critical or that it is not. When $\lambda_m = \lambda_M = 1$ the observer provides correct estimate with probability 1. In what follows, observability with reliability $\lambda \triangleq (\lambda_m, \lambda_M)$ will be called λ-observability. A necessary and sufficient condition for λ-observability is the following.

Proposition 4
Given a hidden Markov model $\mathscr{M}$, the set Q_{c} is observable with reliability (λ_m, λ_M) if and only if

$$\forall p \in \mathscr{P}, \quad \mathbb{P}[\rho \in \mathscr{L}_{Q_{\mathrm{c}}} | P(\rho) = p] \in [0, 1-\lambda_m] \cup [\lambda_M, 1] \tag{4}$$

We introduce a relation between λ-observability and the observability notion given in Definition 4.

Proposition 5
Given a hidden Markov model $\mathscr{M} = (\mathscr{D}, \Pi, \Pi_0)$, then a critical set Q_{c} is λ-observable with $\lambda = (1, 1)$ for $\mathscr{M}$ if and only if Q_{c} is observable for $\mathscr{D}$.

Proof
($\Rightarrow$) Let Q_{c} be λ–observable with $\lambda = (1, 1)$ for $\mathscr{M}$: this implies that, if $p \in \mathscr{P}^{\mathscr{M}} = \mathscr{P}^{\mathscr{D}}$, then $\mathbb{P}[\rho \in \mathscr{L}_{Q_{\mathrm{c}}} | P(\rho) = p]$ is either 0 or 1: if it is 0, this implies that $p \notin \mathscr{P}^{\mathscr{D}}_{Q_{\mathrm{c}}}$, thus $p \notin \mathscr{P}^{\mathscr{M}}_{Q_{\mathrm{c}}}$ as well; if it is 1, this implies that $p \notin \mathscr{P}^{\mathscr{D}}_{Q \setminus Q_{\mathrm{c}}}$, thus $p \notin \mathscr{P}^{\mathscr{M}}_{Q \setminus Q_{\mathrm{c}}}$ as well. It follows that $\mathscr{P}^{\mathscr{D}}_{Q_{\mathrm{c}}} \cap \mathscr{P}^{\mathscr{D}}_{Q \setminus Q_{\mathrm{c}}} = \emptyset$ and Q_{c} is observable for $\mathscr{D}$. ($\Leftarrow$) Let Q_{c} be observable for $\mathscr{D}$, then it is possible to construct an observer that deterministically detects whether the current discrete state of $\mathscr{D}$ belongs to Q_{c} or not: this clearly implies that Q_{c} is λ-observable with $\lambda = (1, 1)$ for $\mathscr{M}$. □

In order to verify observability of a given hidden Markov model, one can check if $\mathbb{P}[\rho \in \mathscr{L}_q | P(\rho) = p]$ satisfies condition (4) for any $p \in \mathscr{P}$. However, $\mathscr{P}$ almost always has infinite cardinality because of cycles in the discrete layer of $\mathscr{M}$. Thus, it is not possible to execute the above computation in finite time. We prove now that the λ-observability verification problem is decidable.

Theorem 3
Given a hidden Markov model $\mathscr{M} = (\mathscr{D}, \Pi, \Pi_0)$ and a set Q_{c}, λ-observability verification problem of Q_{c} is decidable, and belongs to the complexity class EXPTIME.

Proof
As first step of the proof, we remark that condition (4) can be rewritten as

$$\forall p \in \mathscr{P}_{Q_{\mathrm{c}}} \cap \mathscr{P}_{Q \setminus Q_{\mathrm{c}}} \mathbb{P}[\rho \in \mathscr{L}_{Q_{\mathrm{c}}} | P(\rho) = p] \in [0, 1-\lambda_m] \cup [\lambda_M, 1] \tag{5}$$

In fact, for any given λ_m, λ_M, for all $p \in \mathscr{P} \setminus (\mathscr{P}_{Q_{\mathrm{c}}} \cap \mathscr{P}_{Q \setminus Q_{\mathrm{c}}})$ and for all $\rho \in P^{-1}(p)$, then

1. either $\rho \in \mathscr{L}_{Q_{\mathrm{c}}}$, and thus $\mathbb{P}[\rho \in \mathscr{L}_{Q_{\mathrm{c}}} | P(\rho) = p] = 1$.
2. or $\rho \in \mathscr{L}_{Q \setminus Q_{\mathrm{c}}}$, and thus $\mathbb{P}[\rho \in \mathscr{L}_{Q_{\mathrm{c}}} | P(\rho) = p] = 0$.

This implies that condition (4) is already satisfied for any $p \in \mathscr{P} \setminus (\mathscr{P}_{Q_{\mathrm{c}}} \cap \mathscr{P}_{Q \setminus Q_{\mathrm{c}}})$. We recall from Section 2 that $\mathscr{P}_{Q_{\mathrm{c}}} \cap \mathscr{P}_{Q \setminus Q_{\mathrm{c}}}$ is accepted by the DFA $\mathscr{O}_{Q_{\mathrm{c}}}$. As discussed in [3], it is possible to use the structure of $\mathscr{O}_{Q_{\mathrm{c}}}$ to compute the conditional probability distribution

$$\pi_i \triangleq \mathbb{P}[\rho \in \mathscr{L}_{q_i} | P(\rho) = p] \quad \forall i = 1, \ldots, N$$

DOI: 10.1002/rnc

for any observation $p\in\mathscr{P}$. In other words, it is possible to implement the Viterbi algorithm using the discrete layer given by $\mathscr{O}_{Q_c}$ and a continuous variable $\pi=(\pi_1,\ldots,\pi_N)\in[0,1]^N$, $N=|Q|$, which is reset every time a new output symbol is generated by the system $\mathscr{M}$. π_i is the probability that the current discrete state of $\mathscr{M}$ is $q_i\in Q$. The initial discrete state is $\hat{q}_0$, and the initial condition of π is given by the initial probability distribution $\pi(0)=\Pi_0$. For this reason, $\pi(k)$ is the probability distribution during the hybrid time basis interval I_k. When $\mathscr{M}$ generates an output symbol, $\mathscr{O}_{Q_c}$ switches the discrete state according to a transition e, and $\pi(k)$ is reset to a value $\pi(k+1)$ according to a matrix R_e, and normalized such that $\sum_{i=1}^N \pi_i(k+1)=1$

$$\pi'(k)=R_e\pi(k),\quad \pi(k+1)=\frac{\pi'(k)}{\sum_{i=1}^N \pi'_i(k)}$$

Given any $p\in\mathscr{P}$, $\pi(k)$ evolves to a state $\pi(|p|)$, that is the conditional probability distribution of the discrete state of $\mathscr{M}$ given the observation p

$$\pi_i(|p|)=\mathbb{P}[\rho\in\mathscr{L}_{q_i}|P(\rho)=p]$$

It is straightforward to define the conditional probability that the current state is critical given the observation p

$$\pi_c(|p|)\triangleq\sum_{q_i\in Q_c}\pi_i(|p|)=\mathbb{P}[\rho\in\mathscr{L}_{Q_c}|P(\rho)=p]$$

We propose an algorithm to verify in a finite number of steps whether condition (5) holds, by checking that $\pi_c(|p|)$ does not reach the set $(1-\lambda_m,\lambda_M)$ for all $p\in\mathscr{P}_{Q_c}\cap\mathscr{P}_{Q\setminus Q_c}$.

It is sufficient to consider all executions of $\mathscr{O}_{Q_c}$ that terminate in $\hat{Q}_c$ with just one cycle. If for all those executions $\pi_c(|p|)$ does not reach the set $(1-\lambda_m,\lambda_M)$, then for all executions with more than one cycle $\pi_c(|p|)$ does not reach the set $(1-\lambda_m,\lambda_M)$ as well, and condition (5) is satisfied. On the contrary, if there exists just one *bad* cycle that brings $\pi_c(|p|)$ in the set $(1-\lambda_m,\lambda_M)$, then condition (5) is not satisfied. Checking that such *bad* cycles do not exist provides necessary and sufficient conditions for λ-observability.

The algorithm consists of four iterations. For each $\hat{q}\in\hat{Q}_c$ (iteration 1), and for any path without cycles $\hat{q}(0),\hat{q}(1),\ldots,\hat{q}(f)$ where $\hat{q}(0)=\hat{q}_0,\hat{q}(f)=\hat{q}$ (iteration 2), compute the probability distribution at the end of the path

$$\pi_c(f)\triangleq\sum_{i\in Q_c}\pi_i(f)$$

Let $\pi_c(f)$ satisfy condition (5), namely $\pi_c(f)\in[0,1-\lambda_m]\cup[\lambda_M,1]$. For any $k=0,\ldots,f$ (iteration 3), compute $\pi_c(f)$ as function of $\pi(k)$: if $\pi_c(f)$ is independent by $\pi(k)$, then skip to $k+1$. Otherwise, consider each cycle that crosses $q(k)$ (iteration 4): notice that for a cycle of length c and a value of $\pi(k)$ at the beginning of the cycle, then $\pi(k+c)$ computed on a run of the cycle is either equal to $\pi(k)$ or it is different. In the first case, skip to the next cycle since condition (5) on $\pi_c(f)$ is satisfied for any number $n\geqslant 0$ of times the cycle is crossed. In the second case, each $\pi_i(k+nc)$ is either monotone increasing or monotone decreasing w.r.t. n crosses of the cycle. Since any $\pi_i(k+nc)$ is upper bounded by one and lower bounded by zero, then $\lim_{n\to\infty}\pi(k+nc)$ is a vector of zeros and ones. For this reason, $\pi_c(f)$ is either monotone increasing or monotone decreasing as well, and converges to a value that depends on $\lim_{n\to\infty}\pi(k+nc)$. If $\pi_c(f+nc)$ belongs to the set $[0,1-\lambda_m]\cup[\lambda_M,1]$ for any $n\geqslant 0$, then skip to the next cycle. Otherwise, exit

DOI: 10.1002/rnc

algorithm since Q_c is not λ-observable for $\mathscr{M}$. Executing iterations 1–4 completes the verification. The maximum cardinality of $\hat{Q}_c$ is bounded by 2^{N-1} (iteration 1). The number of paths without cycles connecting q_0 and $\hat{q}$ is bounded by N (iteration 2). The number of states for each path without cycles is bounded by N (iteration 3). The number of cycles crossing $q(k)$ is bounded by N^2 (iteration 4). Since the verification requires at most $N^4 \cdot 2^{N-1}$ steps, the result holds. □

If Q_c is observable with reliability $(\lambda_m^*, \lambda_M^*)$, then it is observable with reliability (λ_m, λ_M) for any $\lambda_m < \lambda_m^*$, $\lambda_M < \lambda_M^*$. The maximum reliability $(\lambda_m^{\max}, \lambda_M^{\max})$ such that Q_c is $(\lambda_m^{\max}, \lambda_M^{\max})$-observable can be determined as shown in the proof of Theorem 3. An example of this computation is given in the following section.

The notion of λ-observability can be generalized to a notion of λ-observability in K steps as done in Section 2. Theorem 3 still holds, since it is sufficient to check conditions only for paths of length greater than K.

5. EXAMPLE

Consider the discrete event system $\mathscr{D}$ described in Figure 2. We use the theoretical results discussed above to analyze the discrete state observability. Let $Q_c = \{q_7\}$. It is possible to define the languages of observations for each discrete state by means of regular expression [14]:

$$\mathscr{P}_{q_1} = \{\varepsilon\}, \quad \mathscr{P}_{q_2} = a(aa+bb)^*$$

$$\mathscr{P}_{q_3} = a(bb)^*, \quad \mathscr{P}_{q_4} = a(aa+bb)^* b$$

$$\mathscr{P}_{q_5} = a(aa+bb)^* b, \quad \mathscr{P}_{q_6} = a(bb)^* b$$

$$\mathscr{P}_{q_7} = a(bb)^* b$$

Following Algorithm 1, it is possible to compute the language

$$\mathscr{P}_{Q_c} \cap \mathscr{P}_{Q \setminus Q_c} = \mathscr{P}_{q_7} \cap \bigcup_{i=1}^{6} \mathscr{P}_{q_i} = a(bb)^* b \neq \emptyset$$

The discrete state observer $\mathscr{O}_{q_7}$ associated with $\mathscr{D}$ is illustrated in Figure 2. It is clear that the system is not observable.

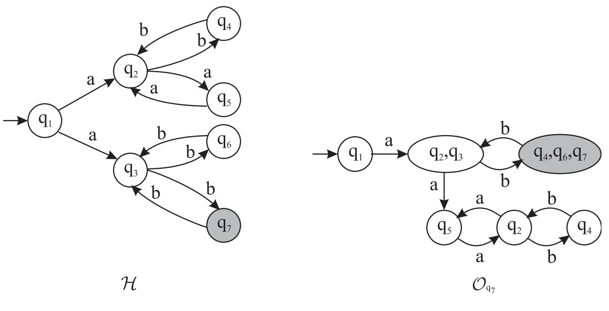

Figure 2. Discrete layers of $\mathscr{D}$ and $\mathscr{O}_{q_7}$.

DOI: 10.1002/rnc

Consider now a switching system $\mathscr{S}$ defined upon $\mathscr{D}$, where all discrete states are characterized by different continuous dynamics except q_4 and q_7, namely $\mathscr{E}_{q_4} = \mathscr{E}_{q_7}$. This implies that $h(q_4) = h(q_7)$, i.e. it is not possible to generate different extra output signals for q_4 and q_7. As discussed before, we can use the information given by the continuous output, and we therefore apply Algorithms 2 and 3 to find the set of extra information we need to achieve observability. The sub-optimal approach yields to a set of extra outputs $\{h(q_4), h(q_6), h(q_7)\}$, that is not a solution to obtain observability of $\{q_7\}$ since $h(q_4) = h(q_7)$. The optimal algorithm provides the set of extra information $\{h(q_2), h(q_3)\}$. In this case, by detecting if the system visited q_2 or q_3, we anticipate the uncertainty between q_4, q_6, q_7 and we use only two extra outputs. Even if the generation times $\delta_{q_2}, \delta_{q_3}$ are greater than zero, Theorem 2 implies that the system augmented with the extra output $\{h(q_2), h(q_3)\}$ is observable with delay zero.

Assume now that it is not possible to use the continuous input and output signals: then the critical state $\{q_7\}$ is not observable in the sense of Definition 4. However, if we own a stochastic characterization of the system execution by means of transition probabilities, we can apply our results in the stochastic setting to analyze the weaker notion of λ-observability. Consider the hidden Markov model $\mathscr{M} = (\mathscr{D}, \Pi_0, \Pi)$, where the output function is deterministic and

$$\Pi_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \Pi = \begin{bmatrix} 0 & 0.9 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0.99 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0.99 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Applying the verification procedure presented in Theorem 3, we compute:

$$\mathbb{P}[\rho \in \mathscr{L}_{q_4} | P(\rho) = ab] = \tfrac{9}{109}$$

$$\mathbb{P}[\rho \in \mathscr{L}_{q_6} | P(\rho) = ab] = \tfrac{1}{109}$$

$$\mathbb{P}[\rho \in \mathscr{L}_{q_7} | P(\rho) = ab] = \tfrac{99}{109} \cong 0.9$$

$$\mathbb{P}[\rho \in \mathscr{L}_{q_7} | P(\rho) = a(bb)^n b] = (\tfrac{10}{99}(\tfrac{1}{99})^n + 1)^{-1}$$

Thus, we can state that

$$\forall p \in \mathscr{P}_{Q_c} \cap \mathscr{P}_{Q \setminus Q_c}, \quad \mathbb{P}[\rho \in \mathscr{L}_{q_7} | P(\rho) = p] \in \{0\} \cup [0.9, 1]$$

hence the critical set $\{q_7\}$ is observable for $\mathscr{M}$ with reliability ($\lambda_m^{\max} = 1, \lambda_M^{\max} = 0.9$). This implies that, even if the system is not 'deterministically' observable, it is possible to detect with probability 1 whether a critical state is currently visited, and the probability to generate a false alarm is in the worst case less than $1 - \lambda_M^{\max} = 0.1$.

DOI: 10.1002/rnc

6. CONCLUSIONS

We introduced a notion of observability with respect to a subset of critical discrete states. For discrete event systems, we exploited properties of regular languages to propose an algorithm for checking observability in polynomial time. We extended our result to switching systems: we proposed an algorithm to find the minimum set of extra output information, retrieved from the continuous observations, to satisfy the observability condition, and discussed a notion of observability with bounded delay. We then extended our results to hidden Markov models: we proposed an observability definition that requires a bound in the probability of observation reliability, and we showed that the verification problem is decidable and belongs to the complexity class EXPTIME.

The framework proposed in this paper can be used for the simulation of real safety critical procedures, and verification of the detection of dangerous operations, as shown in [2, 3]. Future work will focus on the extension of our results to continuous time hidden Markov models.

APPENDIX A

Definition A1 (*NFA*)
A NFA is a tuple $\mathscr{N}=(Q, Q_0, Q_f, E, \Psi, \eta)$, such that the set of initial states $Q_0=\{q_0\}$ is a singleton and $Q_f \subseteq Q$ is the set of final states. The language accepted by an NFA $\mathscr{N}$ is the language of the observations $\mathscr{P}_{Q_f}$ on the alphabet Ψ.

Definition A2 (*DFA*)
A DFA is an NFA $\mathscr{D}=(Q, q_0, Q_f, E, \Psi, \eta)$, such that $\eta: E \rightarrow 2^{\Psi}$ and for each $q \in Q$ the set $\{\eta(e)\}_{e\in E:s(e)=q}$ is a partition of Ψ. The language accepted by a DFA $\mathscr{D}$ is the language of the observations $\mathscr{P}_{Q_f}$ on the alphabet Ψ.

Definition A3 (*Regular language*)
A language $\mathscr{L}$ is called a *regular language* if there exists a NFA that accepts $\mathscr{L}$.

Proposition A1
Given a regular language $\mathscr{L}$ accepted by a NFA $\mathscr{N}$, it is possible to construct a DFA $\mathscr{D}$ that accepts $\mathscr{L}$. The cardinality of the state space of $\mathscr{D}$ is exponential with respect to the cardinality of the state space of $\mathscr{N}$.

Proposition A2
Regular languages are closed with respect to the operations of union, intersection and complement.

Let $\mathrm{cl}_{\varepsilon}(Q^*)$ be the ε-closure [14] of a set of states $Q^* \subseteq Q$, namely the set of states that can be reached from Q^* via a path of edges whose outputs are unobservable.

Algorithm A1 (*Discrete state observer construction*)
Given a discrete event system $\mathscr{D}=(Q, Q_0, E, \Psi, \eta)$, and a critical set Q_{c}, construct a DFA $\mathscr{O}_{Q_{\mathrm{c}}}=(\hat{Q}, \hat{q}_0, \hat{Q}_{\mathrm{c}}, \hat{E}, \hat{\Psi}, \hat{\eta})$ as follows:

1. $\hat{Q} \triangleq cl_{\varepsilon}(Q_0) \subseteq 2^Q$;
2. $\hat{q}_0 \triangleq \{Q_0\} \subseteq 2^Q$;

DOI: 10.1002/rnc

3. $\hat{Q}_{\mathrm{c}} \triangleq \{\hat{q} \in \hat{Q} : \hat{q} \cap Q_{\mathrm{c}} \neq \emptyset \wedge \hat{q} \cap Q \backslash Q_{\mathrm{c}} \neq \emptyset\} \subseteq 2^{Q}$;
4. $\hat{\Psi} \triangleq \Psi \backslash \{\varepsilon\}$;
5. In order to define $\hat{E}$ and $\hat{\eta}$, for each unvisited discrete state $\hat{q} \in \hat{Q}$ do

   5.1 For each $\psi \in \hat{\Psi}$, define $\hat{q}'\{q' \in Q : \exists e \in E, \exists q \in \hat{q}, q = s(e), q' = t(e), \eta(e) = \psi\}$: if $\hat{q}' \neq \emptyset$ then assign $\hat{Q} = \hat{Q} \cup cl_{\varepsilon}(\hat{q}')$, $\hat{E} = \hat{E} \cup \tilde{e} = \{\hat{q}, \hat{q}'\}$, and $\tilde{\eta}(\tilde{e}) = \psi$;
   5.2 Mark $\hat{q}$ as visited.

Definition A4 (*Hybrid time basis* [*24*])
A hybrid time basis $\tau \triangleq \{I_k\}_{0 \leqslant k \leqslant |\tau|}$ is a finite or infinite sequence of intervals $I_k = [t_k, t'_k]$. The length $t'_k - t_k$ of every interval I_k denotes the dwelling time in a discrete state, while the extremes t_k, t'_k specify the switching instants of the hybrid flow. The number of such intervals is the cardinality $|\tau|$ of the time basis. Furthermore, the following hold:

1. $t_k \leqslant t'_k$ for $k > 0$, and $t'_{k-1} = t_k$ for $k > 1$.
2. If the sequence is infinite, i.e. $|\tau| = \infty$, then I_k is closed for all k.
3. If the sequence is finite, i.e. $|\tau| < \infty$, then the last interval $I_{|\tau|}$ might be right-open.

Definition A5 (*Hybrid execution* [*24*])
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 $q : \tau \rightarrow Q$ piecewise continuous, and $x : \tau \rightarrow X$. Functions q, x, defined on the hybrid time basis τ, take values on the hybrid state space, and satisfy the continuous and discrete dynamics.

Definition A6 (*Minimum and maximum dwell time*)
Given a switching system $\mathscr{S}$, we define for each state $q \in Q$ a (possibly infinite) minimum dwell time $\Delta_m(q) \geqslant 0$ and a (possibly infinite) maximum dwell time $\Delta_M(q) \geqslant 0$, namely the minimum and maximum time that can be spent in the discrete state q. This implies that given an execution χ of $\mathscr{S}$, then $\Delta_m(q(I_k)) \leqslant t'_k - t_k \leqslant \Delta_M(q(I_k))$ for all $k = 0, \ldots, |\tau|$.

ACKNOWLEDGEMENTS

The authors thank Elena De Santis and Giordano Pola for useful discussions on observability of hybrid systems. The third author wishes to thank Rajeev Alur for illuminating lectures and discussions on regular languages.

REFERENCES

1. Di Benedetto MD, Di Gennaro S, D'Innocenzo A. Critical observability for a class of stochastic hybrid systems and application to air traffic management. *Deliverable 7.5*, Project IST-2001-32460 HYBRIDGE. Available from: http://www.nlr.nl/public/hosted-sites/hybridge, May 2005.
2. Di Benedetto MD, Di Gennaro S, D'Innocenzo A. Error detection within a specific time horizon and application to air traffic management. *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference* (*CDC–ECC'05*), Seville, Spain, 2005; 7472–7477.
3. Di Benedetto MD, Di Gennaro S, D'Innocenzo A. Critical states detection with bounded probability of false alarm and application to air traffic management. *Proceedings of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems* (*ADHS*), Alghero, Sardinia, Italy, 2006.
4. Ramadge P. Observability of discrete event systems. *Proceedings of the 25th IEEE Conference on Decision and Control*, Athens, Greece, 1986; 1108–1112.

DOI: 10.1002/rnc

5. Ozveren C, Willsky A. Observability of discrete event dynamic systems. *IEEE Transactions on Automatic Control* 1990; **35**(7):797–806.
6. Cassandras CG, Lafortune S. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers: Dordrecht, 1999.
7. Yoo T, Lafortune S. On the computational complexity of some problems arising in partially-observed discrete event systems. *Proceedings of the 2001 American Control Conference*, Arlington, VA, 2001; 25–27.
8. Oishi M, Hwang I, Tomlin C. Immediate observability of discrete event systems with application to user-interface design. *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, U.S.A., 2003; 2665–2672.
9. Balluchi A, Benvenuti L, Di Benedetto MD, Sangiovanni-Vincentelli A. Design of observers for hybrid systems. In *Hybrid Systems*: *Computation and Control*, Tomlin C, Greensreet M (eds). Lecture Notes in Computer Science, vol. 2289. Springer: Berlin, 2002; 76–89.
10. Di Gennaro S. Notes on the nested observers for hybrid systems. *Proceedings of the European Control Conference 2003—ECC'03*, Cambridge, U.K., 2003.
11. De Santis E, Di Benedetto MD, Pola G. On observability and detectability of continuous-time switching linear systems. *Proceedings of the 42nd IEEE Conference on Decision and Control*, *CDC 03*, Maui, HI, U.S.A., 2003; 5777–5782.
12. Di Benedetto MD, Di Gennaro S, D'Innocenzo A. Critical observability and hybrid observers for error detection in air traffic management. *Proceedings of the 13th Mediterranean Conference on Control and Automation*, Limassol, Cyprus, 2005.
13. D'Innocenzo A, Di Benedetto MD, Di Gennaro S. Observability of hybrid automata by abstraction. In *Hybrid Systems*: *Computation and Control*, Hespanha J, Tiwari A (eds). Lecture Notes in Computer Science, vol. 3927. Springer: Berlin, 2006; 169–183.
14. Hopcroft J, Ullman J. *Introduction to Automata Theory*, *Languages and Computation*. Addison-Wesley: Reading, MA, 1979.
15. Frank P. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy—a survey and some new results. *Automatica* 1990; **26**(3):459–474.
16. Massoumnia M, Verghese G, Willsky A. Failure detection and identification. *IEEE Transactions on Automatic Control* 1989; **34**(3):316–321.
17. Balluchi A, Benvenuti L, Lemma C, Sangiovanni-Vincentelli A, Serra G. Actual engaged gear identification: a hybrid observer approach. *Proceedings of the 16th IFAC World Congress*, Prague, CZ, 2005.
18. Mariton M. *Jump Linear Systems in Automatic Control*. M. Dekker Inc.: New York, 1990.
19. Lygeros J. Lecture notes on hybrid systems. *ENSIETA*, *2–6/2*, 2004.
20. Ames A, Abate A, Sastry S. Sufficient conditions for the existence of zeno behavior in hybrid systems. *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005.
21. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 1989; **77**(2):257–286.
22. Forney GD. The Viterbi algorithm. *Proceedings of the IEEE* 1973; **61**(3):268–278.
23. Viterbi AJ. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 1967; **13**(2):260–269.
24. Lygeros J, Tomlin C, Sastry S. Controllers for reachability specications for hybrid systems. *Automatica*, *Special Issue on Hybrid Systems* 1999; **35**:349–370.

DOI: 10.1002/rnc