On Decentralized Observability of Discrete Event Systems

M.P. Cabasino, A. Giua, C. Mahulea, C. Seatzu

Abstract

In this paper we deal with the problem of decentralized observability of discrete event systems. We consider a set of sites that observe a subset of events. Each site transmits its own observation to a coordinator that decides if the word observed belongs to a legal behavior or not. We study two different properties: uniform $q-$observability and $q-$diagnosability. Then, we prove that both properties are decidable for regular languages. Finally, we give an algorithm to compute starting from a given initial state, the time instants at which the synchronization has to be done so as to guarantee that if an illegal word has occurred it is immediately detected.

Published as:

M.P. Cabasino, A. Giua, A. Mahulea, C. Seatzu "On Decentralized Observability of Discrete Event Systems," *50th IEEE Conf. on Decision and Control* (Orlando, Florida, USA), Dec 2011.

This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498). At University of Zaragoza it was partially supported also by CICYT - FEDER projects DPI2010-20413 and by Fundación Aragón I+D.

M.P. Cabasino, A. Giua and C. Seatzu are with the Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy {cabasino,giua,seatzu@diee.unica.it}.

C. Mahulea is with the Aragón Institute of Engineering Research (I3A), University of Zaragoza, Maria de Luna 1, 50018 Zaragoza, Spain {cmahulea@unizar.es}.

I. Introduction

A. Motivation

In [1] Tripakis defines a property that he calls local observability. The idea is the following: a set of n local sites observe, through their own projection masks P_i, a word w of symbols that is known to belong to a language L. A language $K \subset L$ is locally observable if, assuming all local sites send to a coordinator all observed strings $P_i(w)$, the coordinator can decide for any w if the word belongs to K or to $L \setminus K$. Note that this property was shown in [1] to be undecidable even when languages L and K are regular: this is due to the fact that the length of a word w can be arbitrarily long. On the contrary, assuming only words of bounded length are considered, the property is decidable for arbitrary languages, since it must only be checked over a finite number of strings.

We observe that this property is closely related to local *diagnosability* as defined by Sampath *et al*. [2]. In fact, language K in this setting represents the set of all fault-free evolutions, while the larger set L also includes the faulty ones.

The problem we want to address is the following. Assume w describes the event driven evolution of a system. The coordinator can at any moment send a request to all local sites to know the observed words since the previous request: such a mechanism is called *synchronization*. After each synchronization a coordinator should be able to decide if, on the basis of the information received so far from the local sites, the word w generated is legal, i.e, belongs to K. Note that a synchronization is costly, thus although we assume that the maximal number of events that can be generated by the system between two consecutive synchronizations is bounded, the coordinator should request as few synchronizations as needed to solve the observability problem. Also the distance between two consecutive synchronizations, expressed in terms of the number of events generated between them, needs not be constant but may opportunistically vary with the word generated so far.

In this setting, although the basic notion of local observability given by Tripakis is still fundamental, two major extensions are needed. In fact the observability property defined in [1] makes two rather restrictive assumptions.

The first assumption is that the observability property is defined only with respect to words in L. On the contrary, in our setting synchronization occurs repeatedly. Thus if a synchronization

occurs after a word w has been generated we are interested in the observability of the residual language $w^{-1}K$, i.e., the set of all strings that belong to K and whose prefix is w, with respect to the residual language $w^{-1}L$. Correspondingly, we introduce the notion of *uniform* $q-$*observability*.

The second assumption in [1] is that when the observation starts the word generated so far (that as discussed in the previous paragraph is always the empty word) is perfectly known. On the contrary, in our setting when a synchronization occurs the coordinator should be able to determine if the generated string is legal or not, but may not be able to unambiguously estimate it. Thus when next observation starts the word generated so far is only known to belong to a given set. To capture this condition, we introduce the notion of $q-$*diagnosability*.

B. Literature review

Observability is a fundamental property that has received a lot of attention during the last decades due to the importance of reconstructing plant states that cannot be measured. Several contributions have been presented in the framework of automata [3], [4], [5], [6]. In [3] Caines *et al.* showed how it is possible to use the information contained in the past sequence of observations (given as a sequence of observation states and control inputs) to compute the set of consistent states, while in [4] the observer output is used to steer the state of the plant to a desired terminal state. A similar approach was also used by Kumar *et al.* [6] when defining observer based dynamic controllers in the framework of supervisory predicate control problems.

Özveren and Willsky [5] proposed an approach for building observers that allows one to reconstruct the state of finite automata after a word of bounded length has been observed, showing that an observer may have an exponential number of states. A problem strictly related to observability as defined in the present paper is opacity. A system is (current-state) opaque if its (current) state is never exposed to certainly belong to a given set of secret states. See the work of Saboori and Hadjicostis [7], [8] and of Dubreil *et al*. [9].

Finally, a very general approach for observability with communication has been presented by Barret and Lafortune in [10] in the context of supervisory control, and several techniques for designing a possibly optimal communication policy have also been discussed therein. By optimal we mean that the local sites communicate as late as possible, only when strictly necessary to prevent the undesirable behavior. Our work is by large a special case of the architecture in

[10] because we allow communications only between the coordinator and the local observers — and not among local observers — and we do not consider a control problem but simply an observation one. There are, however, a few differences in our approach — derived from [1] — with respect to [10] that motivate the need for additional investigation. These differences are listed here. First, we frame our results in the context of languages, rather than automata: this means that some of our definitions and results apply to possibly non regular languages. Secondly, while in [10] communications are decided by the local observers and are triggered by the observation of an event, in our case the communications are triggered by the coordinator. Finally, we assume that the coordinator knows the number of events generated so far, but cannot directly observe their label; thus the observation structure of the coordinator is not a projection mask but simply a function $f : L \to \mathbb{N}$ that counts the events generated so far.

Recently Ricker and Caillaud [11] have also considered a setting where communications may also be triggered by the receiver, that requests information from a sender. Furthermore, they also discuss policies where communication occurs after prefixes of any of the behaviors involved in a violation of co-observability, not just those that may result in undesired behavior.

II. Basic notations

Let Σ be a finite alphabet: Σ^* denotes the set of all finite strings over Σ, i.e., the Kleene star, and ε denotes the empty string. Given two strings u and v, uv is the concatenation of u and v.

A *deterministic finite automaton* (DFA) is a tuple $G = (X, \Sigma, \delta, x_0, X_m)$ where X is the set of states, Σ is the finite set of events, a partial function $\delta : X \times \Sigma \to X$ is the transition function, $x_0 \in X$ is the initial state, and $X_m \subseteq X$ is the set of *marked states*. The generated and marked languages of G, denoted by $\mathcal{L}(G)$ and $\mathcal{L}_m(G)$, respectively, are defined as $\mathcal{L}(G) = \{w \in \Sigma^* | \delta(x_0, w) \text{ is defined}\}$ and $\mathcal{L}_m(G) = \{w \in \Sigma^* | \delta(x_0, w) \in X_m\}$. Given two deterministic finite automata $G_1 = (X_1, \Sigma_1, \delta_1, x_{0,1}, X_{m,1})$ and $G_2 = (X_2, \Sigma_2, \delta_2, x_{0,2}, X_{m,2})$, the *parallel composition* of G_1 and G_2 is the automaton $G_1 \| G_2 = (X', \Sigma_1 \cup \Sigma_2, \delta', (x_{0,1}, x_{0,2}), X'_m)$, where

$X' \subseteq (X_1 \times X_2)$, $X'_m \subseteq (X_{m,1} \times X_{m,2})$ and

$$\delta'(x,e) = \begin{cases} (\bar{x}_1, x_2) & \text{if } e \in \Sigma_1 \setminus \Sigma_2, \delta_1(x_1,e) = \bar{x}_1; \\ (x_1, \bar{x}_2) & \text{if } e \in \Sigma_2 \setminus \Sigma_1, \delta_2(x_2,e) = \bar{x}_2; \\ (\bar{x}_1, \bar{x}_2) & \text{if } e \in \Sigma_1 \cap \Sigma_2, \delta_1(x_1,e) = \bar{x}_1, \\ & \quad \delta_2(x_2,e) = \bar{x}_2; \\ \text{not defined} & \text{otherwise.} \end{cases}$$

Given a word $w \in \Sigma^*$, and an alphabet $\Sigma_i \subseteq \Sigma$, we denote as $P_i(w)$ the projection of w over Σ_i, that can be recursively defined as follows. If $w = ue$, where $u \in \Sigma^*$ and $e \in \Sigma$, it holds

$$P_i(w) = \begin{cases} P_i(u)e & \text{if } e \in \Sigma_i, \\ P_i(u) & \text{otherwise} \end{cases}$$

Given a language L and a string $w \in \Sigma^*$, the *residual* of L with respect to (wrt) w is the language $w^{-1}L = \{z \mid wz \in L\}$. The language L is regular iff the set of its residuals as w ranges over Σ^* is finite, i.e., iff the set $\{w^{-1}L \mid w \in \Sigma^*\}$ is finite. The cardinality of the set $\{w^{-1}L \mid w \in \Sigma^*\}$ is called the *index* of L.

III. UNIFORM $q-$OBSERVABILITY

Let us consider two prefix-closed languages K and L defined over an alphabet Σ, such that $K \subset L \subseteq \Sigma^*$, and a set of n sub-alphabets $\Sigma_i \subseteq \Sigma$, $i = 1, \ldots n$.

The n sub-alphabets Σ_i's are associated to n sites $\mathcal{S}_i$, $i = 1, \ldots, n$. In particular, Σ_i includes all the events that can be *observed* by $\mathcal{S}_i$.

A first definition of decentralized observability has been given by Tripakis in [1] in the case of regular languages.

***Definition** 3.1:* Let us consider two regular languages L and K. The language K is *jointly observable wrt L and Σ_i*, for $i = 1, \ldots, n$, if there exists a total function $f : \Sigma_1^* \times \ldots \Sigma_n^* \to \{0, 1\}$, such that $\forall w \in L$

$$w \in K \;\Leftrightarrow\; f(P_1(w), \ldots, P_n(w)) = 1. \tag{1}$$

■

The above property uses unbounded memory since the word w may have arbitrary length, thus it is undecidable [1].

In this paper we generalize such a definition to the case of finite memory, i.e., the coordinator can at any moment send a request to all local sites to know the observed words since the previous request. On the basis of the information received so far from the local sites, the coordinator should establish if the evolution is legal.

***Definition** 3.2:* Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \ldots, n$, be n sub-alphabets of Σ. Let L and K be two prefix closed languages such that $K \subset L \subseteq \Sigma^*$.

The language K is called *uniformly $q-$observable wrt L and Σ_i*, for $i = 1, \ldots, n$, if $\forall\ w \in K$ there exists a function $f_w : \Sigma_1^* \times \ldots \times \Sigma_n^* \to \{0, 1\}$ such that $\forall\ u \in w^{-1}L$ with $|u| \leq q$, it holds

$$u \in w^{-1}K \iff f_w\left(P_1(u), \ldots, P_n(u)\right) = 1. \tag{2}$$

■

In simple words, uniform $q-$observability implies the possibility of establishing if the behavior of a given system is *legal*, only looking at the occurrence of no more than q events, and knowing that the sequence w preceding such events is legal.

Let us now introduce an equivalence relation among strings that allows us to rephrase the above definition of uniform observability.

***Definition** 3.3:* Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \ldots, n$, be n sub-alphabets of Σ. Let L and K be two prefix closed languages such that $K \subset L \subseteq \Sigma^*$.

A word $u \in w^{-1}L$ is *observation equivalent* (or simply *equivalent*) to $v \in w^{-1}L$, i.e., $u \equiv v$, if $P_i(u) = P_i(v)$ for all $i = 1, \ldots, n$. We denote $[u]$ the set of words that are equivalent to u. Finally, we say that two words that are not equivalent are *distinguishable*. ■

Using this notion, the definition of uniform $q-$observability of a language can be rewritten as follows.

***Definition** 3.4:* Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \ldots, n$, be n sub-alphabets of Σ. Let L and K be two prefix closed languages such that $K \subset L \subseteq \Sigma^*$.

The language K is called *uniform $q-$observable* wrt L and Σ_i, $i = 1, \ldots, n$, if $\forall\ w \in K$, and $\forall\ u \in w^{-1}L$,

$$w[u] \cap K \neq \emptyset \ \Rightarrow\ w[u] \subseteq K. \tag{3}$$

■

The following example clarifies the above definitions.

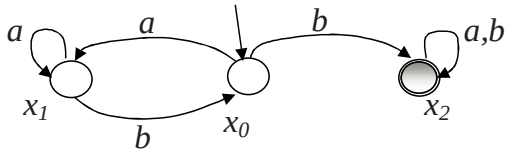

Fig. 1. The DFA considered in Example 3.5.

***Example** 3.5:* Let $\Sigma = \{a, b\}$, $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$, L be the language generated by the regular expression $(a + b)^*$, while K is the language generated by the regular expression $K_1 + K_2$ where $K_1 = (aa^*b)^*$ and $K_2 = (aa^*b)^* aa^*$.

It can be easily verified that L corresponds to the language generated by the DFA in Fig. 1 starting from x_0, while K is the language generated by the same DFA neglecting the state x_2, still assuming x_0 as the initial state. Moreover, K_1 corresponds to the set of words that finish in x_0, while K_2 corresponds to the set of words that finish in x_1.

We want to study the uniform $q-$observability of K wrt L, Σ_1 and Σ_2.

Let's start with $q = 1$. According to the definition of uniform $q-$observability we have to consider all possible words $u \in w^{-1}L$ of unitary length. This is equivalent to consider an arbitrary word $w \in (K_1 + K_2)$ followed by any word u of length 1. Since all words $w \in K_1$ terminate in x_0, then only two words of unitary length may occur after w, namely $u_1 = a$ and $u_2 = b$. Clearly it is $wu_1 \in K$ and $wu_2 \in L \setminus K$, therefore it should be

$$f_w(P_1(u_1), P_2(u_1)) = f_w(a, \varepsilon) = 1$$

and

$$f_w(P_1(u_2), P_2(u_2)) = f_w(\varepsilon, b) = 0.$$

Let us now consider an arbitrary word $w \in K_2$, i.e., an arbitrary word that terminates in x_1. Starting from x_1 the only admissible words of length 1 are $u_3 = a$ and $u_4 = b$. In such a case both wu_3 and wu_4 are in K, thus it should be

$$f_w(P_1(u_3), P_2(u_3)) = f_w(a, \varepsilon) = 1$$

and

$$f_w(P_1(u_4), P_2(u_4)) = f_w(\varepsilon, b) = 1.$$

This enables us to conclude that K is uniformly $1-$observable wrt L, Σ_1 and Σ_2.

Note that the same conclusion can be drawn using the notion of uniform 1-observability based on equivalence classes. Indeed, both u_1 and u_2, and u_3 and u_4 are distinguishable.

Let us now study uniform $2-$observability. As discussed above, if $w \in K_1$ we should consider all words $u \in L$ of length 2 that can be generated from x_0, i.e., $u \in \{aa, ab, ba, bb\}$. However, ab and ba are clearly equivalent but $wab \in K$ while $wba \in L \setminus K$. Thus K is not uniformly $2-$observable wrt L, Σ_1 and Σ_2.

In other terms, we can say that a function f_w satisfying the if and only if condition in (2) could not be defined. Indeed it should simultaneously be

$$f_w(P_1(ab), P_2(ab)) = f_w(b, a) = 1$$

and

$$f_w(P_1(ba), P_2(ba)) = f_w(b, a) = 0,$$

i.e., f_w should assume different values in correspondence to the same arguments. ■

The following result trivially follows from Definition 3.2.

***Proposition** 3.6:* If K is uniformly $q-$observable wrt L and a set of alphabets Σ_i, $i = 1, \ldots, n$, then it is also uniformly $(q-1)$-observable wrt them.

Proof: Follows by the fact that the same f_w function used in the case of uniform $q-$observability can be used in the case of uniform $(q-1)-$observability, simply restricting its arguments to words of length $q-1$ rather than q. □

This implies that, if a language is uniformly $q-$observable for some finite $q > 1$, then it is also uniformly 1-observable.

A simple condition under which uniform 1-observability is guaranteed is now given.

***Proposition** 3.7:* Let us consider a set of alphabets Σ_i, $i = 1, \ldots, n$, such that $\Sigma_1 \cup \ldots \cup \Sigma_n = \Sigma$.

Any language $K \subset L \subseteq \Sigma^*$ is uniformly $1-$observable wrt to L and Σ_i, $i = 1, \ldots, n$.

Proof: Since $\Sigma_1 \cup \ldots \cup \Sigma_n = \Sigma$, there exists at least one site that can detect any event e that has occurred. If the function f_w has been defined for a word w, the new function simply assigns the value 1 if $we \in K$ and 0 otherwise. Being possible to define the function for any observed event, the system is uniformly $1-$observable. □

On the contrary, uniform 1-observability is no more ensured if one or more events in Σ are not observable by all the sites. Let

$$\hat{\Sigma} = \Sigma \setminus \bigcap_{i=1}^{n} \Sigma_i \tag{4}$$

denotes the set of events that are observable by no site. If $\hat{\Sigma} \neq \emptyset$ then $K \subset L \subseteq \Sigma^*$ can be not uniformly 1-observable wrt L and Σ_i's, even if all words formed by the concatenation of a word in K and a word in $\hat{\Sigma}^*$ are still in K, i.e., $K\hat{\Sigma}^* \cap L \subset K$.

A. Regular languages

Particularly interesting results can be proved if K and L are prefix-closed regular languages. First, it can be shown that analyzing uniform q-observability is a decidable problem. Then, a simple criterion can be given to establish if a certain sequence is legal, based on DFA.

***Proposition** 3.8:* Let us consider a set of alphabets Σ_i, $i = 1, \ldots, n$, such that $\Sigma_1 \cup \ldots \cup \Sigma_n = \Sigma$. Let K and L be two prefix-closed languages such that $K \subset L \subseteq \Sigma^*$.

If K and L are *regular* languages, the uniform $q-$observability of K wrt L and Σ_i is decidable for any finite $q \in \mathbb{N}$.

Proof: According to the Myhill-Nerode Theorem [12], each regular language L has a finite *index*, i.e., the set of languages $\{w^{-1}L \mid w \in L\}$ is finite. This implies that it is sufficient to check the existence of a function f_w for a finite number of words w over a finite subset of $\Sigma_1^* \times \cdots \Sigma_n^*$, i.e., the set of projections on Σ_i's, $i = 1, \ldots, n$, with length less than or equal to q. Thus the problem is decidable. □

From the Myhill-Nerode Theorem [12], it follows that to each regular language can be uniquely associated a *minimal* DFA generating it, namely a DFA with the fewest number of states. Now, let L and K be two regular prefix-closed languages, where K represents the *legal* behavior and L represents the set of all possible behaviors, including legal and illegal behavior. Let G_L and G_K be the minimal DFA with generated languages $\mathcal{L}(G_L)$ and $\mathcal{L}(G_K)$, respectively. Being such languages prefix-closed, marked languages coincide with regular languages.

Starting from G_L and G_K, we want to give a procedure to construct a unique DFA H where some states are *good* and others are *bad*. The strings terminating in a good state represent a legal behavior and should belong to K. On the contrary, the strings terminating in a bad state represent the forbidden language, i.e., should belong to $L \setminus K$.

The main steps of the procedure to construct such a DFA can be summarized by Algorithm 1.

Algorithm 1 Construction of the DFA H

Let $G'_K = (X, E, \delta, x_0, X_m)$ be a DFA where X, E, δ and x_0 are the same of G_K and $X_m = \emptyset$.

Add a new marked state to G'_K that has a self-loop containing all events in E.

Add arcs labeled $E \setminus \{e \in E | \delta(x, e)!\}$ from each state $x \in X$ to this new state.

Let $H = G_L \| G'_K$ be the automaton obtained by the parallel composition of automaton G_L and automaton G'_K.

The following property is satisfied by the DFA H built using the above procedure.

***Proposition** 3.9:* Let H be the automaton built according to Algorithm 1, starting from two prefix-closed regular languages K and L.

- All strings that finish in an unmarked state are in K.
- All strings that finish in a marked state are in $L \setminus K$.

Proof: Simply follows from the rules of construction of H using Algorithm 1. □

Note that it can never occur that a string finishes in an unmarked state passing through a marked state. Indeed, by the rules of construction of H, if a string reaches a marked state, all events that follow, never allow the state to be changed.

Uniform $q-$observability can be studied according to Algorithm 2.

***Example** 3.10:* Let L and K be the two languages already considered in Example 3.5, namely, $L = (a + b)^*$ and $K = K_1 + K_2$ where $K_1 = (aa^*b)^*$ and $K_2 = (aa^*b)^* aa^*$.

The DFA in Fig. 1 can be obtained applying Algorithm 1 where G_K is composed by x_0 and x_1 while G_L also includes x_2. Therefore, all strings starting from x_0 avoiding x_2 belong to K. However, if a string finishes in x_2 it belongs to $L \setminus K$, i.e., it is a *bad* word.

To study uniform 1-observability we initially assume $\mathcal{X} = \{x_0, x_1\}$. Let us first focus on x_0. The set of words of unitary length starting from x_0 is $\{a, b\}$: a and b obviously belong to different equivalence classes, i.e., they are distinguishable, thus we continue the algorithm. In particular, we repeat the same reasoning for x_1 and we conclude that K is uniformly 1-observable.

Using similar arguments we conclude that the language K is not uniformly 2-observable. ■

Algorithm 2 Uniform q-observability

Let $\mathcal{X} = \{X \setminus X_m\}$ be the set of unmarked (good) states of H.
while $\mathcal{X} \neq \emptyset$ **do**
  Choose arbitrarily one state $x \in \mathcal{X}$
  $i \leftarrow 1$.
  while $i \leq q$ **do**
    Compute the set of words of length i that can be generated by H starting from x.
    Partition such words in equivalence classes $\mathcal{W}_j$'s.
    if $\exists$ some equivalence class $\bar{\mathcal{W}}$: $\bar{\mathcal{W}} \cap K \neq \emptyset$ but it is not $\bar{\mathcal{W}} \subseteq K$ **then**
      exit. {The language K is not uniformly $q-$observable wrt L and Σ_i's}.
    else
      $i = i + 1$
    end if
  end while
  $\mathcal{X} \leftarrow \mathcal{X} \setminus \{x\}$
end while

IV. $q-$DIAGNOSABILITY

In this section we introduce a new property, strictly related to uniform q-observability, that we denote q-diagnosability. Such a property still concerns the possibility of establishing if a word given by the concatenation of a legal word w, and a word u on which we receive some information, is legal as well. The main difference of q-diagnosability wrt q-observability is on the information on u. We still assume the presence of n observers, each one with its own alphabet, and a coordinator. However, in the case of q-diagnosability observations are sent to the coordination by single sites in the form of a series of a finite number m of synchronized words, rather than a single word.

***Definition** 4.1:* Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \ldots, n$, be n sub-alphabets of Σ. Let L and K be two prefix closed languages such that $K \subset L \subseteq \Sigma^*$.

The language K is called $q-$*diagnosable* wrt L and Σ_i, $i = 1, \ldots, n$, if for all $m \in \mathbb{N}$ and $\forall$ sequence of m words $(u_1, u_2, \ldots, u_m)$ such that $u_1 u_2 \ldots u_m \in L$ and $|u_i| \leq q$, $\forall i = 1, \ldots, m$, it

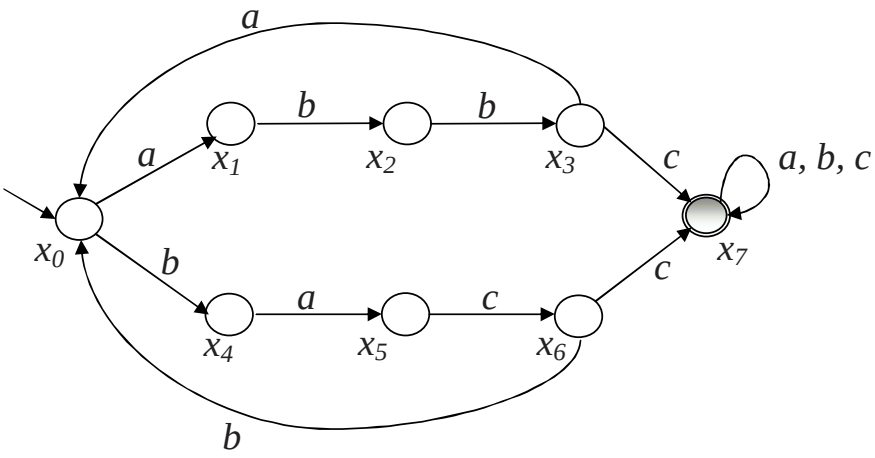

Fig. 2. The DFA considered in Example 4.4 where q_7 is the bad state.

holds

$$\begin{aligned} & u_1 u_2 \dots u_m \in K \iff \\ & f\left(P_1(u_1), \dots, P_n(u_1), \dots,\right. \\ & \qquad \left. \dots, P_1(u_m), \dots, P_n(u_m)\right) = 1. \end{aligned} \tag{5}$$

■

The notion of equivalence can be easily extended to the case of $q-$ diagnosability.

***Definition** 4.2:* Let Σ be a finite alphabet, and $\Sigma_i \subseteq \Sigma$, with $i = 1, \dots, n$, be n sub-alphabets of Σ. Let L and K be two prefix closed languages such that $K \subset L \subseteq \Sigma^*$.

Consider two sequences of word $(u_1, u_2, \dots, u_m)$ and $(v_1, v_2, \dots, v_m)$, where $u_1 u_2 \cdots u_m, v_1 v_2 \cdots v_m \in L$. The two sequences are *diagnosable equivalent*, or simply equivalent, if $P_i(v_j) = P_i(u_j)$ for all $i = 1, \dots, n$ and all $j = 1, \dots, m$. We denote this $(u_1, u_2, \dots, u_m) \equiv (v_1, v_2, \cdots, v_m)$. Finally, we say that two sequences that are not equivalent are *distinguishable*. ■

Obviously, if both languages L and K are regular, by Algorithm 1 the analysis of $q-$diagnosability can be carried out using DFA where final states correspond to bad states, and sequences that terminate in them are not legal.

Moreover, the following implication holds.

***Proposition** 4.3:* If a language K is $q-$diagnosable wrt to a language L and a set of alphabets $\Sigma_1, \dots, \Sigma_n$, then it is also $q-$observable wrt L and $\Sigma_1, \dots, \Sigma_n$.

Proof: It is a consequence of Definitions 3.2 and 4.1. Indeed, consider any word $w \in K$ and write it as $w = u_1 u_2 \cdot u_k$ where $|u_i| \leq q$ for all i.

Then for any word $u \in w^{-1}L$ with $|u| \leq q$ we can define function f_w in Definition 3.2 in terms of function f in Definition 4.1 as follows:

$$\begin{aligned} & f_w\left(P_1(u), \ldots, P_n(u)\right) \\ & = f\left(P_1(u_1), \ldots, P_n(u_1), \ldots, P_1(u_k), \ldots, P_n(u_k),\right. \\ & \qquad \left. P_1(u), \ldots, P_n(u)\right) \end{aligned}$$

showing that K is uniformly $q-$observable wrt L and Σ_i's.

□

On the contrary, q-observability does not imply q-diagnosability as shown by the following example. Although, the results presented above hold for both regular and non regular languages, for the sake of simplicity the following example deals with regular languages.

***Example** 4.4:* Let L be the language generated by the DFA in Fig. 2 where x_0 is the initial state, while K is the language generated by the same DFA with the same initial state, but neglecting x_7, that is the only bad state. Finally, assume three sites with alphabets $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$ and $\Sigma_3 = \{c\}$, respectively.

As shown in the following items, K is uniformly $3-$observable wrt L and Σ_i, $i = 1, 2, 3$.

• Let $w = \varepsilon$. All possible words $u \in w^{-1}L$ with $|u| = 3$ finish in good states, without passing through a bad state. In particular, $u_1 = abb$ terminates in x_3 and $u_2 = bac$ in x_6. Therefore, it is $f_w(P_1(u_1), P_2(u_1), P_3(u_1)) = f_w(a, bb, \varepsilon) = 1$ and $f_w(P_1(u_2), P_2(u_2), P_3(u_2)) = f_w(a, b, c) = 1$.

• Let $w = a$. Two possible sequence of length 3 may follow w, namely $u_3 = bbc \notin w^{-1}K$ and $u_4 = bba \in w^{-1}K$. However $bbc \not\equiv bba$, thus they can be distinguished by the coordinator assuming $f_w(P_1(u_3), P_2(u_3), P_3(u_3)) = f_w(\varepsilon, bb, c) = 0$ and $f_w(P_1(u_4), P_2(u_4), P_3(u_4)) = f_w(a, bb, \varepsilon) = 1$.

• Let $w = b$. As in the above item, there are two sequences of length 3 that can follow w, namely $u_5 = acc \notin w^{-1}K$ and $u_6 = acb \in w^{-1}K$. However, these strings can be distinguished being $acc \not\equiv acb$.

• Let $w = ab$. In this case, there are 4 possible strings of length 3 that may follow w, one in $w^{-1}K$, namely $u_7 = baa$, the other three not in $w^{-1}K$, namely, bca, bcb and bcc. However, the word finishing in a good state can be distinguished by all words finishing in the bad state since it does not contain event c, while all the others do.

• Let $w = ba$. Also in this case the good word (cbb) can be distinguished by the bad ones (cca, ccb, ccc) since it only contains one c, while the others contain at least two c.

• Let $w = abb$. Also in this case the good word can be distinguished by the bad ones since it does not contain c, while all the bad do.

• Let $w = bac$. The same as in previous case: if one c is observed by Σ_3, the coordinator can conclude that the bad state x_7 is reached.

• Note that no other words w need to be considered since the previous ones cover all good states of the DFA. Moreover we do not need to consider words u of length smaller than 3 since by Proposition 3.6 uniform $q-$observability implies uniform $(q-1)$-observability.

Using similar arguments, we can prove that K is not $4-$observable. In particular, the two sequences $abbc \equiv bacb$ may follow $w = \varepsilon$ but $abbc \notin w^{-1}K$ while $bacb \in w^{-1}K$. Thus no function f_w may be defined to distinguish them.

Finally, let us prove that even if K is $3-$observable, it is not $3-$diagnosable. Indeed, let us assume $u_1 = ab$, $u_2 = bc$, $v_1 = ba$ and $v_2 = cb$. Since $P_1(ab) = P_1(ba) = a$, $P_2(ab) = P_2(ba) = b$, $P_3(ab) = P_3(ba) = \varepsilon$, $P_1(bc) = P_1(cb) = \varepsilon$, $P_2(bc) = P_2(cb) = b$ and $P_3(bc) = P_3(cb) = c$ then $v_1v_2 \equiv u_1u_2$. Being $v_1v_2 \in K$ and $u_1u_2 \notin K$ it will be impossible for the coordinator to distinguish among them. ■

The following result provides a useful criterion to the analysis of q-diagnosability.

***Proposition** 4.5:* Let K be $q-$observable wrt a given language L and a set of alphabets Σ_i's. If after any $\hat{q} \leq q$ steps the state is uniquely determined, $q-$observability $\Longrightarrow$ $q-$diagnosability.

Proof: If after $\hat{q} \leq q$ steps the state is uniquely determined and K is $q-$observable it is always possible to say if the concatenated word is in K. Using this argument for a finite number of subsequences, the statement follows.

□

***Example** 4.6:* Let us consider again the case of Example 4.4 whose corresponding DFA H is that reported in Fig. 2. As already proved K is not $3-$diagnosable even if it is $3-$observable.

This result is consistent with Proposition 4.5. Indeed, if we consider $u = ab$, the first site observes a and the second one observes b. Thus the current state is not uniquely determined: both x_2 and x_5 are possible. ■

We finally present the following result.

***Proposition** 4.7:* Let us consider a set of alphabets Σ_i, $i = 1, \ldots, n$, such that $\Sigma_1 \cup \ldots \cup \Sigma_n = \Sigma$. Let K and L be two prefix-closed languages such that $K \subset L \subseteq \Sigma^*$.

If K and L are *regular* languages, the $q-$diagnosability of K wrt L and Σ_i's is decidable for

any finite $q \in \mathbb{N}$.

Proof: We just give a sketch of the proof. Since we are taking into account regular languages we can equivalently speak about a DFA H constructed with Algorithm 1 with state set X. To determine if the property holds for $m = 1$ we need to check all words u_1 of length less than or equal to q that can be generated by the DFA starting from the initial state x_0.

Consider the case $m = 2$. After the first synchronization is performed, we do not know the current state of the DFA but we know it belongs to a set $X(u_1) = X(P_1(u_1), \ldots, P_n(u_1)) \subseteq X$ and the set $\Xi_1 = \{X(u_1) \mid \ u_1 \in K, |u_1| \leq q\}$ is finite. Now, for all possible $X_1 \in \Xi_1$ we consider the the language $L(H \mid X_1) = \cup_{x \in X_1} L(H \mid x)$ where $L(H \mid x)$ denotes the language generated by the automaton with initial state x and we need tho check all words of length less than or equal to q in this language.

As m is increased one may have larger sets Ξ_k to check but eventually $\Xi_k = \Xi_{k+1}$ because for all $k \geq 1$ it holds $\Xi_k \subseteq 2^X$. Hence there are at most $2^{|X|}$ languages $L(H \mid X_k)$ to consider and the problem is decidable. □

V. Dynamic Observability and Diagnosability

In this section we focus on regular prefix-closed languages and consider a problem that may occur in several real applications. We assume that the actual state of the system is known, and we want to develop an algorithm to determine the instants at which it is necessary to synchronize the observations coming from the different sites, so that the bad state is identified exactly *as soon as* it is reached. Obviously, the last instant at which synchronization occurs should be equal to the length of the shortest path (denoted by k) from the actual state to a bad state. Furthermore, according to Proposition 4.5, the state in which the system is after k steps should be uniquely determined such that it is still possible to perform diagnosis.

The proposed algorithm is also based on the following quite intuitive result.

***Proposition** 5.1:* Two consecutive synchronization performed after q_1 and q_2 steps, respectively, lead to a number of consistent words/states smaller than a unique synchronization after $q_1 + q_2$ steps.

Proof: Follows from the trivial consideration that an intermediate additional synchronization can only lead to additional information, thus to a reduced number of consistent words/states. □

Let us now consider two regular languages L and $K \subset L$ generated by two DFA G_L and G_K, respectively. Given an initial state, Algorithm 3 computes the instants at which it is necessary to synchronize to guarantee that a bad state is identified exactly in the instant in which it is reached, and in the case that no bad state is reached after a number of steps equal to the length of the shortest path from the current state to a bad state, the new state is uniquely identified.

Algorithm 3 Synchronization

$k \leftarrow$ length of the shortest path from the actual state to a bad state.

Let $\mathcal{I} = \{k\}$ be the set of indices of steps at which we have to synchronize.

Compute all words of length k and split them in equivalence classes $\mathcal{W}_j$ with the same projections on Σ_i, $\forall i = 1, \ldots, n$.

while $|\mathcal{W}_j| \neq 1$ for all j **do**

Choose randomly one word $\bar{w} \in \bar{\mathcal{W}}$

Compute an index $p \leq |\bar{w}|$ such that if a new intermediate synchronization occurs after p steps, $\bar{w}$ will not be equivalent to any word in $\bar{\mathcal{W}} \setminus \{\bar{w}\}$.

$\mathcal{I} \leftarrow \mathcal{I} \cup \{p\}$.

Update the set of equivalence classes $\mathcal{W}_j$ taking into account the new synchronization.

end while

***Remark** 5.2:* Algorithm 3 ensures that the set of consistent states after the last synchronization at the k-th step is a singleton, i.e., the actual state of the system is known after the last synchronization. This is a trivial consequence of the fact, that after k steps all equivalence classes are singleton. On the contrary, the set of consistent states after the intermediate synchronization is in general not a singleton.

■

***Example** 5.3:* Let us apply Algorithm 3 to the DFA in Fig. 3 assuming $\Sigma_1 = \{a\}$, $\Sigma_2 = \{b\}$ and x_{12} as the bad state.

• The length of the shortest path from x_0 to the bad state x_{12} is $k = 5$. Hence, we intially take $\mathcal{I} = \{5\}$.

• The set of strings of length $k = 5$ starting from x_0 is $\{abbaa, abbab, bbaaa, bbaab, baaba\}$. Therefore, we can define two equivalence classes: $\mathcal{W}_1 = \{abbaa, bbaaa, baaba\}$ and $\mathcal{W}_2 =$

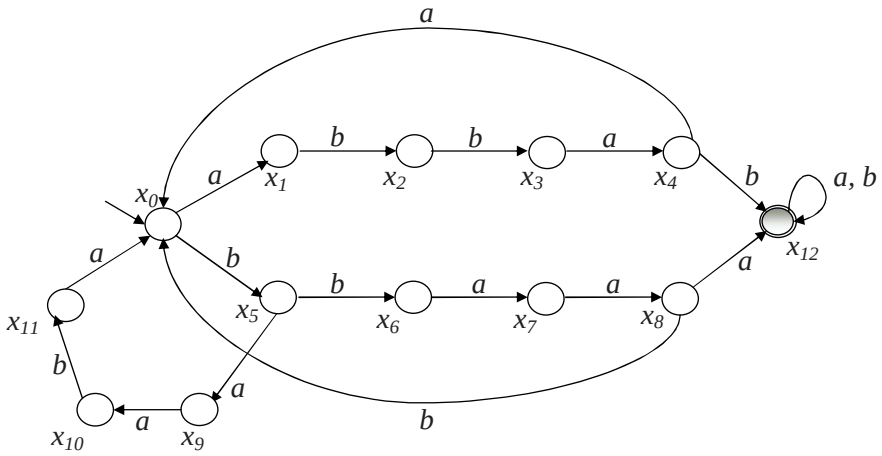

Fig. 3. The DFA considered in Example 5.3 where x_{12} is the bad state.

$\{abbab, bbaab\}$. In fact, $P_1(abbaa) = P_1(bbaaa) = P_1(baaba) = aaa$ and $P_2(abbaa) = P_2(bbaaa) = P_2(baaba) = bb$.

- We randomly choose an equivalence class with cardinality greater than 1, e.g., $\bar{\mathcal{W}} = \mathcal{W}_1$.
- We randomly choose $\bar{w} = abbaa$ and consider $p = 1$. Thus $\bar{w} = \bar{u}\bar{v}$ with $\bar{u} = a$ and $\bar{v} = bbaa$. Indeed, $P_1(\bar{u}) = a$ while the projection of the first event of all other sequences in $\bar{\mathcal{W}}$ is equal to the empty string, thus the new synchronization makes $\bar{w}$ not equivalent to all the sequences in $\bar{\mathcal{W}} \setminus \{\bar{w}\}$.
- Let $\mathcal{I} = \{1, 5\}$.
- The new equivalence classes assuming synchronization at steps 1 and 5 are: $\mathcal{W}_1' = \{abbaa\}$, $\mathcal{W}_1'' = \{bbaaa, baaba\}$, $\mathcal{W}_2' = \{abbab\}$ and $\mathcal{W}_2'' = \{bbaab\}$.
- We randomly choose a new equivalence class of cardinality greater than 1, e.g., $\bar{\mathcal{W}} = \mathcal{W}_1''$.
- We randomly choose $\bar{w} = bbaaa$ and consider $p = 2$.
- Let $\mathcal{I} = \{1, 2, 5\}$.
- It is easy to verify that the new equivalence classes are singleton and the algorithm stops.

Therefore, starting from x_0, in order to be able to uniquely identify the state after 5 steps (the length of the shortest path to the bad state x_{12}) two additional synchronization should be performed. One after one step, the second one after one more step, and the last third one after 3 further steps.

Let us remark that this does not imply that after the two intermediate steps the state is uniquely

determined, while this is ensured after the last synchronization at step 5.

At step 5, the algorithm should be run again considering as initial state the new one that has been actually reached after the occurrence of 5 events. ■

VI. Conclusions

This paper deals with the problem of establishing if a given behavior is legal, based on decentralized observation performed by a finite number of sites, who are only able to observe a subset of the possible events. The sites transmit their observation to a coordinator who takes the decision concerning legacy of the occurred word. Two different properties have been defined, namely $q-$observability and $q-$diagnosability, that differ for the criterion used to synchronize the different sites. Finally, an algorithm to compute the instants in which synchronization should occur, assuming that the initial state is known, has been given. It guarantees that the occurrence of the an illegal word is detected as soon as it has occurred.

References

[1] S. Tripakis, "Undecidable problems of decentralized observation and control on regular languages," *Information Processing Letters*, vol. 90, no. 1, pp. 21–28, 2004.

[2] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of Discrete-Event Systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

[3] P. E. Caines, R. Greiner, and S. Wang, "Dynamical logic observers for finite automata," in *Proc. of* 27^{th} *Conference of Decision and Control*, 1988, pp. 226–233.

[4] P. E. Caines and S. Wang, "Classical and logic based regulator design and its complexity for partially observed automata," in *Proc. of* 28^{th} *Conference on Decision and Control*, 1989, pp. 132–137.

[5] C. M. Özveren and A. S. Willsky, "Observability of discrete event dynamic systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 7, p. 797806, 1990.

[6] R. Kumar, V. Garg, and S. I. Markus, "Predicates and predicate transformers for supervisory control of discrete event dynamical systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, p. 232247, 1993.

[7] A. Saboori and C. Hadjicostis, "Opacity-enforcing supervisory strategies for secure discrete event systems," in *Proc. of the 47th IEEE Conference on Decision and Control*, 2008, pp. 889–894.

[8] A. Saboori and C. N. Hadjicostis, "Opacity verification in stochastic discrete event systems," in *Proc. of the 49th IEEE Conference on Decision and Control*, 2010, pp. 6759–6764.

[9] J. Dubreil, P. Darondeau, and H. Marchand, "Supervisory control for opacity," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1089–1100, 2010.

[10] G. Barrett and S. Lafortune, "Decentralized supervisory control with communicating controllers," *IEEE Trans. on Automatic Control*, vol. 45, no. 9, pp. 1620 –1638, Sept. 2000.

[11] S. Ricker and B. Caillaud, "Mind the gap: Expanding communication options in decentralized discrete-event control," in *46th IEEE Conf. on Decision and Control*, December 2007, pp. 5924 –5929.

[12] J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to Automata Theory, Languages and Computation (Third Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.