IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2020, accepted November 26, 2020, date of publication December 4, 2020, date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3041478

New Results on Deterministic Networked Supervisory Control and Relative Delay Observability

YUNFENG HOU, YUTAO CHEN, QINGDU LI, AND YUNFENG JI, (Associate Member, IEEE)
Institute of Machine Intelligence, University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding author: Yunfeng Ji (ji_yunfeng@usst.edu.cn)

This work was supported in part by the Shanghai Pujiang Program under Grant 2019PJC073, in part by the Interdisciplinary Innovation Center at Shanghai Institute of Higher Learning, and in part by the Program for Industrialization of the Next Generation of Intelligent Humanoid Robots.

ABSTRACT This paper considers supervisory control of discrete event systems with observation delays and control delays, which is termed deterministic networked supervisory control. Delay observability, together with delay controllability, is required for obtaining a networked supervisor so that the language generated by the supervised system is deterministic and is equal to a given specification language. In this article, we simplify this existence condition and prove such a networked supervisor also exists if controllability and delay observability are satisfied. A new language property called relative delay observability, which accommodates observation delays in relative observability, is further defined. Relatively delay observability implies delay observability and enjoys an important property that is closed under union. We then show how to check the relative delay observability in polynomial time complexity. A general algorithm used to calculate the supremal sublanguage satisfying controllability and another user-defined language property in a solution space of interest, is presented. As a special case, we calculate the supremal controllable and relatively delay observable sublanguage. Finally, a practical example is given to illustrate the application of relative delay observability.

INDEX TERMS Deterministic networked supervisory control, discrete event systems, delay observability, relative delay observability.

I. INTRODUCTION

A. MOTIVATION

In the framework of supervisory control proposed by [2], supervisors are used to control the system to ensure the language generated by the controlled system satisfies a given specification language. The previous work shows that such a supervisor exists if and only if the specification language is controllable [2] and observable [3]. However, there does not exist the supremal controllable and observable sublanguage of a given specification language in general since observability is not closed under (set) union. To deal with this undesirable situation, language properties that are stronger than observability are identified, which include, for example, normality [3], [4], weak normality [5], strong observability [6], and relative observability [7], [8]. All the above works assume that the communications between the plant and supervisor are instantaneous, i.e., if an event occurs, then it can be immediately sensed by the supervisor, and the control commands issued by the supervisor can be executed by the actuator of the plant without any delays.

However, the assumption is not true in networked discrete event systems (NDES) [1], [9]–[12], where the communications from a plant to a supervisor for observation or a supervisor to a plant for control are sent over a shared communication network. Due to the network characteristics, the communication network may deliver the sensor signals or the control commands at their destination with a random delay. Moreover, the sensor signals or the control commands may be lost during the transformation. As the network has been widely used in the industry, it is very necessary to study the impact of communication delays on supervisory control.

The associate editor coordinating the review of this manuscript and approving it for publication was Azwirman Gusrialdi.

B. RELATED WORKS

The purpose of supervisory control for NDES (networked supervisory control) is to calculate a networked supervisor to disable the event occurrences that lead to some undesired event sequences under communication delays and losses. This problem was systematically investigated by Lin in [9]. Based on this fundamental work, the networked supervisory control problem has been extensively studied in the past few years. It is shown in [12] that such a networked supervisor exists if both controllability and network observability [9] are satisfied. The networked supervisory control problem is further investigated in [10], where more than one networked supervisors are used to control the networked system. Reference [11] discusses how to calculate a maximally-permissive networked supervisor such that the language generated by the supervised system contains a minimum required language but never exceeds a maximal admissible (legal) language. The authors in [13] show how to synthesize a safe networked control policy on-the-fly with control delays. The networked supervisory control problem for timed NDES is considered in [14], [15]. All the above works assume the communications are carried out over a single channel. Assuming the plant communicates the event occurrences to a centralized supervisor via several communication channels, reference [16] solves the problem of state estimation under observation delays. Under the same assumption as in [16], the authors in [17] further investigate how to determine all the possible initial states that the system may start from based on the current observations, which is called the problem of initial state estimation. Reference [18] discusses how to implement supervisory control of NDES with a timing structure that considers both observation delays and losses. The authors in [18] also assume the observable event occurrences are communicated to the networked supervisor via several communication channels. Moreover, references [19], [20] investigate how to use a group of agents to diagnosis a system with observation delays.

Due to the arbitrary delays in both observation and control, the supervised system can generate many languages nondeterministically. As in [1], [10], [11], the upper bound and lower bound on all possible languages that may be generated by the supervised system is denoted as the large language and the small language, respectively. The deterministic networked control problem for discrete event systems with nondeterministic communication delays is formulated in [1] as calculating a networked supervisor such that the language generated by a supervised system is deterministic, i.e., the large language is equal to the small language and is equal to the specification language. The necessary and sufficient conditions for the existence of a deterministic networked supervisor are termed delay controllability and delay observability [1]. Delay controllability requires that if the occurrence of an event needs to be disabled, that event must be controllable and all the possible controls must disable it. Delay observability requires that if the disablements/enablements after two sequences of events are different, all the possible observations of the two sequences of events must be different.

When the given specification language cannot be exactly obtained via networked control under possible communication delays, computing the supremal delay controllable and delay observable sublanguage is desired. However, it is shown in [1] that this is too difficult with the notions of delay controllability and delay observability since both delay controllability and delay observability are not closed under union.

C. MAIN GOAL

We inherit two key assumptions from [1] in this paper. First, we assume that the delays do not change the order of the observations, i.e., the observation events arrive in the same order as they were generated, commonly known as first-in-first-out (FIFO). Second, we assume that there are no communication losses, i.e., all the occurrences of observable events can be sensed by the supervisor. The main contributions of this paper are threefold.

First, we simplify the existence condition for the deterministic networked supervisor introduced in [1]. Specifically, we prove that the deterministic networked control problem is solvable if and only if a given specification language is controllable and delay observable. The main advantage of the simplification is that controllability is algebraically well-behaved than delay controllability: controllability is closed under union while delay controllability is not.

Second, to overcome the difficulty that delay observability is not preserved under union, we introduce, in this paper, a new language property called relative delay observability, which is an extension of relative observability in the case of observation delays. If there are no observation delays, relative delay observability is reduced to relative observability. We show that relative delay observability implies delay observability and is closed under union. To the best of our knowledge, there does not exist another language property that is stronger than delay observability, weaker than relative delay observability, and closed under union.

Finally, it is often the case that a given specification language does not satisfy controllability and some user-defined language property (for example, normality). We present a general algorithm in this paper to calculate the supremal sublanguage that satisfies controllability and another user-defined language property in a finite solution space of interest. We show such a supremal sublanguage always exists if the user-defined language property is closed under union, and there exists at least one element in the solution space satisfying controllability and the user-defined property. As a special case, we calculate the supremal controllable and relatively delay observable sublanguage. Since relative delay observability implies delay observability, a deterministic networked supervisor that synthesizes the supremal controllable and relatively delay observable sublanguage always exists.

The rest of this paper is organized as follows. In Section 2, the deterministic networked control theory for discrete event systems is reviewed. In Section 3, we introduce the definition of relative delay observability and discuss some of its properties. In Section 4, we show how to check relative delay observability, and an algorithm is developed to calculate the supremal controllable and relatively delay observable sublanguage. Section 5 presents a practical example to illustrate the application of theories introduced in this paper. Section 6 concludes this paper.

II. DETERMINISTIC NETWORKED CONTROL

A. PRELIMINARIES

A discrete event system (DES) is modeled by a deterministic finite-state automaton $G = (Q, \Sigma, \delta, \Gamma, q_0, Q_m)$, where Q is the set of states; Σ is the set of events; $\delta : Q \times \Sigma \to Q$ is the transition function; $\Gamma : Q \to 2^\Sigma$ is the set of active events; Q_m is the set of marked states. The language generated by G is denoted by $\mathcal{L}(G)$. The marked language generated by G is given as $\mathcal{L}_m(G)$.

Σ^* is the Kleene closure of Σ. In other words, Σ^* contains all the sequences over events in Σ. δ is extended to $Q \times \Sigma^*$ in the usual way. $\mathbb{N}$ is the natural number set. Given $n \in \mathbb{N}$, let $\Sigma^{\leq N}$ be the set of all sequences in Σ^* with a length no larger than N.

In general, not all the events in Σ are controllable and observable. We often use $\Sigma_c \subseteq \Sigma$ to denote the set of controllable events and $\Sigma_{uc} = \Sigma \setminus \Sigma_c$ to denote the set of uncontrollable events. Meanwhile, we also use $\Sigma_o \subseteq \Sigma$ to denote the set of observable events and $\Sigma_{uo} = \Sigma \setminus \Sigma_o$ to denote the set of unobservable events. The natural projection $P : \mathcal{L}(G) \to \Sigma_o^*$ is defined, for all $s, s\sigma \in \mathcal{L}(G)$, as:

$$P(\varepsilon) = \varepsilon, \; P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{if } \sigma \in \Sigma_{uo} \end{cases}$$

Given G_1 and G_2, we say G_1 is a sub-automaton of G_2, denoted by $G_1 \sqsubseteq G_2$, if G_1 can be obtained from G_2 by deleting some states in G_2 and all the transitions connected to these states. $Ac(G)$ denotes the accessiable part of G [21].

Given a sequence s, $\overline{s} = \{s' : (\exists s'')s = s's''\}$ is the set of all prefixes of s. The prefix-closure of a language $L \subseteq \Sigma^*$ is denoted as $\overline{L}$. We say L is $\mathcal{L}_m(G)$-closed if $L = \overline{L} \cap \mathcal{L}_m(G)$. L is said to be prefix-closed if $L = \overline{L}$. For brevity, only prefix-closed languages are considered in this paper and all the results derived in this paper still hold for non-closed languages. $|s|$ is the length of s. Let s_{-i} be the prefix of s satisfying $|s_{-i}| = \max\{0, |s| - i\}$.

The cardinality of an array or a set A is denoted as $|A|$, Given a set of sets M, the union of all sets in M is denoted by $\bigcup M$.

B. DETERMINISTIC NETWORKED CONTROL

As shown in Fig. 1, in NDES, communications between the supervisor and the plant are carried out over some shared networks so that the communication delays in both the observation channel (from the plant to the supervisor) and the control channel (from the supervisor to the plant) are unavoidable. Communication delays in both observation channel (called observation delays) and control channel (called control delays) are random. The assumptions made in this paper are inherited from [1] as follows:

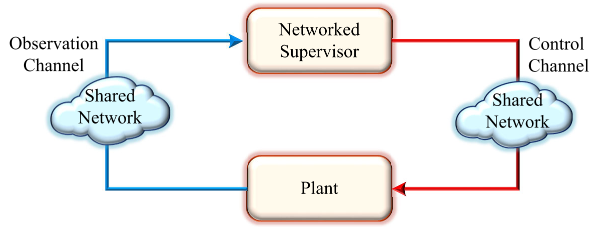


FIGURE 1. Networked supervisory control.

1) The observation delays are upper bounded by N_o event occurrences and the control delays are upper bounded by N_c event occurrences;
2) The delays do not change the order of observations, i.e., FIFO is satisfied in the observation;
3) There are no observation and control losses;
4) The actuator of the plant always uses the most recently received command;
5) The initial control command can be executed without any delays.

By assumption 1), the observation delays are upper bounded by N_o event occurrences. That is to say, an observable event occurrence can always be delivered to the networked supervisor before no more than N_o additional event occurrences. By assumption 2), the control delays are upper bounded by N_c event occurrences. That is to say, an issued control command can be executed by the actuator of the plant before no more than N_c additional event occurrences. Moreover, since the control command initially issued can be executed beforehand, as in assumption 5), we assume the initial control command can be executed without any delays.

Suppose a sequence s occurs in G, what the networked supervisor may see is nondeterministic because of observation delays. Formally, the set of all the possible observations after the occurrence of s is denoted by

$$\Theta_D^{N_o}(s) = \{P(t) : (\exists m \leq N_o)t = s_{-m}\}.$$

We use $\theta(s)$ to denote some element in $\Theta_D^{N_o}(s)$, i.e., $\theta(s) \in \Theta_D^{N_o}(s)$.

Let $K \subseteq \mathcal{L}(G)$ be the specification language for a given control objective. We assume, without loss of generality (w.l.o.g.), K can be generated by a sub-automaton $H = (Q_H, \Sigma, \delta_H, \Gamma_H, q_0) \sqsubseteq G$ of G. The control objective is achieved using a networked supervisor $\pi : \Sigma^* \times \Sigma_o^* \to 2^\Sigma$, where $\pi(s, \theta(s))$ is the set of events to be enabled when a sequence s occurs and the networked supervisor sees $\theta(s) \in \Theta_D^{N_o}(s)$.

Any two sequences that appear identical must be followed by the same control command. Correspondingly, π is said to be observation feasible if $(\forall (s,\theta(s)),(s',\theta(s')) \in \Sigma^* \times \Sigma_o^*)[\theta(s)=\theta(s') \Rightarrow \pi(s,\theta(s))=\pi(s',\theta(s'))]$. Moreover, since only the controllable events can be dynamically disabled, π is said to be control feasible if $(\forall (s,\theta(s)) \in \Sigma^* \times \Sigma_o^*)\Sigma_{uc} \subseteq \pi(s,\theta(s))$. To guarantee a networked control policy to be practically feasible, π must be observation feasible and control feasible.

As in [1], the lower bound (or small language) and upper bound (or large language) of languages generated by the supervised system under communication delays are defined as follows.

Given system G with observation delays bounded by N_o as well as control delays bounded by N_c, the small language $\mathcal{L}_r(\pi/G)$ generated by the supervised system is defined iteratively as

$$\begin{aligned}&\varepsilon \in \mathcal{L}_r(\pi/G),\ s\sigma \in \mathcal{L}_r(\pi/G) \Leftrightarrow s \in \mathcal{L}_r(\pi/G)\\ &\quad \wedge s\sigma \in \mathcal{L}(G) \wedge [(\forall m \le N_c)(\forall \theta(s_{-m}) \in \Theta_D^{N_o}(s_{-m}))\\ &\quad \sigma \in \pi(s_{-m},\theta(s_{-m}))],\end{aligned}$$

and the large language $\mathcal{L}_a(\pi/G)$ generated by the supervised system is defined iteratively as

$$\begin{aligned}&\varepsilon \in \mathcal{L}_a(\pi/G),\ s\sigma \in \mathcal{L}_a(\pi/G) \Leftrightarrow s \in \mathcal{L}_a(\pi/G)\\ &\quad \wedge s\sigma \in \mathcal{L}(G) \wedge [(\exists m \le N_c)(\exists \theta(s_{-m}) \in \Theta_D^{N_o}(s_{-m}))\\ &\quad \sigma \in \pi(s_{-m},\theta(s_{-m}))].\end{aligned}$$

Recall that K is controllable [2] with respect to (w.r.t.) Σ_{uc} and G if

$$\begin{aligned}&(\forall s \in \Sigma^*)(\forall \sigma \in \Sigma)s \in K \wedge s\sigma \in \mathcal{L}(G) \wedge s\sigma \notin K\\ &\Rightarrow \sigma \in \Sigma_c.\end{aligned}$$

Recall that K is network observable [9] w.r.t. Θ_D^N and $\mathcal{L}(G)$ with $N = N_c + N_o$ if

$$\begin{aligned}&(\forall \sigma \in \Sigma)(\forall s \in K)s\sigma \in K \Rightarrow [(\exists t \in \Theta_D^N(s))\\ &(\forall s' \in (\Theta_D^N)^{-1}(t))s' \in K \wedge s'\sigma \in \mathcal{L}(G) \Rightarrow s'\sigma \in K],\end{aligned}$$

where $(\Theta_D^N)^{-1}$ is the inverse mapping: $(\Theta_D^N)^{-1}(t) = \{s' \in \mathcal{L}(G) : t \in \Theta_D^N(s')\}$.

The necessary and sufficient conditions for the existence of a networked supervisor π such that $\mathcal{L}_a(\pi/G) = K$ is characterized by controllability and network observability, as shown in [12]. To overcome the difficulty that network observability is not closed under union, the notion of relative nework observability [22] that is stronger than network observability but closed under union was introduced as follows.

Given specification language K, ambinent language C, and $\mathcal{L}(G)$, K is relatively network observable [22] w.r.t. C, Θ_D^N and $\mathcal{L}(G)$ with $N = N_c + N_o$ if

$$\begin{aligned}&(\forall \sigma \in \Sigma)(\forall s \in K)s\sigma \in K \Rightarrow [(\exists t \in \Theta_D^N(s))\\ &(\forall s' \in (\Theta_D^N)^{-1}(t))s' \in C \wedge s'\sigma \in \mathcal{L}(G) \Rightarrow s'\sigma \in K].\end{aligned}$$

As stated in [1], controllability and network observability only ensure the upper bound of languages generated by the supervised system can never exceed the specification language K. The supervised system can generate many languages (nondeterministically) between the small language and the large language. To eliminate the nondeterminism and ensure the language generated by the supervised system is deterministic and is equal to K, i.e., $\mathcal{L}_r(\pi/G) = \mathcal{L}_a(\pi/G) = K$, delay controllability and delay observability are introduced in [1] as follows.

Recall that K is delay controllable [1] w.r.t. N_c, Σ_{uc}, and $\mathcal{L}(G)$ if

$$\begin{aligned}&(\forall \sigma \in \Sigma)(\forall s \in K)s \in K \wedge s\sigma \in \mathcal{L}(G) \wedge s\sigma \notin K\\ &\Rightarrow (\forall s' \in \Sigma^{\le N_c})(\forall k \le N_c)s_{-k}s'\sigma \notin K \wedge \sigma \in \Sigma_c.\end{aligned}$$

When there are no control delays, i.e., $N_c = 0$, delay controllability reduces to controllability.

Recall K is delay observable [1] w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G)$ if

$$\begin{aligned}&(\forall \sigma \in \Sigma)(\forall s,s' \in K)s\sigma \in \mathcal{L}(G) \wedge s'\sigma \in \mathcal{L}(G)\\ &\wedge s\sigma \in K \wedge s'\sigma \notin K \Rightarrow \Theta_D^{N_o}(s) \cap \Theta_D^{N_o}(s') = \emptyset.\end{aligned} \tag{1}$$

When there are no observation delays, i.e., $N_o = 0$, delay observability reduces to observability [3], i.e.,

$$\begin{aligned}&(\forall \sigma \in \Sigma)(\forall s,s' \in K)s\sigma \in K \wedge s'\sigma \in \mathcal{L}(G)\\ &\wedge P(s) = P(s') \Rightarrow s'\sigma \in K.\end{aligned}$$

Next, let us recall some notations in [1]. Given automata G and $H \sqsubseteq G$ with $\mathcal{L}(H) = K$, the set of all states in H reachable from q in N_c steps is denoted by $\mathrm{RCH}(q,N_c) = \{q' \in Q_H : (\exists s \in \Sigma^{\le N_c})q' = \delta_H(q,s)\}$, where "RCH" means "reachable". The set of events which must be enabled at some states in $\mathrm{RCH}(q,N_c)$ is denoted as $\mathrm{EN}(q,N_c) = \cup_{q' \in \mathrm{RCH}(q,N_c)}\Gamma_H(q')$, where "EN" means "enable". The set of events which must be disabled at some states in $\mathrm{RCH}(q,N_c)$ is defined as $\mathrm{DIS}(q,N_c) = \cup_{q' \in \mathrm{RCH}(q,N_c)}\Gamma(q') \setminus \Gamma_H(q')$, where "DIS" means "disable". The set of indistinguishable state pairs in H when there exist observation delays is given by $\mathrm{SP}_{IN}(K,N_o) = \{(q,q') \in Q_H \times Q_H : (\exists s,s' \in K)q = \delta_H(q_0,s) \wedge q' = \delta_H(q_0,s') \wedge \Theta_D^{N_o}(s) \cap \Theta_D^{N_o}(s') \neq \emptyset\}$, where "SP" means "state pairs" and "*IN*" means "indistinguishable".

To make $\mathrm{EN}(q,N_c)$ and $\mathrm{DIS}(q,N_c)$ independent of $\mathrm{RCH}(q,N_c)$, reference [1] extends G to an argumented automaton $G_{N_c}^{aug} = (Q \cup \{q_{dis}\}, \Sigma, \delta^{aug}, \Gamma^{aug}, q_0)$ such that for all $q \in Q$, $\delta^{aug}(q,\sigma) = \delta(q,\sigma)$ if $\sigma \in \Gamma(q)$ and for all $q \in Q_H$, $\delta^{aug}(q,\sigma) = q_{dis}$ if $\sigma \in (\cup_{q' \in \mathrm{RCH}(q,N_c)}\Gamma(q')) \setminus \Gamma(q)$. Similarly, H is extended to an argumented automaton $H_{N_c}^{aug} = (Q_H \cup \{q_{dis}\}, \Sigma, \delta_H^{aug}, \Gamma_H^{aug}, q_0)$ such that for all $q \in Q_H$, $\delta_H^{aug}(q,\sigma) = \delta_H(q,\sigma)$ if $\sigma \in \Gamma_H(q)$ and for all $q \in Q_H$, $\delta_H^{aug}(q,\sigma) = q_{dis}$ if $\sigma \in \mathrm{EN}(q,N_c) \setminus \Gamma_H(q)$. The language generated by $H_{N_c}^{aug}$ is denoted by $K_{N_c}^{aug}$, i.e., $\mathcal{L}(H_{N_c}^{aug}) = K_{N_c}^{aug}$.

Theorem 1: Given system G with observation delays bounded by N_o and control delays bounded by N_c, there exists a networked supervisor π such that $\mathcal{L}_r(\pi/G) = \mathcal{L}_a(\pi/G) = K$ if and only if K is delay controllable w.r.t. N_c, Σ_{uc}, and $\mathcal{L}(G)$, and $K_{N_c}^{aug}$ is delay observable w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G_{N_c}^{aug})$.

Theorem 1 was proven in [1]. Both delay controllability and delay observability are not closed under union.

C. SIMPLICATION OF THE EXISTENCE CONDITION

In this section, we prove that the existence condition of a networked supervisor π such that $\mathcal{L}_r(\pi/S) = \mathcal{L}_a(\pi/S) = K$ as shown in Theorem 1 can be simplified into K is controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$, and delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$ with $N = N_c + N_o$.

Due to control delays, the control action issued when the system is in a state $q \in Q_H$ can take effects when the system in any state in $\mathrm{RCH}(q, N_c)$. Correspondingly, to guarantee any legal transitions to be enabled, the supervisor can only disable an event $\sigma \in \Gamma^{aug}(q) \setminus \Gamma_H^{aug}(q)$. But, to guarantee the system remaining in K, the supervisor needs to disable all events in $\mathrm{DIS}(q, N_c)$ when the system is in q. All the above leads to Theorem 2 and Proposition 4 of [1], which are reproduced as Lemmas 1 and 2 as follows.

Lemma 1: $K = \mathcal{L}(H)$ is delay controllable w.r.t. N_c, Σ_{uc}, and $\mathcal{L}(G)$ iff (1) K is controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$, and (2) $(\forall q \in Q_H)\mathrm{EN}(q, N_c) \cap \mathrm{DIS}(q, N_c) = \emptyset$.

Lemma 2: If $K = \mathcal{L}(H)$ is delay controllable w.r.t. N_c, Σ_{uc}, and $\mathcal{L}(G)$, then $K_{N_c}^{aug}$ is delay observable w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G_{N_c}^{aug})$ iff $(\forall (q, q') \in \mathrm{SP}_{IN}(K, N_o))\mathrm{EN}(q, N_c) \cap \mathrm{DIS}(q', N_c) = \emptyset$.

Combining these two lemmas gives us Proposition 1.

Proposition 1: $K = \mathcal{L}(H)$ is delay controllable w.r.t. N_c, Σ_{uc} and $\mathcal{L}(G)$, and $K_{N_c}^{aug}$ is delay observable w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G_{N_c}^{aug})$ iff K is controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$, and delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$ with $N = N_c + N_o$.

Proof:

Let us define the following relations:

1) C_1:=K is delay controllable w.r.t. N_c, Σ_{uc} and $\mathcal{L}(G)$;
2) C_2:=$K_{N_c}^{aug}$ is delay observable w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G_{N_c}^{aug})$;
3) C_3:=K is controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$;
4) C_4:=K is delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$ with $N = N_c + N_o$.

($\Leftarrow$) We prove $C_3 \wedge C_4 \Rightarrow C_1 \wedge C_2$ by proving $\neg C_1 \vee \neg C_2 \Rightarrow \neg C_3 \vee \neg C_4$.

We first prove $\neg C_1 \Rightarrow \neg C_3 \vee \neg C_4$. Since K is not delay controllable w.r.t. N_c, Σ_{uc} and $\mathcal{L}(G)$, by Lemma 1, K is not controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$ or $(\forall q \in Q_H)\mathrm{EN}(q, N_c) \cap \mathrm{DIS}(q, N_c) \neq \emptyset$.

If K is not controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$, we have $\neg C_3$ is ture, which implies $\neg C_3 \vee \neg C_4$ is true. Hence, $\neg C_1 \Rightarrow \neg C_3 \vee \neg C_4$. On the other hand, if $(\forall q \in Q_H)\mathrm{EN}(q, N_c) \cap \mathrm{DIS}(q, N_c) \neq \emptyset$, Since H is accessible, there exists $s \in K$ and $\sigma \in \Sigma$ such that $q = \delta_H(q_0, s)$ and $\sigma \in \mathrm{DIS}(q, N_c) \cap \mathrm{EN}(q, N_c)$. By the definitions of $\mathrm{DIS}(\cdot)$ and $\mathrm{EN}(\cdot)$, there exist $s', s'' \in \Sigma^{\leq N_c}$ such that $(ss', ss'', ss''\sigma \in K) \wedge (ss'\sigma \in \mathcal{L}(G) \setminus K)$. Then, since $s', s'' \in \Sigma^{\leq N_c}$ and $N = N_o + N_c$, we have $P(s) \in \Theta_D^N(ss') \cap \Theta_D^N(ss'')$. Overall,

$$(\exists \sigma \in \Sigma)(\exists ss', ss'', ss''\sigma \in K)ss'\sigma \in \mathcal{L}(G) \setminus K$$
$$\wedge \Theta_D^N(ss') \cap \Theta_D^N(ss'') \neq \emptyset,$$

which implies K is not delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$, i.e., $\neg C_4$ is ture. Hence, we also have $\neg C_1 \Rightarrow \neg C_3 \vee \neg C_4$.

Next, we prove $\neg C_2 \Rightarrow \neg C_3 \vee \neg C_4$. Since $K_{N_c}^{aug}$ is not delay observable w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G_{N_c}^{aug})$. By Lemma 2, there exists $(q, q') \in \mathrm{SP}_{IN}(K, N_o)$ such that $\mathrm{EN}(q, N_c) \cap \mathrm{DIS}(q', N_c) \neq \emptyset$. Since H is accessible, there exists $s, s' \in K$ and $\sigma \in \Sigma$ such that $q = \delta_H(q_0, s)$, $q' = \delta_H(q_0, s')$, $\Theta_D^{N_o}(s) \cap \Theta_D^{N_o}(s') \neq \emptyset$, and $\sigma \in \mathrm{EN}(q, N_c) \cap \mathrm{DIS}(q', N_c)$. Hence, there exists $s'', s''' \in \Sigma^{\leq N_c}$ such that $ss'', ss''\sigma, s's''' \in K$ and $s's'''\sigma \in \mathcal{L}(G) \setminus K$. Since $\Theta_D^{N_o}(s) \cap \Theta_D^{N_o}(s') \neq \emptyset$, $s'', s''' \in \Sigma^{\leq N_c}$, and $N = N_c + N_o$, we have $\Theta_D^N(ss'') \cap \Theta_D^N(s's''') \neq \emptyset$. Overall,

$$(\exists \sigma \in \Sigma)(\exists ss'', ss''\sigma, ss''' \in K)s's'''\sigma \in \mathcal{L}(G) \setminus K$$
$$\wedge \Theta_D^N(ss'') \cap \Theta_D^N(s's''') \neq \emptyset,$$

which says K is not delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$, i.e., $\neg C_4$ is ture. Hence, $\neg C_2 \Rightarrow \neg C_3 \vee \neg C_4$. Overall, we have $\neg C_1 \vee \neg C_2 \Rightarrow \neg C_3 \vee \neg C_4$.

($\Rightarrow$) We prove $C_1 \wedge C_2 \Rightarrow C_3 \wedge C_4$ by proving $\neg C_3 \vee \neg C_4 \Rightarrow \neg C_1 \vee \neg C_2$.

By Lemma 1, if K is not controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$, K is not delay controllable w.r.t. N_c, Σ_{uc} and $\mathcal{L}(G)$, i.e.,$\neg C_3 \Rightarrow \neg C_1$ which implies $\neg C_3 \Rightarrow \neg C_1 \vee \neg C_2$.

Assume $\neg C_4$ holds. By the definition of C_4, K is not delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$, i.e.,

$$(\exists \sigma \in \Sigma)(\exists s, s\sigma, s' \in K)s'\sigma \in \mathcal{L}(G) \setminus K$$
$$\wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset.$$

Since $\Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset$ and $N = N_o + N_c$, we have there exists $m, n \leq N_c$ such that $\Theta_D^{N_o}(s_{-m}) \cap \Theta_D^{N_o}(s'_{-n}) \neq \emptyset$. Since $s_{-m}, s'_{-n} \in K$, we have $q, q' \in Q_H$ such that $q = \delta_H(q_0, s_{-m})$ and $q' = \delta_H(q_0, s'_{-n})$. Moreover, since $q = \delta_H(q_0, s_{-m})$, $q' = \delta_H(q_0, s'_{-n})$ and $\Theta_D^{N_o}(s_{-m}) \cap \Theta_D^{N_o}(s'_{-n}) \neq \emptyset$, we have $(q, q') \in \mathrm{SP}_{IN}(K, N_o)$. Overall, there exists $(q, q') \in \mathrm{SP}_{IN}(K, N_o)$ such that $\mathrm{EN}(q, N_c) \cap \mathrm{DIS}(q', N_c) \neq \emptyset$. By Lemma 2, $K_{N_c}^{aug}$ is not delay observable w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G_{N_c}^{aug})$, which implies $\neg C_2$ is true. Hence, $\neg C_4 \Rightarrow \neg C_1 \vee \neg C_2$. Overall, we have $\neg C_3 \vee \neg C_4 \Rightarrow \neg C_1 \vee \neg C_2$.

□

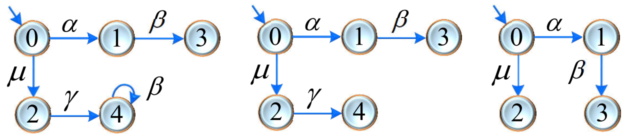
Corollary 1: Given system G with observation delays bounded by N_o event occurrences and control delays bounded by N_c event occurrences, there exists a networked supervisor π such that $\mathcal{L}_r(\pi/G) = \mathcal{L}_a(\pi/G) = K$ iff K is controllable w.r.t. Σ_{uc} and $\mathcal{L}(G)$, and delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$ with $N = N_c + N_o$.

Proof: Corollary 1 directly follows from Theorem 1 and Proposition 1. □

Since controllability is a special case of delay controllability and it is closed under union, the existence condition presented in Corollary 1 is much more simple and easier to handle than the existence condition presented in Theorem 1.

III. RELATIVE DELAY OBSERVABILITY

When a given specification language is not delay controllable and delay observable, one would like to calculate its supremal delay controllable and delay observable sublanguage that can be achieved deterministically under communication delays. However, this is too difficult since both delay controllability and delay observability are not closed under union [1]. Corollary 1 simplifies this problem because controllability is closed under union. To overcome the difficulty that delay



(a) Automaton G (b) Automaton A (c) Automaton H

FIGURE 2. Automata *G*, *A*, and *H*.

observability is not closed under union, we accommodate observation delays in relative observability [7] and define new language property called relative delay observability. The definition of relative observability in [7] is as follows.

Definition 1: Given the desired language K, ambient language C, and system language $\mathcal{L}(G)$ such that $K \subseteq C \subseteq \mathcal{L}(G)$, K is relatively observable w.r.t. C, P, and $\mathcal{L}(G)$, if

$$(\forall \sigma \in \Sigma)(\forall s\sigma \in K)$$
$$[((\forall s'\sigma \in \mathcal{L}(G))s' \in C)P(s) = P(s') \Rightarrow s'\sigma \in K].$$

As in [7], given a fixed C, relative observability is stronger than observability and closed under union. We refer the reader to [22] for the engineering application of relative observability. Relative observability is extended to relative delay observability as follows.

Definition 2: Given nonempty prefix-closed languages K, C, and $\mathcal{L}(G)$ such that $K \subseteq C \subseteq \mathcal{L}(G)$, we say K is relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$, where $N = N_o + N_c$, if

$$(\forall \sigma \in \Sigma)(\forall s \in \mathcal{L}(G))s\sigma \in K \Rightarrow [(\forall s' \in C)$$
$$s'\sigma \in \mathcal{L}(G) \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset \Rightarrow s'\sigma \in K]. \quad (2)$$

Note that when there are no observation and control delays, i.e., $N_c = N_o = 0$, delay relative observability is reduced to relative observability.

The following proposition investigates the relationship between relative delay observability and delay observability, relative network observability.

Proposition 2: Let $K \subseteq C \subseteq \mathcal{L}(G)$.

(1) If K is relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$, then K is delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$.

(2) If K is relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$, then K is relatively network observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$.

(3) However, the converse statements of (1) and (2) are not true in general.

Proof: We prove the first part of Proposition 2 by contradiction. Suppose K is relatively delay observsable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$ but is not delay observable w.r.t. Θ_D^N and $\mathcal{L}(G)$, i.e.,

$$(\exists \sigma \in \Sigma)(\exists s, s' \in K)s\sigma \in K \wedge s'\sigma \in \mathcal{L}(G)$$
$$\wedge s'\sigma \notin K \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset.$$

Since $K \subseteq C$, $s' \in K \wedge s'\sigma \in \mathcal{L}(G) \Rightarrow s' \in C \wedge s'\sigma \in \mathcal{L}(G)$. Hence,

$$(\exists \sigma \in \Sigma)(\exists s \in K)(\exists s' \in C)s\sigma \in K \wedge s'\sigma \in \mathcal{L}(G)$$
$$\wedge s'\sigma \notin K \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset,$$

which violates the definition of relative delay observability.

We prove the second part of Proposition 2 also by contradiction. Suppose K is relatively delay observsable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$ but is not relatively network observsable w.r.t. C, Θ_D^N and $\mathcal{L}(G)$, i.e.,

$$(\exists \sigma \in \Sigma)(\exists s, s\sigma \in K)(\forall t \in \Theta_D^N(s))$$
$$(\exists s' \in (\Theta_D^N)^{-1}(t))s' \in C \wedge s'\sigma \in \mathcal{L}(G) \setminus K. \quad (3)$$

Since $s' \in (\Theta_D^N)^{-1}(t)$, by the definition of $(\Theta_D^N)^{-1}$, $t \in \Theta_D^N(s')$. Moreover, since $t \in \Theta_D^N(s)$, we have $t \in \Theta_D^N(s) \cap \Theta_D^N(s')$, which implies $\Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset$. Hence,

$$(\exists \sigma \in \Sigma)(\exists s, s\sigma \in K)(\exists s' \in C)$$
$$s'\sigma \in \mathcal{L}(G) \setminus K \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset,$$

which violates the definition of relative delay observability.

To show the third part of of Proposition 2 is true, consider automata H, A and G that are depicted in Fig. 2 that generate K, C and $\mathcal{L}(G)$, respectively. We can see $\Sigma = \{\alpha, \beta, \gamma, \mu\}$. Assume $\Sigma_o = \{\alpha, \gamma\}$, $\Sigma_c = \Sigma$, and the upper bounds on observation delays and control delays are 1, i.e., $N_o = N_c = 1$.

Note that, for H and G depicted in Fig.2, only an occurrence of γ can cause the system leaves K, and all sequences in K do not contain γ. Consequently, for all $s\sigma \in K$ with some $\sigma \in \Sigma$, there does not exist $s' \in K$ with $s'\sigma \in \mathcal{L}(G) \setminus K$. By (1), K is delay observable regardless of Θ_D^N.

K is not relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$. To see this, take $s = \alpha$ and $s' = \mu\gamma$. By the definition of Θ_D^N, $\Theta_D^N(s) = \{\varepsilon, \alpha\}$ and $\Theta_D^N(s') = \{\varepsilon, \gamma\}$. Then, $\Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset$, $s \in K$, $s\beta \in K$, $s' \in C$, $s'\beta \in \mathcal{L}(G)$, and $s'\beta \notin K$, which violates the definition of relative delay observability.

K is relatively network observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$. By Fig.2(c), $K = \{\varepsilon, \mu, \alpha, \alpha\beta\}$. To show K is relatively network observable, we need to show (3) is not true for $s\sigma = \mu$, $s\sigma = \alpha$, and $s\sigma = \alpha\beta$. Since for all $s' \in C$ and $\sigma \in \Sigma$ with $s'\sigma \in \mathcal{L}(G) \setminus K$, by Fig.2, we have $\sigma = \beta$ or $\sigma = \gamma$. Hence, (3) is not true for $s\sigma = \mu$ and $s\sigma = \alpha$ regardless of Θ_D^N. Next, we consider $s\sigma = \alpha\beta$ with $s = \alpha$ and $\sigma = \beta$. By the definition of Θ_D^N, $t = \alpha \in \Theta_D^N(s)$. By Fig.2, it is not hard to find that all sequences containing α are in K. That is to say, for all $s' \in (\Theta_D^N)^{-1}(t)$ and $s'\sigma \in \mathcal{L}(G)$, we have $s'\sigma \in K$. Therefore, (3) is not true for $s\sigma = \alpha\beta$.

By K is delay observable and not relatively delay observable, we have the converse statement of (1) is not true. By K is relatively network observable and not relatively delay observable, we have the converse statement of (2) is not true. □

By Proposition 2, relative delay observability implies delay observability and relative network observability. Proposition 3 states that relative delay observability is closed under union.

Proposition 3: Let M be a set of languages K with $K \subseteq C \subseteq \mathcal{L}(G)$ that are relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$, then so is $\bigcup M$.

Proof: If $M = \emptyset$, the proposition is trivially true for $\bigcup M$.

The case of $M \neq \emptyset$ is proven as follows. Given arbitrary $\sigma \in \Sigma$ and arbitrary $s\sigma \in \bigcup M$, there exists $K \in M$ such that $s\sigma \in K$. Since K is relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$, for all $s' \in C$

$$s'\sigma \in \mathcal{L}(G) \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset \Rightarrow s'\sigma \in K.$$

Since $K \subseteq \bigcup M$, we have $s'\sigma \in \bigcup M$. Since $\sigma \in \Sigma$ and $s\sigma \in \bigcup M$ are arbitrary given, $\bigcup M$ is relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$. □

Note that Proposition 3 does not require K to be closed. As stated in [23], for a non-closed language $K \subseteq \mathcal{L}_m(G)$, there always exists a nonblocking networked supervisor π such that $\mathcal{L}_r(G,\pi) = \mathcal{L}_a(G,\pi) = K$ iff (1) K is delay controllable w.r.t. Σ_{uc}, N_c, and $\mathcal{L}(G)$; (2) $K_{N_c}^{aug}$ is delay observable w.r.t. $\Theta_D^{N_o}$ and $\mathcal{L}(G_{N_c}^{aug})$; (3) K is $\mathcal{L}_m(G)$-closed. Consequently, when blocking is considered, by Proposition 1, 2, and 3, there always exists a supremal controllable, relatively delay observable, and $\mathcal{L}_m(G)$-closed sublanguage of K that can be achieved deterministically by nonblocking networked supervisory control.

IV. SUPREMAL CONTROLLABLE AND RELATIVELY DELAY OBSERVABLE SUBLANGUAGE

In this section, we first show how to check relative delay observability. We then show how to synthesize the supremal controllable and relatively delay observable sublanguage.

A. CHECKING RELATIVE DELAY OBSERVABILITY

Let

$$\mathcal{T}_{conf}(P) = \{(x,y) \in Q \times Q : (\exists s,t \in \mathcal{L}(G))x = \delta(q_0,s) \wedge y = \delta(q_0,t) \wedge P(s) = P(t)\}, \quad (4)$$

be the set of confusable pairs in G under the partial observation P. An algorithm for calculating $\mathcal{T}_{conf}(P)$ with polynomial time complexity w.r.t. the sizes of state space and the event set in G was proposed in [24].

Denote the set of all possible confusable state pairs in G under the delayed observation mapping Θ_D^N as:

$$\mathcal{T}_{conf}(\Theta_D^N) = \{(x,y) \in Q \times Q : (\exists s,t \in \mathcal{L}(G))x = \delta(q_0,s) \wedge y = \delta(q_0,t) \wedge \Theta_D^N(s) \cap \Theta_D^N(t) \neq \emptyset\}. \quad (5)$$

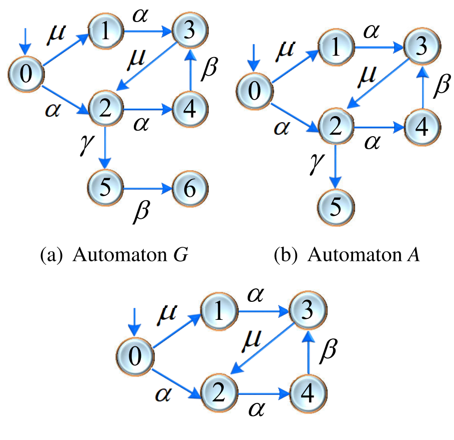
For all $q \in Q$, let $Reach(q,N) = \{q' \in Q : (\exists s \in \Sigma^{\leq N})q' = \delta(q,s)\}$ be the set of states in G that can be reached from state q via a sequence s with $|s| \leq N$. By the defintions of $\mathcal{T}_{conf}(P)$ and $\mathcal{T}_{conf}(\Theta_D^N)$, we have the following proposition.

Proposition 4: $\mathcal{T}_{conf}(\Theta_D^N)$ can be calculated from $\mathcal{T}_{conf}(P)$ as:

$$\mathcal{T}_{conf}(\Theta_D^N) = \{(x,y) \in Q \times Q : (\exists (q,q') \in \mathcal{T}_{conf}(P)) \; x = Reach(q,N) \wedge y = Reach(q',N)\}.$$

Proof:

Let $T = \{(x,y) \in Q \times Q : (\exists (q,q') \in \mathcal{T}_{conf}(P))x = Reach(q,N) \wedge y = Reach(q',N)\}$.



(a) Automaton G (b) Automaton A

(c) Automaton H

FIGURE 3. Automata *G*, *A*, and *H*.

We first prove $T \subseteq \mathcal{T}_{conf}(\Theta_D^N)$. For an arbitrary state pair $(x,y) \in T$, by definition, there exist $(q,q') \in \mathcal{T}_{conf}(P)$ and $t,t' \in \Sigma^{\leq N}$ with $x = \delta(q,t)$ and $y = \delta(q',t')$. Since $(q,q') \in \mathcal{T}_{conf}(P)$, by (4), there exists $s,s' \in \mathcal{L}(G)$ such that $q = \delta(q_0,s) \wedge q' = \delta(q_0,s') \wedge P(s) = P(s')$. Since $q = \delta(q_0,s)$, $q' = \delta(q_0,s')$, $x = \delta(q,t)$, and $y = \delta(q',t')$, we have $x = \delta(q_0,st)$ and $y = \delta(q_0,s't')$. Moreover, since $t,t' \in \Sigma^{\leq N}$ and $P(s) = P(s')$, by the definition of $\Theta_D^N(\cdot)$, $P(s) \in \Theta_D^N(st) \cap \Theta_D^N(s't') \neq \emptyset$. Therefore, there exists $st, s't' \in \mathcal{L}(G)$ such that $x = \delta(q_0,st) \wedge y = \delta(q_0,s't') \wedge \Theta_D^N(st) \cap \Theta_D^N(s't') \neq \emptyset$, which implies $(x,y) \in \mathcal{T}_{conf}(\Theta_D^N)$. Since $(x,y) \in T$ is arbitrarily given, $T \subseteq \mathcal{T}_{conf}(\Theta_D^N)$.

Next, we prove $\mathcal{T}_{conf}(\Theta_D^N) \subseteq T$. For an arbitrary state pair $(x,y) \in \mathcal{T}_{conf}(\Theta_D^N)$, by (5), there exists $s,s' \in \mathcal{L}(G)$ such that $x = \delta(q_0,s) \wedge y = \delta(q_0,s') \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset$. Since $\Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset$, by the definition of $\Theta_D^N(\cdot)$, there exists $v,v' \leq \Sigma^{\leq N}$ such that $s = uv \wedge s' = u'v' \wedge P(u) = P(u')$. Suppose $\delta(q_0,u) = q$ and $\delta(q_0,u') = q'$. Since $\delta(q_0,s) = x$, $\delta(q_0,s') = y$, $s = uv$, and $s' = u'v'$, we have $\delta(q,v) = x$ and $\delta(q',v') = y$. Since $P(u) = P(u')$, $\delta(q_0,u) = q$, and $\delta(q_0,u') = q'$, by (4), $(q,q') \in \mathcal{T}_{conf}(P)$. Therefore, there exists $(q,q') \in \mathcal{T}_{conf}(P)$ such that $\delta(q,v) = x$ and $\delta(q',v') = y$ with $v,v' \leq \Sigma^{\leq N}$. By the definition of T, $(x,y) \in T$. Since $(x,y) \in \mathcal{T}_{conf}(\Theta_D^N)$ is arbitrarily given, $\mathcal{T}_{conf}(\Theta_D^N) \subseteq T$. □

Example 1: Let us consider the automaton G depicted in Fig.3(a). Assume $\Sigma_o = \{\alpha,\gamma,\mu\}$ and the upper bounds on observation delays and control delays are 1, i.e., $N_o = N_c = 1$. By (4),

$$\mathcal{T}_{conf}(P) = \{(0,0),(1,1),(2,2),(3,3),(4,4), 5,5),(6,6),(3,4),(5,6)\}.$$

Let $N = N_o + N_c = 2$. By the definition of $Reach(\cdot)$, we have $Reach(0,N) = \{0,1,2,3,4,5\}$, $Reach(1,N) = \{1,2,3\}$, $Reach(2,N) = \{2,3,4,5,6\}$, $Reach(3,N) = \{2,3,4,5\}$, $Reach(4,N) = \{2,3,4\}$, $Reach(5,N) = \{5,6\}$, and $Reach(6,N) = \{6\}$. By (5),

$$\mathcal{T}_{conf}(\Theta_D^N) = \mathcal{T}_{conf}(P) \cup \{(0,1),(0,2),(0,3),$$

$$(0,4),(0,5),(1,2),(1,3),(2,3),$$
$$(2,4),(2,5),(2,6),(3,5),(5,4)\}.$$

Note that, we only list (q,q') *in* $\mathcal{T}_{conf}(P)$ *and* $\mathcal{T}_{conf}(\Theta_D^N)$, *and omit* (q',q) *for brevity.*

Given system G, A, and H with $H \sqsubseteq A \sqsubseteq G$, let

$$\mathscr{T}_{spec}(H,A,G) = \{(q,q') \in Q_H \times Q_A : (\exists \sigma \in \Sigma)\sigma \in \Gamma(q) \wedge \sigma \in \Gamma(q') \wedge \sigma \in \Gamma_H(q) \wedge [q' \in Q_H \Rightarrow \sigma \notin \Gamma_H(q')]\}, \quad (6)$$

be the set of state pairs $(q,q') \in Q_H \times Q_A$ that there exists a $\sigma \in \Sigma$ defined at q and q' in G, but only defined at q and not defined at q' in H if $q' \in Q_H$.

Proposition 5: Given G, A, *and* H *with* $H \sqsubseteq A \sqsubseteq G$ *that generate* $\mathcal{L}(G)$, C, *and* K, *respectively,* K *is relatively delay observable w.r.t.* C, Θ_D^N, *and* $\mathcal{L}(G)$ *if and only if* $\mathscr{T}_{spec}(H,A,G) \cap \mathscr{T}_{conf}(\Theta_D^N) = \emptyset$.

Proof: ($\Leftarrow$) Suppose that K is relatively delay observable w.r.t. C, Θ_D^N, and $\mathscr{L}(G)$, i.e.,

$$(\exists \sigma \in \Sigma)(\exists s \in K)(\exists s' \in C) s\sigma \in K \wedge s'\sigma \in \mathcal{L}(G) \setminus K \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset.$$

Suppose $\delta(q_0,s) = x$ and $\delta(q_0,s') = y$. Since $\Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset$, by (5), $(x,y) \in \mathscr{T}_{conf}(\Theta_D^N)$. Since $s\sigma \in K \wedge s'\sigma \in \mathscr{L}(G)$, we have $\sigma \in \Gamma_H(x) \wedge \sigma \in \Gamma(x) \wedge \sigma \in \Gamma(y)$. If $y \in Q_H$, i.e., $s' \in K$, by $s'\sigma \notin K$, we have $\sigma \notin \Gamma_H(x)$. By (6), $(x,y) \in \mathscr{T}_{spec}(H,A,G)$. If $y \notin Q_H$, since $\sigma \in \Gamma(x) \wedge \sigma \in \Gamma_H(x) \wedge \sigma \in \Gamma(y)$ and $y \notin Q_H$, by (6), $(x,y) \in \mathscr{T}_{spec}(H,A,G)$. Therefore, there exists $(x,y) \in \mathscr{T}_{spec}(H,A,G) \cap \mathscr{T}_{conf}(\Theta_D^N)$, which violates $\mathscr{T}_{spec}(H,A,G) \cap \mathscr{T}_{conf}(\Theta_D^N) = \emptyset$.

($\Rightarrow$) Suppose $\mathscr{T}_{spec}(H,A,G) \cap \mathscr{T}_{conf}(\Theta_D^N) \neq \emptyset$, by definitions, there exist $s \in K$ and $s' \in C$ such that $\Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset$ and $(\delta(q_0,s),\delta(q_0,s')) \in \mathcal{T}_{spec}(H,A,G)$. Write $\delta(q_0,s) = x$ and $\delta(q_0,s') = y$, then, $x \in Q_H \wedge y \in Q_A$. Since $H \sqsubseteq A$ and $y \in Q_A$, y must satisfy one of the following two cases:

(1) $y \in Q_H$. In this case, by (6), there exists $\sigma \in \Sigma$ with $\sigma \in \Gamma(x)$, $\sigma \in \Gamma(y)$, $\sigma \in \Gamma_H(x)$, and $\sigma \notin \Gamma_H(y)$, i.e., $s\sigma \in K \subseteq \mathcal{L}(G)$, $s'\sigma \in \mathcal{L}(G)$, and $s'\sigma \notin K$.

(2) $y \in Q_A \backslash Q_H$. In this case, also by (6), there exists $\sigma \in \Sigma$ with $\sigma \in \Gamma(x)$, $\sigma \in \Gamma(y)$, and $\sigma \in \Gamma_H(x)$, i.e., $s\sigma \in K \subseteq \mathcal{L}(G)$ and $s'\sigma \in \mathcal{L}(G)$. Moreover, since $y \in Q_A \setminus Q_H$ and $\delta(q_0,s') = y$, $s' \in C \setminus K$, which implies $s'\sigma \notin K$.

Therefore, we have

$$(\exists \sigma \in \Sigma)(\exists s \in K)(\exists s' \in C) s\sigma \in K \; s'\sigma \in \mathcal{L}(G) \setminus K \wedge \Theta_D^N(s) \cap \Theta_D^N(s') \neq \emptyset,$$

which violates K is relatively delay observable w.r.t. C, Θ_D^N, and $\mathscr{L}(G)$. □

Theorem 2: Given G, A, *and* H *with* $H \sqsubseteq A \sqsubseteq G$, *the worst-case time complexity of checking relative delay observability is* $\mathcal{O}(|Q|^2 \times |\Sigma|)$.

Proof: By Proposition 4, the complexity of calculating $\mathcal{T}_{conf}(\Theta_D^N)$ is determined by the complexity of calculating $\mathcal{T}_{conf}(P)$, which is of the order $|Q|^2 \times |\Sigma|$ in the worst case [24]. To calculate $\mathcal{T}_{spec}(H,A,G)$, for all $(q,q') \in Q_H \times Q_A$ and $\sigma \in \Sigma_c$, we need check if $\sigma \in \Gamma(q) \wedge \sigma \in \Gamma(q') \wedge \sigma \in \Gamma_H(q) \wedge [q' \in Q_H \Rightarrow \sigma \notin \Gamma_H(q')]$ is true. Correspondingly, the worst-case complexity for calculating $\mathcal{T}_{spec}(H,A,G)$ is of the order $|Q|^2 \times |\Sigma|$. Therefore, the worst-case time complexity of checking relative delay observability is $\mathcal{O}(|Q|^2 \times |\Sigma|)$. □

Let us use the following example to further illustrate all the above results.

Example 2: Continue with Example 1. Consider the automata G, A, *and* H *depicted in Fig.3 that generate* $\mathcal{L}(G)$, C, *and* K, *respectively. Clearly, automaton* G *has the event set* $\Sigma = \{\alpha,\beta,\gamma,\mu\}$. *Since* $\beta \in \Gamma(5) \wedge 5 \notin Q_H \wedge \beta \in \Gamma(4) \wedge \beta \in \Gamma_H(4)$, *by (6),* $(5,4) \in \mathscr{T}_{spec}(H,A,G)$. *On the other hand, as described in Example 1,* $(5,4) \in \mathcal{T}_{conf}(\Theta_D^N)$, *which implies* $(5,4) \in \mathcal{T}_{spec}(H,A,G) \cap \mathcal{T}_{conf}(\Theta_D^N) \neq \emptyset$. *By Proposition 5,* $\mathcal{L}(H)$ *is not relatively delay observable w.r.t.* C, Θ_D^N, *and* $\mathcal{L}(G)$.

B. SUPREMAL CONTROLLABLE AND RELATIVELY DELAY OBSERVABLE SUBLANGUAGE

When the specification language does not satisfy some language properties, in many applications, it may be useful or preferable to calculate a supremal element that satisfies all these properties in a solution space of interest, which can be finite and countable. In this subsection, we introduce a general method to calculate a supremal sublanguage that satisfies both controllability and another user-defined language property in a finite solution space. As a special case, we calculate the supremal sublanguage that is controllable and relatively delay observable.

We consider in this paper the finite solution space Ξ for calculating the supremal sublanguage of K satifies: 1) Ξ is closed under union, i.e., for all $K_1, K_2 \in \Xi$, $K_1 \cup K_2 \in \Xi$; 2) $\cup\Xi = K$; 3) Ξ only contains regular languages. Note that these assumptions are actually not a restriction because we can always make Ξ closed by adding elements that are the union of elements in Ξ.

Let Ψ be the set of automata whose elements individually represent languages in Ξ. We assume the automata that represent languages in Ξ satisfies 1) $H' \in \Psi$ with $H' \sqsubseteq A \sqsubseteq G$, and 2) for all $H', H'' \in \Psi$, $\mathcal{L}(H') \subseteq \mathcal{L}(H'') \Rightarrow H' \sqsubseteq H''$. Since Ξ is finite and only contains regular languages, we can always change automaton presentations of languages in Ξ to satisfy these requirements.

Let ϖ, L, and $M = \{G_1, G_2, \ldots, G_k\}$ be a language property, a regular language, and a finite list of automata, respectively. Note that all the elements in a list are ordered. We say L is ϖ-preserved w.r.t. M, denoted by $L \models_M \varpi$, if language L satisfies the property ϖ w.r.t. M. For example, let ϖ be the property of relative delay observability and $M = \{A, G\}$, then $K \models_M \varpi$ means K is relatively delay observable w.r.t. $\mathcal{L}(A)$, Θ_D^N, and $\mathcal{L}(G)$.

Given system G, language property ϖ, list of automata M, and Ψ, we want to calculate $H^\uparrow \in \Psi$ such that $\mathcal{L}(H^\uparrow)$ is controllable w.r.t. G and $\mathcal{L}(H^{\uparrow}) \models_M \varpi$ and there does not exist $H' \in \Psi$ with $\mathcal{L}(H')$ is controllable w.r.t. G, $\mathcal{L}(H') \models_M \varpi$, and $H' \sqsubset H^{\uparrow}$. Formally, we call $\mathcal{L}(H^{\uparrow})$ the supremal controllable and ϖ-preserved sublanguage of K w.r.t. M and Ξ. The following proposition says if the property of ϖ is closed under union, a unique $H^{\uparrow}$ exists.

Proposition 6: Given a language property ϖ that is closed under union, there exists a $H' \in \Psi$ such that $\mathcal{L}(H')$ is controllable w.r.t. G and $\mathcal{L}(H') \models_M \varpi$ iff there exists a unique $H^{\uparrow} \in \Psi$ mentioned above.

Proof: ($\Rightarrow$) Since $\mathcal{L}(H^{\uparrow})$ is controllable, $\mathcal{L}(H^{\uparrow}) \models_M \varpi$ and $H^{\uparrow} \in \Psi$, the sufficiency holds.

($\Leftarrow$) Suppose there exists $H' \in \Psi$ such that $\mathcal{L}(H')$ is controllable and $\mathcal{L}(H') \models_M \varpi$. Let $\mathcal{L}(H^{\uparrow}) = \{\cup \mathcal{L}(H') : H' \in \Psi \wedge \mathcal{L}(H') \models_M \varpi \wedge \mathcal{L}(H')$ is controllable w.r.t. $G\}$ Then, $\mathcal{L}(H^{\uparrow})$ is unique and $H'' \sqsubseteq H^{\uparrow}$ for all $H'' \in \Psi$ with $\mathcal{L}(H'')$ is controllable w.r.t. G and $\mathcal{L}(H'') \models_M \varpi$. Moreover, since both controllability and property of ϖ are closed under union, $\mathcal{L}(H^{\uparrow})$ is controllable w.r.t. G and $\mathcal{L}(H^{\uparrow}) \models_M \varpi$. Since Ξ is closed under union and $\mathcal{L}(H') \in \Xi$, $\mathcal{L}(H^{\uparrow}) \in \Xi$ and $H^{\uparrow} \in \Psi$.

Overall, there exists a unique $H^{\uparrow} \in \Psi$ such that $\mathcal{L}(H^{\uparrow})$ is controllable w.r.t. G and $\mathcal{L}(H^{\uparrow}) \models_M \varpi$, and $H'' \sqsubset H^{\uparrow}$ for all $H'' \in \Psi$ with $\mathcal{L}(H'')$ is controllable w.r.t. G and $\mathcal{L}(H'') \models_M \varpi$. Therefore, the necessity holds. □

Given H, A, and G with $H \sqsubseteq A \sqsubseteq G$ that generate K, C, and $\mathcal{L}(G)$, respectively, a closed language property ϖ, and a set Ψ, we generalize the algorithm introduced in our previous work [22] for calculating the supremal controllable and relatively network obsetvable suablanguage to calculate the supremal controllable and ϖ-preserved sublanguage in Algorithm 1. ϖ can be but is not limited to, normality [3], weakly normality [5], relative observability [7], strong observability [6], relative network observability [22], and relative delay observability. Clearly, the problem solved in our previous work [22] is one of the problems solved by the algorithm introduced in this paper. In other words, the algorithm introduced in this paper is more general.

Given G and H with $H \sqsubseteq G$, if H is not controllable w.r.t. G, an algorithm is introduced in [25] to calculate $H^{\uparrow C} \sqsubseteq H \sqsubseteq G$ that generates the supremal controllable sublanguage of $\mathcal{L}(H)$. We call $H^{\uparrow C}$ the supremal controllable sub-automaton of H in this paper. Extending this algorithm, Algorithm 1 calculates $H^{\uparrow}$.

We outline the mechanism of Algorithm 1 as follows.

At first, by Step 2, $\mathcal{A} = \{H^{\uparrow C}\}$. If $\mathcal{L}(H^{\uparrow C}) \models_M \varpi$ is true, by Step 3, $H^* \leftarrow H$. Since there exists no $H' \in \Psi$ that is controllable and $H^{\uparrow C} \sqsubset H'$, we have $H^* = H^{\uparrow}$. If $\mathcal{L}(H) \models_M \varpi$ is false, the solution is some element $H' \in \Psi$ with $H' \sqsubseteq H^{\uparrow C}$ if exists. Hence, at the end of the first iteration of Algorithm 1, if the solution exists, it is either 1) a supremal controllable sub-automaton of an element in $\mathcal{R}$, or 2) a sub-automaton of one of these supremal controllable automata. All the mentioned automata will be considered one by one according to the decreasing numbers of transitions for future iterations until returning the solution or declaring no solution

---

Algorithm 1: Supremal sublanguage

Input: System G, a list of automata M, a user-defined language property ϖ, and Ψ;

Output: Automaton H^* if $\exists H' \in \Psi$: $\mathcal{L}(H')$ is controllable w.r.t. G and $\mathcal{L}(H') \models_M \varpi$, and declaring no solution exits, otherwise;

Step 1: Set $\mathcal{A}$ to be the empty array of automata, $\mathcal{R} \leftarrow \{H\}$, and $i = 1$;

Step 2: For each $H' \in \mathcal{R}$, use techniques in [25] to calculate $H'^{\uparrow C}$ with the inputs H' and G, and insert $H'^{\uparrow C}$ into $\mathcal{A}$ according to decreasing numbers of transitions if $H'^{\uparrow C} \notin \mathcal{A}$;

Step 3: If $i > |\mathcal{A}|$, declaring no solution exists, and otherwise, pick up the i-th element in $\mathcal{A}$, says $\widetilde{H}$, and check if $\mathcal{L}(\widetilde{H}) \models_M \varpi$ is satisfied. If the check is true and $\widetilde{H} \in \Psi$, return $H^* \leftarrow \widetilde{H}$, and otherwise, set $\mathcal{R} \leftarrow \{H' \in \Psi | H' \sqsubset \widetilde{H} \wedge [(\nexists H'' \in \Psi) H'' \sqsubset \widetilde{H} \wedge H' \sqsubset H'']\}$;

Step 4: Set $i = i + 1$;

Step 5: Iterate Step 2, 3, and 4 until returning automaton H^* or declaring no solution exists.

---

exists. On the other hand, if $\mathcal{L}(H) \models_M \varpi$ is not true and there exists no $H' \in \Psi$ with $H' \sqsubset H^{\uparrow C}$, we have $\mathcal{R} = \emptyset$ at Step 3 in the first iteration. The algorithm will declare no solution exists in the second iteration.

Theorem 3 says Algorithm 1 calculates $H^{\uparrow}$.

Theorem 3: Given system G, a list of automata M, a closed language property ϖ, and Ψ, if there exists $H' \in \Psi$ such that $\mathcal{L}(H')$ is controllable w.r.t. G and $\mathcal{L}(H') \models_M \varpi$, Algorithm 1 returns $H^{\uparrow} = H^*$; it declares no solution, otherwise.

Proof: The proof is by induction.

In the first iteration, by Step 1, $\mathcal{R} = \{H\}$. Step 2 calculates $H^{\uparrow C}$ that generates the supremal controllable sublanguage of $\mathcal{L}(H)$. We then have $\mathcal{A} = \{H^{\uparrow C}\}$. Since $i = 1$ and $|\mathcal{A}| = 1$, by Step 3, if $\mathcal{L}(H^{\uparrow C}) \models_M \varpi$ is true, $H^* \leftarrow H^{\uparrow C}$. Since there exists no $H' \in \Psi$ with $H^{\uparrow C} \sqsubset H'$ and $\mathcal{L}(H')$ is controllable w.r.t. G and $\mathcal{L}(H') \models_M \varpi$, by Proposition 6, $H^{\uparrow} = H^*$. On the other hand, if $\mathcal{L}(H^{\uparrow C}) \models_M \varpi$ is not true, since $H^{\uparrow C}$ and all the automata $H' \in \Psi$ with $H^{\uparrow C} \sqsubset H'$ are not the solution, the solution is some automaton H' in Ψ with $H' \sqsubset H^{\uparrow C}$ if exists. Hence, at the end of the first iteration, if the solution $H^{\uparrow}$ exists, it is either 1) a supremal controllable sub-automaton of an element in $\mathcal{R}$, or 2) a sub-automaton of one of these supremal controllable automata. If $\mathcal{R} = \emptyset$, we can conclude no solution exists. In the second iteration, since $\mathcal{R} = \emptyset$, by Step 2, $\mathcal{A} = \{H^{\uparrow C}\}$. Since $i = 2 > |\mathcal{A}|$, the algorithm declares no solution exists.

Suppose the algorithm does not terminate by the end of the kth iteration. Also suppose that if $H^{\uparrow}$ exists, it is one of the following three cases: 1) a supremal controllable sub-automaton of an element in $\mathcal{R}$; 2) the lth element in $\mathcal{A}$ with $l \geq k$; 3) an sub-automaton of one of automata mentioned in 1) and 2).

We now consider the $k+1$st iteration as follows. After the kth iteration, all the supremal controllable sub-automata of elements in $\mathcal{R}$ are listed after the kth element in $\mathcal{A}$. w.l.o.g., we denoted the nth element of $\mathcal{A}$ by H_n. Then, if the solution exists, it is either an element listed after H_k or a sub-automaton of such an element then. Since $i = k + 1$ at the beginning of $k + 1$st iteration, if $\mathcal{A}$ only has k elements, by Step 3, the algorithm declares no solution exists. Otherwise, if $\mathcal{A}$ has more than k elements, since elements in $\mathcal{A}$ are arranged according to decreasing numbers of transitions, no elements listed after H_{k+1} is a sub-automaton of H_{k+1}. Then, if $\mathcal{L}(H_{k+1}) \models_M \varpi$ is true, we conclude that H_{k+1} is the solution, which is returned by Step 3. If $\mathcal{L}(H_{k+1}) \models_M \varpi$ is not true, we conclude that H_{k+1} is not the solution. However, by Step 3, we include its subautomata for future iterations. Therefore, the induction hypothesis still holds by the end of the $k+1$st iterations.

The iteration number of the algorithm is upper bounded by the numbers of all the sub-automata of H, which is upper bounded by $2^{|Q|\times|\Sigma|}$. Therefore, the algorithm will terminate within finite steps. $\square$

The following corollary states how to calculate the supremal controllable and relatively delay observable sublanguage.

Corollary 2: Given G, A, and H with $H \sqsubseteq A \sqsubseteq G$ that generate $\mathcal{L}(G)$, C, and K, respectively, and Ψ. Let $M = \{A, G\}$ and ϖ be the class of properties of controllability and relative delay observability. If there exists $H' \in \Psi$ such that $\mathcal{L}(H')$ is controllable and relatively delay observable, Algorithm 1 computes $H^{\uparrow} = H^*$ that is unique; it declares no solution, otherwise.

Proof: By Theorem 3, if $H' \in \Psi$ is controllable and relatively delay observable, Algorithm 1 computes $H^{\uparrow} = H^*$, and otherwise, declaring no soulution exists. Since both controllability and relative delay observability are closed under union, by Proposition 6, $H^{\uparrow}$ is unique. $\square$

Theorem 4: The worst-case time complexity of synthesizing the supremal controllable and relatively delay observable sublanguage using Algorithm 1 is $2^{\mathcal{O}(|Q|\times|\Sigma|)}$.

Proof: By Theorem 2, the worst-case time complexity of checking relative delay observability is of the order $|Q|^2 \times |\Sigma|$. The complexity of calculating $H^{\uparrow C}$ with $H \sqsubseteq G$ is of the order $|Q| \times |\Sigma|$ in the worst case [21]. Correspondingly, the worst-case complexity of Algorithm 1 for calculating the supremal controllable and relatively delay observable sublanguage is of the order $|\mathcal{A}| \times |Q| \times |\Sigma| + |\mathcal{A}| \times |Q|^2 \times |\Sigma|$. Since every $H' \in \mathcal{A}$ has $H' \sqsubseteq H \sqsubseteq G$, $|\mathcal{A}| \leq 2^{|Q|\times|\Sigma|}$. Therefore, the worst-case time complexity of Algorithm 1 for calculating the supremal controllable and relatively delay observable sublanguage is $2^{\mathcal{O}(|Q|\times|\Sigma|)}$. $\square$

V. EXAMPLE

In this section, we use a practical example to show the application of results derived in this paper.

Let us consider a simple signalized intersection as shown in Fig.4. When a vehicle arrives at the intersection, it needs to communicate with the intersection to observe the color of the traffic light and make decisions accordingly. The communication is over some shared network. We assume the observation delays are upper bounded by 1, i.e., $N_o = 1$, and all the control commands can be executed immediately when they are issued, i.e., $N_c = 0$.

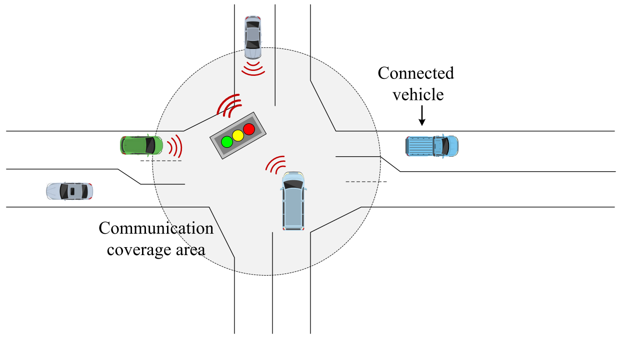
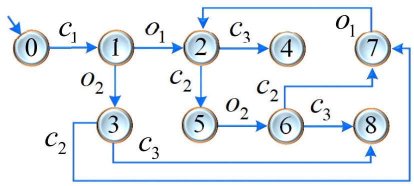


FIGURE 4. A signalized intersection.

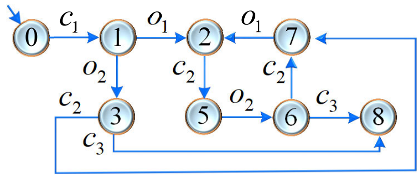
The transport safety model $G = (Q, \Sigma, \delta, \Gamma, q_0)$ is shown in Fig.5(a). The event set $\Sigma = \{c_1, c_2, c_3, o_1, o_2\}$, where c_1 means vehicle x approaches a signalized intersection; c_2 means vehicle x stops; c_3 means vehicle x passes the intersection; o_1 means the traffic light at the intersection approached is red; o_2 means the traffic light at the intersection approached is green. Define the set of observable events as $\Sigma_o = \{c_1, c_2, o_1, o_2\}$ and the set of controllable events as $\Sigma_c = \{c_1, c_2, c_3\}$.

We interpret the construction of G as follows. If vehicle x arrives at the intersection, upon the occurrence of c_1, the system makes a state transition from 0 to 1. Then, vehicle x needs to observe the signal and make decisions. If the signal in the forward direction of vehicle x is switched to red (green), upon the occurrence of o_1 (o_2), the system will make a state transition from 1 to 2 (3). If the system is uncontrolled, vehicle x can stop or pass the intersection. Hence, event c_2 and c_3 are defined at both 2 and 3. If the system now is in state 2 and vehicle x passes the intersection, upon the occurrence of c_3, the system will move to 4. On the other hand, If vehicle x stops, upon the occurrence of c_2, the system will move from 2 to 5. Since the vehicle has decided to stop at the intersection, when the system is in state 5, vehicle x needs to wait until the signal is switched from red to green. And after o_2 occurs at 5, the system will make a state transition from 5 to 6. Then, vehicle x can pass the intersection or continue to stop at the intersection. Upon the occurrences of c_3 and c_2 at 6, the system will move to 8 and 7, respectively. Meanwhile, if c_3 occurs at 3, vehicle x passes the intersection, and if c_2 occurs at 3, vehicle x stops at the intersection. Upon the occurrences of c_3 and c_2 at 3, the system will move to 8 and 7, respectively. When the system is in 7, vehicle x needs to wait until the signal is switched from green to red. After o_1 occurs at 7, the system will move to 2 and make state transitions as mentioned above.

By traffic laws, passing the intersection (enabling c_3) is not permitted if the traffic light at the intersection approached

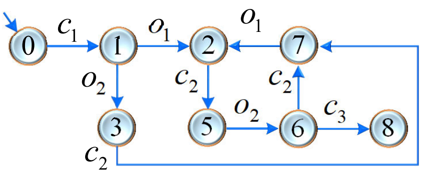


(a) Uncontrolled system G

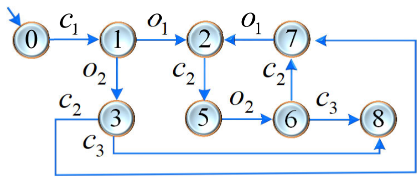


(b) Desired system H

FIGURE 5. Automata *G* and *H*.



(a) Automaton H_1



(b) Automaton H_2

FIGURE 6. Automata H_1 and H_2.

is red. On the other hand, if the traffic light at the intersection approached is green, vehicle x can pass the intersection (enabling c_3). Hence, as shown in Fig.5(b), the desired system $H = (Q_H, \Sigma, \delta_H, \Gamma_H, q_0)$ is obtained from G by deleting state 4 and the transition $(2, c_3, 4)$.

Let $s = c_1o_1$ and $s' = c_1o_2$, by the definition of Θ_D^N, $\Theta_D^N(s) = \{c_1, c_1o_1\}$ and $\Theta_D^N(s') = \{c_1, c_1o_2\}$. That is to say, when the networked supervisor sees c_1, the system could be in state 2 or 3 due to the observation delay of o_1 or o_2. A control conflict occurs since we need to enable c_3 in state 3 but disable c_3 in state 2. By the definition of delay observability, $\mathcal{L}(H)$ is not delay observable w.r.t. Θ_D^N, and $\mathcal{L}(G)$ with $N = N_c + N_o = 1$. Next, we apply Algorithm 1 to calculate the supremal controllable and relatively delay observable sublanguage of $\mathcal{L}(H)$.

First, given a solution space $\Xi = \{L_1, L_2\}$ with $L_1 = \mathcal{L}(H) \setminus \{c_1o_2c_3\}$ and $L_2 = \mathcal{L}(H)$. Then, as shown in Fig.6, $\Psi = \{H_1, H_2\}$ with $H_1 \sqsubseteq H_2$ and $L_i = \mathcal{L}(H_i)$, $i = 1, 2$. We denote by $H_i = (Q_{H_i}, \Sigma, \delta_{H_i}, \Gamma_{H_i}, q_0)$, $i = 1, 2$. Let $A = H$, i.e., the ambient language $C = \mathcal{L}(A) = \mathcal{L}(H)$.

Consider the first iteration. By Step 1, $\mathcal{R} = \{H\}$ and $i = 1$. Since H is controllable, by Step 2, $\mathcal{A} = \{H\}$. Since $|\mathcal{A}| = 1$ and $i = 1$, by Step 3, we pick the only element $H = H_2$ in $\mathcal{A}$ and let $\widetilde{H} \leftarrow H_2$. Since $c_3 \in \Gamma_{H_2}(3) = \Gamma(3)$ and $c_3 \in \Gamma(2) \wedge 2 \in Q_{H_2} \wedge c_3 \notin \Gamma_{H_2}(2)$, by (6), $(3, 2) \in \mathcal{T}_{spec}(H_2, A, G)$. Moreover, since $\Theta_D^N(c_1o_1) \cap \Theta_D^N(c_1o_2) \neq \emptyset$, $(3, 2) \in \mathcal{T}_{conf}(\Theta_D^N)$. Hence, we have $\mathcal{T}_{spec}(H_2, A, G) \cap \mathcal{T}_{conf}(\Theta_D^N) \neq \emptyset$. By Proposition 6, $\mathcal{L}(H_2)$ is not relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$. By Step 3, $\mathcal{R} \leftarrow \{H_1\}$. By Step 4, $i = 2$.

Consider the second iteration. Since $\mathcal{R} = \{H_1\}$, $H_1 \notin \mathcal{A}$, and H_1 is controllable, by Step 2, $\mathcal{A} = \{H_2, H_1\}$. Since $i = 2$ and $|\mathcal{A}| = 2$, by Step 3, pick the second element of $\mathcal{A}$, i.e., H_1. To achieve H_1, we need to disable c_3 at both state 2 and state 3 in H, and enable c_3 at state 6, i.e., $c_3 \notin \Gamma_{H_1}(2) \wedge c_3 \notin \Gamma_{H_1}(3) \wedge c_3 \in \Gamma_{H_1}(6)$ Since $c_3 \in \Gamma(2) \wedge c_3 \in \Gamma(3) \wedge c_3 \in \Gamma(6)$ and $2, 3, 6 \in Q_{H_1}$, by (6), $\mathcal{T}_{spec}(H_2, A, G) = \{(6, 2), (6, 3)\}$. Moreover, by (4), the set of confusable state pairs in G under P is $\mathcal{T}_{conf}(P) = \{(q, q) : q \in Q\} \cup \{(2, 4), (3, 8), (6, 8)\}$. By Proposition 4,

$$\begin{aligned}\mathcal{T}_{conf}(\Theta_D^N) = \{&(0, 1), (0, 2), (0, 3), (1, 2), (1, 3), (2, 3), \\ &(2, 4), (2, 5), (4, 5), (5, 6), (6, 7), (6, 8), \\ &(7, 8), (2, 7), (3, 7), (3, 8)\} \cup \{(q, q) : q \in Q\}.\end{aligned}$$

Note that in $\mathcal{T}_{conf}(P)$ and $\mathcal{T}_{conf}(\Theta_D^N)$, we only list (q, q') and omit (q', q) for brevity. Then, it can be verified that $\mathcal{T}_{spec}(H_1, A, G) \cap \mathcal{T}_{conf}(\Theta_D^N) = \emptyset$. By Proposition 6, $\mathcal{L}(H_1)$ is relatively delay observable w.r.t. C, Θ_D^N, and $\mathcal{L}(G)$. By Step 3, $H^* \leftarrow H_1$. By Corollary 2, $H^\uparrow = H^*$ and $H^\uparrow$ is unique.

To pass the intersection safely with observation delays, by $H^\uparrow = H_1$, vehicle x needs to stop when it approaches the intersection no matter whether the traffic signal in the forward direction is red or not. In other words, it should enable c_2 and disable c_3 when the system is in 2 and 3. After that, vehicle x can pass the intersection (enabling c_3) at the time the traffic signal in the forward direction is switched to green (o_2 occurs at state 5).

VI. CONCLUSION

In this article, we simplify the existence condition for a deterministic networked supervisor introduced in [1] and prove there exists a deterministic networked supervisor that can be used to achieve the specification language if the specification language is controllable and delay observable. To overcome the difficulty delay observability is not closed under union, relative delay observability is introduced. Techniques are proposed for checking relative delay observability. To calculate the supremal controllable and relatively delay observable sublanguage for the specification language, a general algorithm is introduced to calculate the supremal sublanguage (of the specification language) that satisfies controllability and another given language property. We prove such a unique supremal element exists if the given language property is closed under union, and there exits at least one element in the finite solution space that satifies both controllability and the given property.

REFERENCES

[1] S. Shu and F. Lin, "Deterministic networked control of discrete event systems with nondeterministic communication delays," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 190–205, Jan. 2017.

[2] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, Jan. 1987.

[3] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inf. Sci.*, vol. 44, no. 3, pp. 173–198, Apr. 1988.

[4] R. Cieslak, C. Desclaux, A. S. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Trans. Autom. Control*, vol. 33, no. 3, pp. 249–260, Mar. 1988.

[5] S. Takai and T. Ushio, "A modified normality for decentralized supervisory control of discrete event systems," *Automatica*, vol. 38, no. 1, pp. 185–189, 2002.

[6] S. Takai, R. Kumar, and T. Ushio, "Characterization of co-observable languages and formulas for their super/sublanguages," *IEEE Trans. Autom. Control*, vol. 50, no. 4, pp. 434–447, Apr. 2005.

[7] K. Cai, R. Zhang, and W. M. Wonham, "Relative observability of discrete-event systems and its supremal sublanguages," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 659–670, Mar. 2015.

[8] M. V. S. Alves, L. K. Carvalho, and J. C. Basilio, "New algorithms for verification of relative observability and computation of supremal relatively observable sublanguage," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5902–5908, Nov. 2017.

[9] F. Lin, "Control of networked discrete event systems: Dealing with communication delays and losses," *SIAM J. Control Optim.*, vol. 52, no. 2, pp. 1276–1298, Jan. 2014.

[10] S. Shu and F. Lin, "Decentralized control of networked discrete event systems with communication delays," *Automatica*, vol. 50, no. 8, pp. 2108–2112, Aug. 2014.

[11] S. Shu and F. Lin, "Supervisor synthesis for networked discrete event systems with communication delays," *IEEE Trans. Autom. Control*, vol. 60, no. 8, pp. 2183–2188, Aug. 2015.

[12] S. Shu and F. Lin, "Predictive networked control of discrete event systems," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4698–4705, Sep. 2017.

[13] Z. Liu, X. Yin, S. Shu, and S. Li, "Online supervisory control of networked discrete-event systems with control delays," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 6706–6711.

[14] B. Zhao, F. Lin, C. Wang, X. Zhang, M. P. Polis, and L. Y. Wang, "Supervisory control of networked timed discrete event systems and its applications to power distribution networks," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 2, pp. 146–158, Jun. 2017.

[15] A. Rashidinejad, M. Reniers, and L. Feng, "Supervisory control of timed discrete-event systems subject to communication delays and non-fifo observations," in *Proc. 14th Int. Workshop Discrete Event Syst. (WODES)*, 2018, pp. 456–463.

[16] F. Lin, W. Wang, L. Han, and B. Shen, "State estimation of multichannel networked discrete event systems," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 53–63, Mar. 2020.

[17] Y. Yao, Y. Tong, and H. Lan, "Initial-state estimation of multi-channel networked discrete event systems," *IEEE Control Syst. Lett.*, vol. 4, no. 4, pp. 1024–1029, Oct. 2020.

[18] M. V. S. Alves, L. K. Carvalho, and J. C. Basilio, "Supervisory control of networked discrete event systems with timing structure," *IEEE Trans. Autom. Control*, early access, Jul. 7, 2020, doi: 10.1109/TAC.2020.3007384.

[19] C. E. V. Nunes, M. V. Moreira, M. V. S. Alves, L. K. Carvalho, and J. C. Basilio, "Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation," *Discrete Event Dyn. Syst.*, vol. 28, no. 2, pp. 215–246, Jun. 2018.

[20] G. S. Viana, M. S. Alves, and J. C. Basilio, "Codiagnosability of timed networked discrete-event systems subject to event communication delays and intermittent loss of observation," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 4211–4216.

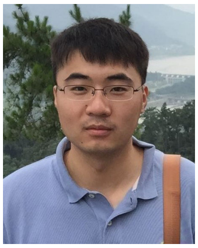
[21] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York, NY, USA: Springer, 2008.

[22] Y. Hou, W. Wang, Y. Zang, F. Lin, M. Yu, and C. Gong, "Relative network observability and its relation with network observability," *IEEE Trans. Autom. Control*, vol. 65, no. 8, pp. 735–3584, Nov. 2020.

[23] P. Xu, S. Shu, and F. Lin, "Nonblocking networked control of discrete event systems," in *Proc. Chin. Automat. Congr. (CAC)*, Oct. 2017, pp. 1911–1916.

[24] W. Wang, S. Lafortune, and F. Lin, "An algorithm for calculating indistinguishable states and clusters in finite-state automata with partially observable transitions," *Syst. Control Lett.*, vol. 56, nos. 9–10, pp. 656–661, Sep. 2007.

[25] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM J. Control Optim.*, vol. 25, no. 3, pp. 637–659, May 1987.



YUNFENG HOU received the B.Eng. degree in electrical engineering from the Shandong University of Technology, in 2012, and the M.S. and Ph.D. degrees in control science and engineering from the University of Shanghai for Science and Technology, in 2015 and 2020, respectively. He is currently a Research Associate with the Institute of Machine Intelligence, University of Shanghai for Science and Technology. His research interests include modeling, control and optimization in discrete-event systems, intelligent transportation systems, and cooperative control for multi-robot systems.

YUTAO CHEN is currently pursuing the bachelor's degree in electronic engineering with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology. His research interests include multi-agent systems and robotics.

QINGDU LI received the B.S. degree in electronic engineering and the M.S. degree in microelectronics engineering from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2002 and 2004, respectively, and the Ph.D. degree in engineering from the Huazhong University of Science and Technology, in 2008. He was a Postdoctoral Researcher with the Department of Theoretical and Applied Mechanics, Cornell University, in 2011, and a Research Fellow with the Department of Informatics, University of Hamburg, in 2017. He is currently a Professor with the Institute of Machine Intelligence, University of Shanghai for Science and Technology, China. His research interests include biped robots, dynamic walking, nonlinear circuits, complex systems, and bifurcation and chaos. He serves as an Editorial Board Member for the Complexity. He was a recipient of the Chongqing Natural Science Award (Second Prize), in 2016.

YUNFENG JI (Associate Member, IEEE) received the B.S. and M.S. degrees from the University of Shanghai for Science and Technology, Shanghai, China, in 2012 and 2015, respectively, and the Ph.D. degree from the China Table Tennis College, Shanghai, in 2018. He was a Senior Visiting Scholar with the Department of Informatics, University of Hamburg, Hamburg, Germany, in 2018. He is currently with the Institute of Machine Intelligence, University of Shanghai for Science and Technology. His research interests include table tennis robots, image processing, machine learning, and object detection.