OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Deterministic supervisory control with flexible upper-bounds on observation delay and control delay
Author(s)	Wang, Weilin; Zang, Yanwei; Takai, Shigemasa et al.
Citation	Automatica. 2022, 144, p. 110480
Version Type	AM
URL	https://hdl.handle.net/11094/106741
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Deterministic Supervisory Control with Flexible Upper-bounds on Observation Delay and Control Delay ⋆

Weilin Wang [a,b], Yanwei Zang [b,c], Shigemasa Takai [d], Lachlan L. H. Andrew [e], Chaohui Gong [a,b]

[a] *Department of Control Science & Engineering, University of Shanghai for Science & Technology, China*

[b] *Zhejiang Research Center on Smart Rail Transportation, China*

[c] *Huadong Engineering Corporation Limited, China*

[d] *Division of Electrical, Electronic & Information Engineering, Osaka University, Japan*

[e] *School of Computing & Information Systems, University of Melbourne, Australia*

Abstract

This paper considers supervisory control of discrete event systems with observation delay and control delay. We first extend the formulation of networked supervisory control in the literature by allowing both the upper bound of observation delay and the upper bound of control delay vary among different event strings, and accommodating dynamic observation in which past influence whether or not an occurrence of an event is observable. We then combine observation delay and control delay and prove that satisfying both controllability and delay observability over the combined delay is necessary and sufficient for the existence of a supervisor by which the desired language can be obtained deterministically. After that, we reduce the type of problems concerning the networked supervisory control to the corresponding type of problems concerning traditional supervisory control without any delay on observation or control. As an immediate application, the verifier of observability can be used to test the corresponding delay observability. Finally, this reduction is used to leverage methods for minimizing sensor activations for supervisory control without control delay to minimize sensor activations with control delay.

Key words: Supervisory control, distributed discrete-event systems, command-control systems, automata, sensor activations.

1 Introduction

Supervisory control for discrete event systems (DES) has been investigated for decades (Ramadge & Wonham 1987, Lin & Wonham 1988*a*,*b*, Balemi & Brunner 1992, Lin 1993, Heymann & Lin 1994). Its objective is to compute a supervisor to prevent event occurrences that lead to undesired event strings. For example, a supervisor of a traffic responsive signaling system may wish not to switch on a green light of a path immediately after red lights of the conflicting paths turned on, but to wait until the intersection has cleared up, in this case events include signals on/off, an intersection clearing up, etc. The majority of supervisory control builds on the work of (Ramadge & Wonham 1987). In this framework, if it is possible for not all events to be observed, the condition that is necessary and sufficient for the existence of a supervisor is to satisfy both controllability (Ramadge & Wonham 1987) and observability (Lin & Wonham 1988*b*). Controllability says that, if an event occurrence would lead the system to produce some undesired strings, the supervisor can prevent that event from occurring, i.e., that event needs to be controllable. Observability says that, if two strings can lead to the same partial observation to the supervisor, the control action (event enablement/disablement) needed after the occurrence of these two strings must be consistent.

Supervisory control has been extended in several other directions, including decentralized control (Lin & Wonham

⋆ This work is supported in part by PowerChina Grant KY2018-JT-20-01-2019, PowerChina Grant KY2020-JT-12-01-2022, National Natural Science Foundation of China Grant CNSF-61374058, Australian Research Council Grant DP-130100156.
Corresponding author: Chaohui Gong, Tel. (+86)-187-2189-7136.

Email addresses: wlwang@usst.edu.cn (Weilin Wang), zang_yw@ecidi.com (Yanwei Zang), takai@eei.eng.osaka-u.ac.jp (Shigemasa Takai), lachlan.andrew@unimelb.edu.au (Lachlan L. H. Andrew), chgong@usst.edu.cn (Chaohui Gong).

1988*a*), where multiple supervisors are applied to control the system; robust control (Lin 1993), where the controlled system is not entirely known; and online control (Heymann & Lin 1994), where a control action is calculated based on the current observation of the system string. When agents activate or deactivate sensors for observing events (Wang, Lafortune, Lin & Girard 2010, Wang 2016) or decide when and to whom to communicate which event occurs (Wang et al. 2008), the observation or non-observation of an occurrence of an event depends on the prior events; this is termed *dynamic* observation. Wang et al. (2007), Huang et al. (2008), Wang et al. (2011) investigated supervisory control under dynamic observations. The set of strings a system can produce is called its language. Another extension considers the case when no supervisor can achieve the desired language exactly, one task is to calculate a maximally permissive supervisor, which synthesizes its maximal controllable and observable sublanguage (Yin & Lafortune 2016). Further Yin & Lafortune (2017*b*) consider range control, whose range is given by both an upper bound specification for the legal behavior and a lower bound specification for the desired behavior. Considering extended models, Ouedraogo et al. (2011) investigate how to synthesize a non-blocking supervisor for systems modeled by finite automata augmented with variables and predicates over them. In addition, Rashidinejad et al. (2020) extend those results for systems modeled by timed automata with resettable local timers reset by *forcible* events, which are introduced to avoid the explosion of the space of timer-reading labeled states. The above works mainly focus on supervisory control with purely logical specifications. Recently, the supervisory control theory has been extended to handle quantitative specifications; see, e.g., (Ji et al. 2021, 2022).

Most of the above work assumes the network is ideal. Any supervisor can observe any observable instances of an event occurring instantly; any control commands are executed reliably and immediately; the communications among supervisors are instantaneous and without any loss. These are not valid assumptions in practices. For example, distractions while driving cause delays and losses in observation and delays in controlling vehicles, which lead to most traffic accidents.

To investigate the supervisory control with an imperfect network between the supervisor and the plant, Balemi & Brunner (1992) proposed a framework that is different from the one by Ramadge & Wonham (1987). In (Balemi & Brunner 1992, Balemi 1992), the controlled system is modeled as the connection of two input/output processes: the plant and the supervisor. Each of them outputs events in response to the inputs from the other. A process observes all outputs of itself immediately, and the output events of the other could be delayed, which causes them to appear in the string of observation later than they do in reality. Assuming all occurrences of all events are observable to both processes despite the delay, their problem is to compute a non-blocking supervisor, by which a nonempty subset of the desired language can be obtained. Although this framework has modeling flexibility in representing strings of control actions, applying it to design the supervisor is complex, and few further results are available.

The majority of work concerning the delay in supervisory control uses the framework from (Ramadge & Wonham 1987, Lin & Wonham 1988*b*). In addition to all occurrences of all events being observable, Park & Cho (2006, 2007) assume that each controllable event is disabled by default and cannot occur if any of the local supervisors disables it. Recent research, including this paper, allows some occurrences of events to be unobservable, and focuses on the existence and construction of a supervisor that tolerates the delays in observation and control (Lin 2014, Shu & Lin 2014, 2015, 2016, 2017). Shu & Lin (2017) assume that an event occurrence is disabled if and only if it causes the system to leave the desired language, which is inherited by this paper. Such a supervisor is different from the one presented in Lin (2014), Shu & Lin (2016) because that work tolerated excluding a string within the desired language in some situations, as long as that string could occur in at least one situation. More recently, Liu et al. (2022) proposed a novel online approach that completely solves the networked supervisory control problem under control delays.

Another approach is to model the delays in observation and control in units of time represented by the number of periodical occurrences of a "tick" event instead of regular events. For example, Rashidinejad et al. (2018) show how to synthesize a supervisor to control a system in which both observation delay and control delay are constant. In (Rashidinejad et al. 2018), as the system modeled by a timed automaton, all observations occur N_o ticks after the event, and all implementations of control decisions occur N_c ticks after their issuance, although a non-deterministic extension to variable delays is briefly proposed in (Rashidinejad 2021, p. 49).

Consider events being observed out of order. In (Balemi 1992), a processor observes events from itself immediately, and can also observe events from another process after some delay. Similarly, in (Lin et al. 2019) events are divided into disjoint subsets such that different channels are used to communicate events in different subsets and have different upper bounds of communication delay. All these cause the observation order to differ from the order of generation. In contrast, Rashidinejad et al. (2018) allow events that occur between the same pair of ticks of the clock to be observed in an arbitrary order. Balemi (1992), Lin et al. (2019) and Rashidinejad et al. (2018) focus on supervision that tolerate observed strings containing limited reordering induced by the delays in their model, whereas Shu & Lin (2014, 2015, 2017), Liu et al. (2022) and this paper focus on uncertainties in the delays and assume FIFO delivery to retain tractability.

There are two problems that have been considered, that are related to ours. One is considering imperfect networks allowing supervisors to communicate with each other, matched with instantaneous communication with a plant (Tripakis 2004, Zhang et al. 2016). The other is designing supervisors

to be robust against a malicious adversary, which can alter the event observations to manipulate the system (Su 2018, Meira-Góes et al. 2020).

This paper builds on (Lin 2014, Shu & Lin 2014, 2015, 2016, 2017) by considering the existence and the construction of a supervisor by which an event occurrence is disabled if it causes the system to leave the desired language, under the observation delay and control delay. This paper assumes no observation and control losses exist, both observation delay and control delay are bounded above in terms of the numbers of events, which is similar to (Shu & Lin 2014, 2015, 2017). Moreover, the actuator always uses the most recently received control command.

Comparing with (Lin 2014, Shu & Lin 2014, 2015, 2016, 2017), this paper has distinctive features as follows. This paper allows upper bounds of observation delay and upper bounds of control delay to depend on the string processed so far, which is in contrast with the fixed upper bounds of the works mentioned above. Moreover, for supervisor design, we then merge these two types of upper bounds to define one upper bound of observation-plus-control delay as a function over the system language. In addition, the works above assume that all occurrences of an event are either always observed or never observed. In this paper, we study the impact of observation-plus-control delay under dynamic observations. Then we reduce the type of problems concerning deterministic networked control under delayed dynamic observations and control to the corresponding type of problems concerning supervisory control under non-delayed dynamic observations and control. Moreover, we define the upper bound of observation-plus-control delay over the state space. This imposes structural restrictions to the upper bound over the language, but is more flexible than the fixed upper bound in (Lin 2014, Shu & Lin 2014, 2015, 2016, 2017). With a modest complexity, we reduce the relevant supervisory control problems to the corresponding problems without delay. If a supervisor exists, it can apply the resulted model to control the system as if no delay involved.

As many techniques have been developed to solve problems concerning supervisory control, this reduction leverages these techniques to solve the corresponding problems concerning deterministic networked supervisory control. As an example, we reuse a verifier designed for observability without any delay to test the observability with delayed observation and delayed control. For a second example, different approaches have been proposed to optimize sensor activation policy (SAP) while preserving a particular system property such as observability (Wang, Lafortune, Lin & Girard 2010, Wang 2016, Wang & Gong 2017, Yin & Lafortune 2017*a*), diagnosability (Cassez & Tripakis 2008, Wang, Lafortune, Girard & Lin 2010), or detectability (Shu et al. 2013), etc. We present techniques to use the observability-preserving methods that optimize SAP for supervisory control without control delay to optimize SAP for supervisory control with control delay.

The rest of this paper is organized as follows. In Section 2, we present a system model. We then show in Section 3 the reduction from the type of problems concerning supervisory control with both observation delay and control delay to the type of problems concerning supervisory control without these delays, and apply this reduction to leverage the verifier by Wang et al. (2007) to verify observability with both observation delay and control delay. In Section 4, techniques for minimizing SAP without observation and control delay by Wang, Lafortune, Lin & Girard (2010), Yin & Lafortune (2017*a*) are used to compute a minimal SAP with a delay in control. Finally, we conclude the paper in Section 5.

2 Supervisory control under delayed observation and delayed control

2.1 Preliminaries

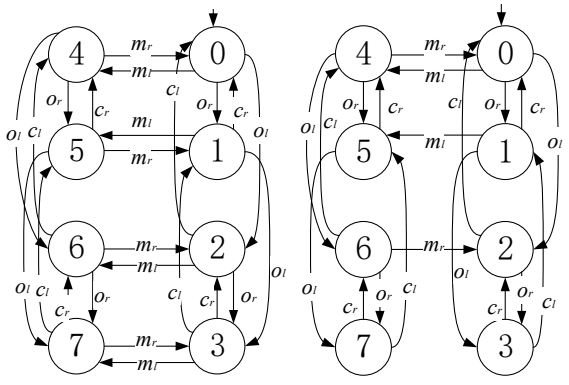
A DES is modeled by a deterministic automaton $G = (Q, \Sigma, \delta, \Gamma, q_0)$, where Q is a set of states; Σ is a set of events; $\delta : Q \times \Sigma \to Q$ is a transition partial function; $\Gamma : Q \to 2^{\Sigma}$ is a set of active events when the system is in a given state; and q_0 is a designated initial state.

Let Σ^* denote the Kleene closure of Σ. Transition function δ is extended to $\delta : Q \times \Sigma^* \to Q$ as usual by $\delta(q, s\sigma) = \delta(\delta(q,s),\sigma)$. Let $\mathscr{L}(G) = \{s \in \Sigma^* : \delta(q_0, s) \text{ is defined}\}$ denote the language generated by G. Assume that G is accessible, i.e., any state in Q can be reached by some string in $\mathscr{L}(G)$ starting from q_0. Let $\Sigma_o \subseteq \Sigma$ denote the set of observable events and $\Sigma_{uo} = \Sigma \setminus \Sigma_o$ denote the set of unobservable events. Let $\Sigma_c \subseteq \Sigma$ denote the set of controllable events and $\Sigma_{uc} = \Sigma \setminus \Sigma_c$ denote the set of uncontrollable events. Each of the controllable events can be effectively disabled by a "controller", defined in Section 2.2.

Given $s \in \Sigma^*$, $\overline{s} = \{t : s = tt'\}$ is the set of all prefixes of s. A language is called prefix-closed if every prefix of a string in that language is also in the language. Given language L, $\overline{L} = \{t \in \overline{s} : s \in L\}$ is the prefix closure of L. $|s|$ is the number of event occurrences in s. Given an $s \in \mathscr{L}(G)$, let s_{-i} be the prefix of s with $|s_{-i}| = \max\{|s| - i, 0\}$. Given languages L_1 and L_2, let $L_1L_2 = \{st : s \in L_1 \wedge t \in L_2\}$ be the language collecting strings each of which formed by concatenating a string in L_1 with a string L_2. We distinguish between an *event*, which is a symbol $\sigma \in \Sigma$, and an *occurrence* of the event, which is a point in a string $s \in \Sigma^*$ at which σ occurs.

Let $\mathbb{N}$ be the set of natural numbers including 0 and for $N \in \mathbb{N}$, let $[N] = \{0, 1, \ldots, N\}$.

Definition 1 (Observation mapping ω and information mapping θ^{ω}) *Observations are said to be dynamic if whether or not an event occurrence is observable is described by an observation mapping $\omega : \mathscr{L}(G) \to 2^{\Sigma_o}$ such that given $s\sigma \in \mathscr{L}(G)$ with $\sigma \in \Sigma$, σ is observable after s iff $\sigma \in \omega(s)$. Given ω, the information mapping*



(a) System G (b) Desired system H

Fig. 1. Automata G and H in Example 1.

$\theta^\omega : \mathscr{L}(G) \to \Sigma_o^*$ is defined iteratively as, for the empty string ε, $\theta^\omega(\varepsilon) = \varepsilon$, and for all $s, s\sigma \in \mathscr{L}(G)$ with $\sigma \in \Sigma$, $\theta^\omega(s\sigma) = \theta^\omega(s)\sigma$ if $\sigma \in \omega(s)$, and $\theta^\omega(s)$, otherwise.

Given ω' and ω'' defined on $\mathscr{L}(G)$, $\omega' \subseteq \omega''$ means that, for all $s \in \mathscr{L}(G)$, $\omega'(s) \subseteq \omega''(s)$. $\omega = \omega' \cup \omega''$ means that, for all $s \in \mathscr{L}(G)$, $\omega(s) = \omega'(s) \cup \omega''(s)$. $\omega' \subset \omega''$ means that $\omega' \subseteq \omega''$ and not $\omega' = \omega''$. Given languages L and $L' \subseteq L$ and observation mapping $\omega : L \to 2^{\Sigma_o}$, $\omega|_{L'} : L' \to 2^{\Sigma_o}$ denote the sub-function of ω that has domain L'. Given languages L, $\omega^{all} : L \to \Sigma_o$ denotes the observation mapping that corresponds to all events in Σ_o always being observed.

2.2 Deterministic networked control for DES

In this subsection, we motivate this study, generalize the observation problem for deterministic networked control by Shu & Lin (2017), and solve the generalized problem.

The following example shows an engineering imperative of investigating delayed observation and delayed control.

Example 1 *Consider a simple road transport model of a vehicle x changing lane in a two-lane road, shown in Fig. 1. In this model, the event set is $\Sigma = \{m_l, m_r, o_l, o_r, c_l, c_r\}$ with the following events: m_l occurs when x changes into the left lane, m_r occurs when x changes into the right lane, o_l occurs when another vehicle enters the part of the left lane that is too close to x when that part is empty, o_r occurs when another vehicle enters the part of the right lane that is too close to x while that part is empty, c_l occurs when the last vehicle in the part of the left lane that is too close to x leaves that part, c_r occurs when the last vehicle in the part of the right lane that is too close to x leaves that part. As in practice, we have $\Sigma_c = \{m_l, m_r\}$ and $\Sigma_o = \Sigma$. Changing lane is unsafe when the part of the targeted lane is occupied. Hence, comparing desired system H with uncontrolled system G, m_l is not defined in states 2 and 3 in H but is defined in those states in G, and m_r is not defined in states 5 and 7 in H but is defined in those states in G. Without observation delay and control delay, this system is controllable and observable. Any occurrences of m_l and m_r can be immediately observed. But the observation of all other events are dynamic, which is consistent with the fact that the driver only checks whether the corresponding part of lane is occupied when necessary. Moreover, observing o_l, o_r, c_l, and c_r usually involves delay, as the driver observes whether the part of lane is occupied or empty, instead of directly observing these events. The observation delays and control delays of this system could cause m_r to be enabled at states 5 or 7 and m_l to be enabled at state 2 or 3, each of which could lead to a collision. We will not limit ourselves to transport applications, but will develop techniques for systematically handling observation delays and control delays under dynamic observation in supervisory control.*

As defined in (Lin 2014, Shu & Lin 2014, 2015, 2016, 2017), the observation delay is the number of occurrences of events in between an event occurring and the supervisor observing that occurrence. Similarly the control delay is the number of occurrences of events in between the supervisor issuing an control command and that command being implemented. In the works mentioned above, the observation delays and control delay are both non-deterministic but bounded above by fixed numbers of events for all strings. In contrast, we allow both of these upper bounds to depend on strings. We then show that for computing a supervisor, they can be combined into a single upper bound of observation-plus-control delay.

Definition 2 (Observation delay) *Recall that, given an $s \in \mathscr{L}(G)$, s_{-i} is the prefix of s with $|s_{-i}| = \max\{|s| - i, 0\}$. Observation delays are random. We say that an $s \in \mathscr{L}(G)$ has an observation delay i in a particular instance if the current string of the system is s and, at that time, the most recent observation made by the supervisor is string s_{-i}. Let $\Theta^\omega(s,n) = \{\theta^\omega(s_{-i}) : i \in [n]\}$ denote the set of information mappings of s_{-i}, $i \in [n]$, under ω.*

Note that, in our model, the observed string is an information mapping of a prefix of the string that actually occur. So, with or without delay, the order of events in the string observed and the order of events in the string that actually occur remain the same. Also note that $\Theta^\omega(s,n)$ is also termed (delayed) information mapping by Lin (2014).

With the concept of observation delay being cleared, we define the upper bounds of observation delay and control delay as follows.

Definition 3 (Bounds of observation and control delays) *A DES is said to have a observation delay bound $N_o : \mathscr{L}(G) \to \mathbb{N}$ if N_o satisfies the following. For all $s \in \mathscr{L}(G)$, $N_o(s) \leq |s|$ and s may have delay i iff $i \in [N_o(s)]$. Correspondingly, given ω, $\Theta^\omega(s, N_o(s))$ is the set of possible strings that might be observed by the supervisor in the half-open interval of time starting from the occurrence of the last event in s and ending just before the occurrence of the first event after s. Moreover, a DES is said to have a control delay bound $N_c : \mathscr{L}(G) \to \mathbb{N}$ if N_c satisfying the following. If $|t| \in [N_c(s)]$ then a control command issued following an*

$s \in \mathscr{L}(G)$ could be implemented following st given that st could occur. If $|t| \notin [N_c(s)]$ then the command cannot be implemented following st.

If $N_o(s\sigma) > N_o(s)+1$ for some $\sigma \in \Sigma$, then the supervisor can become aware of the symbol at position $|s|-N_o(s)$, but then be unaware of that same symbol after the extra symbol σ occurs. This does not reflect the physical process of information gathering, which is why we impose the requirement of $N_o(s\sigma) \leq N_o(s)+1$. Similarly, if $N_c(s\sigma) < N_c(s)-1$, it becomes possible that the control command issued following s implemented after the control command issued after $s\sigma$ being implemented. We should avoid this during the process of designing the control system. Therefore, we impose Assumption 1 on the upper-bound of observation delay and the upper-bound of control delay.

Assumption 1 *For all $s, s\sigma \in \mathscr{L}(G)$ with $\sigma \in \Sigma$, $N_o(s\sigma) \leq N_o(s)+1$ and $N_c(s\sigma) \geq N_c(s)-1$.*

Let $K \subseteq \mathscr{L}(G)$ be the specification language for a given control objective. That is, we wish to control the DES such that only strings in K can be generated. We assume that K can be generated by an automaton $H = (Q_D, \Sigma, \delta_D, \Gamma_D, q_0)$, which is a sub-automaton of G, denoted by $H \sqsubseteq G$. Correspondingly, $Q_D \subseteq Q$ and $\delta_D \subseteq \delta$. In this case, both $\mathscr{L}(G)$ and K are prefix closed, as defined in (Cassandras & Lafortune 2009). Assume that H is accessible.

Control is performed by a networked supervisor defined in (Lin 2014, Shu & Lin 2016), which controls the processing of event occurrences subject to Σ_c, N_o, and N_c. The networked supervisor is a function $S : \theta^{\omega}(\mathscr{L}(G)) \to 2^{\Sigma}$ where $S(t), t \in \theta^{\omega}(\mathscr{L}(G))$, is a set of events allowed to occur when the networked supervisor observes t. Following (Wonham & Ramadge 1987), we restrict S such that for all t, $S(t) \supseteq \Sigma_{uc}$ if $S(t)$ is defined, which models the fact that an occurrence of an uncontrollable event is unpreventable.

To use the upper-bounds defined in Definition 3, we extend the definitions in (Lin 2014, Shu & Lin 2016) of the smallest language and of the largest language generated by the supervised system.

Definition 4 *Given system G with observation mapping ω, observation delays bounded by N_o, and control delays bounded by N_c, iteratively define the smallest language (lower bound) $\mathscr{L}_r(G,S)$ generated by the supervised system as: Start with $\varepsilon \in \mathscr{L}_r(G,S)$, and repeatedly apply*

$$(\forall \sigma \in \Sigma)[s\sigma \in \mathscr{L}_r(G,S) \Leftrightarrow (s\sigma \in \mathscr{L}(G) \wedge s \in \mathscr{L}_r(G,S) \wedge (\forall i)[N_c(s_{-i}) \geq i \Rightarrow (\forall t \in \Theta^{\omega}(s_{-i}, N_o(s_{-i})))\ \sigma \in S(t)])] \quad (1)$$

and iteratively define the largest language (upper bound) $\mathscr{L}_a(G,S)$ generated by the supervised system as: Start with $\varepsilon \in \mathscr{L}_a(G,S)$, and repeatedly apply

$$(\forall \sigma \in \Sigma)[s\sigma \in \mathscr{L}_a(G,S) \Leftrightarrow (s\sigma \in \mathscr{L}(G) \wedge s \in \mathscr{L}_a(G,S) \wedge (\exists i)[N_c(s_{-i}) \geq i \wedge (\exists t \in \Theta^{\omega}(s_{-i}, N_o(s_{-i})))\ \sigma \in S(t)])] \quad (2)$$

Here $\mathscr{L}_r(G,S)$ and $\mathscr{L}_a(G,S)$ generalize the corresponding definitions in (Shu & Lin 2017) by allowing upper bounds of delays to depend on which string follows that after which the control command is issued. Language $\mathscr{L}_r(G,S)$ includes all strings, that are not possibly excluded by the supervisor S. In other words, for all $s\sigma \in \mathscr{L}_r(G,S)$ with some $\sigma \in \Sigma$, all possible control commands, in term of the set of events allowed to occur, implemented following s include σ. In contrast, $\mathscr{L}_a(G,S)$ include all strings that may occur under the control of S, including strings in $\mathscr{L}_r(G,S)$ and those $s\sigma$ with σ which is disabled following s in some circumstances but is allowable following s in at least one circumstance. Technically, the only difference between the definitions $\mathscr{L}_r$ and $\mathscr{L}_a$ is the use of universal versus existential quantifiers in the last line. In fact, the networked supervisor may have different observations for the same string and the corresponding control commands adopted are therefore different as discussed by (Shu & Lin 2016). [1] Even for the same observations, the controls may be different due to control delays. As shown by (Shu & Lin 2017), the supervised system can potentially generate many languages (nondeterministically) between $\mathscr{L}_r(G,S)$ and $\mathscr{L}_a(G,S)$.

We define predicate P to indicate the existence of a supervisor under which the desired language K can be obtained deterministically.

Definition 5 *Consider a networked DES G with the set of uncontrollable events Σ_{uc}, desired language K, observation mapping ω, upper bounds of observation delays given by N_o, and upper bounds of control delays given by N_c. Let $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega, N_c, N_o)$ be true if there exists a networked supervisor S such that $\mathscr{L}_r(G,S) = \mathscr{L}_a(G,S) = K$, and false otherwise.*

Given an $s \in \mathscr{L}(G)$, Definition 6 captures the length difference between the shortest $s' \in \bar{s}$ and s such that, if a control command is issued right after an observation of s' is made, that command may be implemented following s due to observation-plus-control delay defined below.

Definition 6 (Observation-plus-control delay) *Given N_c, N_o, and $s \in \mathscr{L}(G)$, define the upper bound of observation-plus-control delay $N_{o+c} : \mathscr{L}(G) \to \mathbb{N}$ as, for each $s \in \mathscr{L}(G)$,*

$$N_{o+c}(s) = \min\{|s|, \max\{N_o(s_{-i}) + i : i \in [N_c(s_{-i})]\}\} \quad (3)$$

[1] The definition of network observability, found in (Lin 2014), is omitted for brevity.

Lemma 1 *N_{o+c} satisfies the part for observation delay of Definition 3 and of Assumption 1, i.e.,*

(1) For all $t \in \mathscr{L}(G)$, $N_{o+c}(t) \leq |t|$.
(2) For all $s\sigma \in \mathscr{L}(G)$ such that $\sigma \in \Sigma$, $N_{o+c}(s\sigma) \leq N_{o+c}(s)+1$.

PROOF. The first part follows directly from (3).

For the second part, consider an arbitrary $\sigma \in \Sigma$ and an arbitrary $s\sigma \in \mathscr{L}(G)$. Let $k = \arg\max_{i\in\mathbb{N}}\{N_o((s\sigma)_{-i})+i : i \in [N_c((s\sigma)_{-i})]\}$. Suppose $k = 0$. The minimum of the right-hand-side of (3) takes the second term. By the part of Assumption 1 concerning observation delay and (3),

$$N_{o+c}(s\sigma) = N_o(s\sigma) \leq N_o(s)+1 \leq N_{o+c}(s)+1$$

Suppose $1 \leq k \leq |s\sigma|$. The minimum of the right-hand-side of (3) takes the second term. By (3), $k \in [N_c((s\sigma)_{-k})]$. Then, by the part of Assumption 1 concerning control delay and $(s\sigma)_{-k} = s_{-(k-1)}$, we have $k-1 \in [N_c(s_{-(k-1)})]$. Thus, $N_{o+c}(s\sigma) = N_o((s\sigma)_{-k})+k = N_o(s_{-(k-1)})+(k-1)+1 \leq N_{o+c}(s)+1$. Suppose $k > |s\sigma|$. The minimum of the right-hand-side of (3) takes the first term. Note that, by Definition 3, for all i, $N_o((s\sigma)_{-i}) \leq |(s\sigma)_{-i}|$. Then, by (3), we have $N_c(s_{-|s|}) = N_c(\varepsilon) = N_c((s\sigma)_{-|s\sigma|}) > |s\sigma| > |s|$. Then, also by (3), $N_{o+c}(s\sigma) = |s\sigma| = |s|+1 = N_{o+c}(s)+1$. Overall, since σ and $s\sigma$ are arbitrary, we complete the proof. □

Lemma 2 *Given an arbitrary $s \in \mathscr{L}(G)$, we have*

$$\Theta^{\omega}(s,N_{o+c}(s)) = \bigcup_{i:N_c(s_{-i})\geq i} \Theta^{\omega}(s_{-i},N_o(s_{-i})) \tag{4}$$

PROOF. Let L denote the language of the right-hand-side of (4). By Definition 6 and the definition of Θ^{ω}, for all i such that $N_c(s_{-i}) \geq i$, we have $\Theta^{\omega}(s,N_{o+c}(s)) \supseteq \Theta^{\omega}(s_{-i},N_o(s_{-i}))$. Hence, $\Theta^{\omega}(s,N_{o+c}(s)) \supseteq L$. We show $\Theta^{\omega}(s,N_{o+c}(s)) \subseteq L$ as follows. Let $k = \arg\max_i\{N_o(s_{-i})+i : i \in [N_c(s_{-i})]\}$. By definition, $N_{o+c}(s) = \min\{|s|, N_o(s_{-k})+k\}$. Consider the case of $N_o(s_{-k})+k \leq |s|$. We have $0 \leq k \leq |s|$. Consider an arbitrary j such that $k \leq j \leq N_o(s_{-k})+k$. By the definition of Θ^{ω}, we have $\theta^{\omega}(s_{-j}) \in \Theta^{\omega}(s_{-k},N_o(s_{-k}))$. Then, since $N_c(s_{-k}) \geq k$ by Definition 6, $\theta^{\omega}(s_{-j}) \in L$. Consider an arbitrary j such that $j \in [k]$. Since $N_c(s_{-k}) \geq k$, by iteratively applying the part of Assumption 1 for the upper-bound of control delay for $s_{-k}, s_{-(k-1)}, \ldots, s_{-j}$ in this order, we have $N_c(s_{-j}) \geq j$. Then, since $\theta^{\omega}(s_{-j}) \in \Theta^{\omega}(s_{-j},N_o(s_{-j}))$, we have $\theta^{\omega}(s_{-j}) \in L$. Therefore, in this case, for all $j \in [N_{o+c}(s)]$, we have $\theta^{\omega}(s_{-j}) \in L$, which gives $\Theta^{\omega}(s,N_{o+c}(s)) \subseteq L$. Consider the case of $k > |s|$. Then, we have, for all $l = |s|,\ldots,k$, $\Theta^{\omega}(s_{-l},N_o(s_{-l})) = \Theta^{\omega}(\varepsilon,N_o(\varepsilon)) = \{\varepsilon\}$ and $N_c(s_{-l}) = N_c(\varepsilon) = k$. Then, applying the same reasoning as for the second occasion of the previous case, we have, for all $l = 0,\ldots,|s|$, $\theta^{\omega}(s_{-l}) \in L$. Thus, $\Theta^{\omega}(s,N_{o+c}(s)) = \{\theta^{\omega}(s_{-l}) : l = 0,\ldots,|s|\} \subseteq L$. □

As the control command implemented following $s \in \mathscr{L}(G)$ could have been issued when the supervisor made an observation of any $s'' \in \bar{s}$ with $|s''| \geq |s| - N_{o+c}(s)$, we can combine N_o and N_c in P as follows.

Proposition 1 *Suppose G, K, Σ_{uc}, ω, N_o, and N_c are given. Then, $\mathsf{P}(\mathscr{L}(G),K,\Sigma_{uc},\omega,0,N_{o+c})$ is well defined and*

$$\mathsf{P}(\mathscr{L}(G),K,\Sigma_{uc},\omega,N_c,N_o) \Leftrightarrow \mathsf{P}(\mathscr{L}(G),K,\Sigma_{uc},\omega,0,N_{o+c})$$

PROOF. Since N_{o+c} satisfies the part of Definition 3 and of Assumption 1 concerning observation delay (Lemma 1), $\mathsf{P}(\mathscr{L}(G),K,\Sigma_{uc},\omega,0,N_{o+c})$ is well defined.

By Lemma 2, for each S, (1) and (2) are equivalent to

$$\begin{aligned}&(\forall\sigma\in\Sigma)[s\sigma \in \mathscr{L}_r(G,S) \Leftrightarrow [s\sigma \in \mathscr{L}(G)\\ &\wedge s \in \mathscr{L}_r(G,S) \wedge (\forall t \in \Theta^{\omega}(s,N_{o+c}(s)))\ \sigma \in S(t)]]\end{aligned} \tag{5}$$

$$\begin{aligned}&(\forall\sigma\in\Sigma)[s\sigma \in \mathscr{L}_a(G,S) \Leftrightarrow [s\sigma \in \mathscr{L}(G)\\ &\wedge s \in \mathscr{L}_a(G,S) \wedge (\exists t \in \Theta^{\omega}(s,N_{o+c}(s)))\ \sigma \in S(t)]]\end{aligned} \tag{6}$$

Since these depend on N_o and N_c only through N_{o+c}, this completes the proof, by Definition 5. □

We now introduce controllability, delayed observability.

Definition 7 (Controllability) *We say K is controllable with respect to (w.r.t.) Σ_{uc} and $\mathscr{L}(G)$ if for all $s \in K$ and all $\sigma \in \Sigma$, $s\sigma \in \mathscr{L}(G) \setminus K$ implies $\sigma \in \Sigma_c$.*

To test whether the language generated by the supervised system is deterministic and is K, delay observability is introduced as follows:

Definition 8 (Delay observability) *K is delay observable w.r.t. ω, Σ_c, N_o, and $\mathscr{L}(G)$ if, for all $\sigma \in \Sigma_c$ and all $s\sigma, s' \in K$, $s'\sigma \in \mathscr{L}(G) \setminus K$ implies $\Theta^{\omega}(s,N_o(s)) \cap \Theta^{\omega}(s',N_o(s')) = \emptyset$.*

Definition 8 generalize the corresponding definition in (Shu & Lin 2017) by allowing bounds of delays depend on which string. When there are no observation delays, that is, $(\forall s)N_o(s) = 0$, delay observability reduces to observability.

Theorem 1 has a counterpart in (Shu & Lin 2017).

Theorem 1 *$\mathsf{P}(\mathscr{L}(G),K,\Sigma_{uc},\omega,N_c,N_o)$ is true iff $K \subseteq \mathscr{L}(G)$ is controllable w.r.t. Σ_{uc}, and $\mathscr{L}(G)$, and K is delay observable w.r.t. ω, Σ_c, N_{o+c}, and $\mathscr{L}(G)$.*

PROOF. By Proposition 1, it is sufficient to show this for the case of observation delay bounded by N_{o+c} and no control delay.

For sufficiency, suppose K is controllable and delay observable. Define the supervisor $S : \theta^{\omega}(\mathscr{L}(G)) \to 2^{\Sigma}$ as

$$(\forall t \in \theta^{\omega}(\mathscr{L}(G)))(\forall \sigma \in \Sigma)[\sigma \in S(t) \iff (\sigma \in \Sigma_{uc} \vee (\exists s \in K)[s\sigma \in K \wedge t \in \Theta^{\omega}(s, N_{o+c}(s))])] \quad (7)$$

Note that $S(t) \supseteq \Sigma_{uc}$ for all t as required.

Since $\mathscr{L}_r(G,S) \subseteq \mathscr{L}_a(G,S)$ by Definition 4, to prove $\mathscr{L}_r(G,S) = \mathscr{L}_a(G,S) = K$, we prove $K \subseteq \mathscr{L}_r(G,S)$ and $K \supseteq \mathscr{L}_a(G,S)$ as follows. We show $K \subseteq \mathscr{L}_r(G,S)$ by contradiction. Suppose $K \setminus \mathscr{L}_r(G,S) \neq \emptyset$. Let s be a shortest string in $K \setminus \mathscr{L}_r(G,S)$. By definition both K and $\mathscr{L}_r(G,S)$ are prefix closed and $\varepsilon \in K \cap \mathscr{L}_r(G,S)$. Hence we can write $s = s'\sigma$ for some $\sigma \in \Sigma$. Then, since s' is shorter than s, we have $s' \in K \cap \mathscr{L}_r(G,S)$. Then, by (1) and (7), $s = s'\sigma \in \mathscr{L}_r(G,S)$, which is a contradiction. Thus, $K \subseteq \mathscr{L}_r(G,S)$. We show $K \supseteq \mathscr{L}_a(G,S)$ by contradiction. Suppose $\mathscr{L}_a(G,S) \setminus K \neq \emptyset$. Let u be a shortest string in $\mathscr{L}_a(G,S) \setminus K$. By definition both K and $\mathscr{L}_a(G,S)$ are prefix closed and $\varepsilon \in K \cap \mathscr{L}_a(G,S)$. Hence we can write $u = u'\sigma$ for some $\sigma \in \Sigma$. Then, since u' is shorter than u, we have $u' \in K \cap \mathscr{L}_a(G,S)$. Then, by (2), $(\exists t \in \Theta^{\omega}(u', (N_{o+c})(u')))\sigma \in S(t)$. Moreover, since K is controllable, we have $\sigma \notin \Sigma_{uc}$. Then, by (7), $(\exists\ v\sigma \in K)t \in \Theta^{\omega}(v, N_{o+c}(v))$. Hence, $t \in \Theta^{\omega}(u', N_{o+c}(u')) \cap \Theta^{\omega}(v, N_{o+c}(v))$. Then, by Definition 8, K is not delay observable, which is a contradiction. Thus, $K \supseteq \mathscr{L}_a(G,S)$. Together, these establish $\mathscr{L}_r(G,S) = \mathscr{L}_a(G,S) = K$.

We prove necessity by proving the following contrapositive. If K is not controllable, since, for all S and all t, $S(t) \supseteq \Sigma_{uc}$, then $(\forall S)[\mathscr{L}_r(G,S) \supseteq K \Rightarrow \mathscr{L}_r(G,S) \supset K]$. Thus, $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega, 0, N_{o+c})$ is false. If K is controllable but not delay observable, by Definition 8,

$$(\exists \sigma \in \Sigma_c)(\exists s, s' \in K)[s\sigma \in K \wedge s'\sigma \in \mathscr{L}(G) \setminus K \wedge \Theta^{\omega}(s, N_{o+c}(s)) \cap \Theta^{\omega}(s', N_{o+c}(s')) \neq \emptyset]$$

We write $t \in \Theta^{\omega}(s, N_{o+c}(s)) \cap \Theta^{\omega}(s', N_{o+c}(s'))$. If there exists S such that $\mathscr{L}_r(G,S) = K$, by (1), $\sigma \in S(t)$. Then, since $\mathscr{L}_r(G,S) \subseteq \mathscr{L}_a(G,S)$ and $s' \in K$, $s' \in \mathscr{L}_a(G,S)$. Then, since $t \in \Theta^{\omega}(s', N_{o+c}(s'))$ and $s'\sigma \in \mathscr{L}(G)$, by (2), $s'\sigma \in \mathscr{L}_a(G,S)$, which implies $\mathscr{L}_r(G,S) \neq \mathscr{L}_a(G,S)$. Since S that satisfies $\mathscr{L}_r(G,S) = K$ is arbitrary, there does not exist S such that $\mathscr{L}_r(G,S) = \mathscr{L}_r(G,S) = K$, i.e., $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega, 0, N_{o+c})$ is false. □

Example 2 *Continue with Example 1. The observation of o_l, o_r, c_l, and c_r involves a delay of at least one event occurrence. Suppose all events are observable and control delay does not exist. We have that, for all s such that $\delta_D(q_0, s) = 0$, $\theta^{\omega^{all}}(s) \in \Theta^{\omega^{all}}(s) \cap \Theta^{\omega^{all}}(so_l)$. Then, since $\delta_D(q_0, so_l) = 2$, $m_l \in \Gamma_D(0)$, and $m_l \in \Gamma(2) \setminus \Gamma_D(2)$, by Definition 8, $\mathscr{L}(H)$ is not delay observable. Hence, by Theorem 1, given $N_o(so_l) \geq 1$, $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \emptyset, \omega^{all}, 0, N_o)$ is not true.*

3 Reduction from delay observability to observability

In this section, we investigate the relationship between on one hand deterministic networked control defined in Section 2.2 and on the other hand the traditional supervisory control specified by Ramadge & Wonham (1987) and Lin & Wonham (1988*b*).

3.1 Delay upper-bound is a function over the language

Definition 9 (Observability) *K is observable w.r.t. ω, Σ_{uc}, and $\mathscr{L}(G)$ (Lin & Wonham 1988*b*) if, for all $\sigma \in \Sigma_c$ and all $s, s' \in K$, $(s\sigma \in K \wedge s'\sigma \in \mathscr{L}(G) \setminus K) \Rightarrow \theta^{\omega}(s) \neq \theta^{\omega}(s')$.*

Definition 9 says that, for two strings s and s' in K followed by σ, if one of the resulting strings is in the desired language K and the other is not, then our limited observations θ^{ω} are able to distinguish between s and s'. Obviously, observability is a special case of delay observability that has $N_o \equiv 0$. However, we will form a new DES that is observable if and only if the original was delay observable. This task can be stated formally as follows:

Problem 1 *Consider G and $K \subseteq \mathscr{L}(G)$ with Σ_o and Σ_c, $\omega : \mathscr{L}(G) \to 2^{\Sigma_o}$, and delay bounds given by N_o and N_c yielding N_{o+c}. Find languages $\tilde{L}$ and $\tilde{K}$, together with their corresponding set of events $\tilde{\Sigma}$, set of controllable events $\tilde{\Sigma}_c$, and observation mapping $\tilde{\omega} : \tilde{L} \to 2^{\Sigma}$ such that K is both controllable and delay observable w.r.t. $\mathscr{L}(G)$, Σ_c, ω, and N_{o+c} iff $\tilde{K}$ is both controllable and observable w.r.t. $\tilde{L}$, $\tilde{\Sigma}_c$, and $\tilde{\omega}$.*

The first part of the solution to Problem 1 is to specify $\tilde{L}$ and $\tilde{K}$. The definition of $\tilde{\omega}$ will follow.

In the following definition, the symbol a following $s \in K$ precedes the last symbol σ if σ should be allowed while the current observed string is (the information mapping of) s, and the system is waiting for the next observation; recall that observation delay cannot reorder observations. Similarly b following $s \in K$ precedes σ if σ should be blocked.

Definition 10 *Given G and K, let $\tilde{\Sigma} = \Sigma \cup \{a, b\}$ with $a, b \notin \Sigma$, $\tilde{\Sigma}_c = \Sigma_c$, and $\tilde{\Sigma}_o = \Sigma_o$ so that a and b are uncontrollable and unobservable. Define*

$$K' = \{sa, sb : s \in K\} \quad (8)$$

$$K'' = \{sa\sigma : (\exists t)[st\sigma \in K \wedge \sigma \in \Sigma_c \wedge |t| \in [N_{o+c}(st)]]\} \quad (9)$$

$$L' = \{s\sigma \in \mathscr{L}(G) \setminus K : s \in K\} \quad (10)$$

$$L'' = \{sb\sigma : s \in K \wedge \sigma \in \Sigma_c \wedge (\exists t)(st \in K \wedge st\sigma \in \mathscr{L}(G) \setminus K \wedge |t| \in [N_{o+c}(st)])\} \quad (11)$$

Then, define desired language $\tilde{K} = K \cup K' \cup K''$ *and uncontrolled language* $\tilde{L} = \tilde{K} \cup L' \cup L''$.

Recall that, by the time $s \in K$ is observed, the string that has actually occurred can be some st' such that $|t'| \leq N_o(st')$ and the control command issued can still be in effect when some string $st't''$ satisfying $|t''| \leq N_c(st')$ occurs. Definition 6 states $|t't''| \leq N_{o+c}(st't'')$, and Theorem 1 states that the existence of a supervisor such that the supervised system language is K deterministically depends on the delay observability with delay bound above by N_{o+c}. Note that, later in Definition 11, a and b are set to be unobservable. Thus, for $st't'' \in K$ and all symbols σ such that $st't''\sigma \in \mathscr{L}(G)$, if $st't''\sigma \in K$, we include $sa\sigma$ into $\tilde{K}$ in (9) to denote that σ should be enabled when s is observed. If $st't''\sigma \in \mathscr{L}(G) \setminus K$, we include sb into $\tilde{K}$ and $sb\sigma$ into $\tilde{L} \setminus \tilde{K}$ in (11) to denote that σ should be disabled when s is observed.

The following definition says that the observation mapping of the augmented system is essentially that of the original, except after strings of the form $sa\sigma$ and $sb\sigma$, which cannot be followed by any new events – even in the augmented system – and so may allow arbitrary observation mapping follow them, without affecting the information mapping θ^{ω}. This flexibility will be used in Section 4.

Definition 11 *Given languages K,* $\tilde{L}$*, L′, and L″ in Definition 10 and* ω *over* $\mathscr{L}(G)$*, define* $\tilde{\omega}$ *over* $\tilde{L}$ *as:*

- *For all* $s \in K \cup L'$, $\tilde{\omega}(s) = \omega(s)$.
- *For all* $sa, sb \in \tilde{L} \setminus K$, $\tilde{\omega}(sa) = \tilde{\omega}(sb) = \omega(s)$.
- *For each* $sa\sigma$ *and each* $s'b\sigma'$ *in* $\tilde{L} \setminus K$ *with some* $\sigma, \sigma' \in \Sigma_c$*, use arbitrary* $\tilde{\omega}(sa\sigma), \tilde{\omega}(s'b\sigma') \subseteq \tilde{\Sigma}_o$.

This $\tilde{\omega}$ over $\tilde{L}$ is well defined since three cases in the definition are mutually exclusive and collectively exhaustive, and ω is well defined. To understand Definition 11, consider the supervisor who controls $\tilde{L}$ to obtain $\tilde{K}$. Since events a and b are artificial, all strings in $\tilde{L}$ that may occur (without control) are given by $K \cup L'$. For an arbitrary string s that possibly occur, the first line states that the new definition does not change the observation mapping of s. Then, since $K \cup L'$ is prefix closed, information mapping $\theta^{\tilde{\omega}}(s)$, which is how s appears to the supervisor, remains the same as $\theta^{\omega}(s)$. Since neither a nor b is defined in the original system, both are excluded from $\tilde{\omega}(s) = \omega(s)$ and are unobservable. Consequently, the decisions of not disabling events following a but disabling events following b are based on the observation of s, and $K \cup L'$ contains all strings whose observation mappings can affect the supervisor's decisions. Although the observation mappings of other strings have no influence on the observability, the way that we specify them (second and third lines) facilitates the application of the reduction in Section 4.

Theorem 2 establishes that $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega, N_c, N_o) \equiv \mathsf{P}(\tilde{L}, \tilde{K}, \tilde{\Sigma}_{uc}, \tilde{\omega}, 0, 0)$.

Theorem 2 *Given K,* $\mathscr{L}(G)$*,* Σ_c*,* ω*, and* N_{o+c}*, let* $\tilde{L}$*,* $\tilde{K}$*,* $\tilde{\Sigma}_c$*, and* $\tilde{\omega}$ *to be obtained by Definitions 10 and 11. Then,*

- *K is controllable w.r.t.* $\mathscr{L}(G)$ *and* Σ_c *iff* $\tilde{K}$ *is controllable w.r.t.* $\tilde{L}$ *and* $\tilde{\Sigma}_c$.
- *K is delay observable w.r.t.* $\mathscr{L}(G)$*,* ω*,* Σ_c*, and* N_{o+c} *iff* $\tilde{K}$ *is observable w.r.t.* $\tilde{L}$*,* $\tilde{\Sigma}_c$*, and* $\tilde{\omega}$.

PROOF. For the first part, note that $\tilde{K}$, L', and L'' are mutually exclusive. For all $sb\sigma \in L''$, by (10), $\sigma \in \tilde{\Sigma}_c = \Sigma_c$. Then, by Definition 7, the proof of this part follows (11).

The proof of the second part is as follows.

For sufficiency, if K is not delay observable w.r.t. ω, Σ_c, N_{o+c}, and $\mathscr{L}(G)$, by Definition 8,

$$(\exists \sigma \in \Sigma_c)(\exists s, s' \in K)[s\sigma \in K \wedge s'\sigma \in \mathscr{L}(G) \setminus K \\ \wedge \Theta^{\omega}(s, N_{o+c}(s)) \cap \Theta^{\omega}(s', N_{o+c}(s')) \neq \emptyset] \tag{12}$$

Then, by the definition of Θ^{ω}, there exist $i \in [N_{o+c}(s)]$ and $j \in [N_{o+c}(s')]$ such that $\theta^{\omega}(s_{-i}) = \theta^{\omega}(s'_{-j})$. We write $s = s_{-i}t$ and $s' = s'_{-j}t'$. Then, we have $s_{-i}, s'_{-j} \in K$, $|t| \in [N_{o+c}(s_{-i}t)]$, and $|t'| \in [N_{o+c}(s'_{-j}t')]$. Then, by Definition 10, $s_{-i}a, s_{-i}a\sigma, s'_{-j}b \in \tilde{K}$ and $s'_{-j}b\sigma \in \tilde{L} \setminus \tilde{K}$. Then, by Definition 11, $\theta^{\tilde{\omega}}(s_{-i}a) = \theta^{\omega}(s_{-i}) = \theta^{\omega}(s'_{-j}) = \theta^{\tilde{\omega}}(s'_{-j}b)$, which implies that $\tilde{K}$ is not observable w.r.t. $\tilde{L}$, $\tilde{\Sigma}_c$, and $\tilde{\omega}$.

For necessity, suppose $\tilde{K}$ is not observable w.r.t. $\tilde{L}$, $\tilde{\Sigma}_c$, and $\tilde{\omega}$. According to Definitions 10 and 11, we partition $\tilde{K} \times (\tilde{L} \setminus \tilde{K})$ into $A = K \times L'$, $B = (\tilde{K} \setminus K) \times L''$, $C = K \times L''$, and $D = (\tilde{K} \setminus K) \times L'$. Then, by Definition 9, there exists $\sigma \in \Sigma_c$ and $s, s' \in K$ such that at least one of (13), (14), (15), and (16) is true.

$$(s\sigma, s'\sigma) \in A \wedge \theta^{\tilde{\omega}}(s) = \theta^{\tilde{\omega}}(s') \tag{13}$$
$$(sa\sigma, s'b\sigma) \in B \wedge \theta^{\tilde{\omega}}(sa) = \theta^{\tilde{\omega}}(s'b) \tag{14}$$
$$(s\sigma, s'b\sigma) \in C \wedge \theta^{\tilde{\omega}}(s) = \theta^{\tilde{\omega}}(s'b) \tag{15}$$
$$(sa\sigma, s'\sigma) \in D \wedge \theta^{\tilde{\omega}}(sa) = \theta^{\tilde{\omega}}(s') \tag{16}$$

If (13) is true, since $\theta^{\omega}(s) = \theta^{\tilde{\omega}}(s)$, by the definition of Θ^{ω}, we have $\theta^{\omega}(s) \in \Theta^{\omega}(s, N_{o+c}(s)) \cap \Theta^{\omega}(s', N_{o+c}(s'))$, i.e., K is not delay observable. If (14) is true, let $w = \theta^{\omega}(s)$. Then, by Definition 11, $w = \theta^{\omega}(s) = \theta^{\tilde{\omega}}(sa) = \theta^{\tilde{\omega}}(s'b) = \theta^{\omega}(s')$. By Definition 10, in particular (9) and (10),

$$(\exists t, t')[(st\sigma, s't' \in K) \wedge s't'\sigma \in \mathscr{L}(G) \setminus K \wedge |t| \in [N_{o+c}(st)] \wedge \\ |t'| \in [N_{o+c}(s't')] \wedge w \in \Theta^{\omega}(st, N_{o+c}(st)) \cap \Theta^{\omega}(s't', N_{o+c}(s't'))]$$

i.e., (12) is true, and K is not delay observable. If (15) is true, then because $|\varepsilon| \in [N_{o+c}]$, (9) implies that there is an $sa\sigma \in \tilde{K} \setminus K$ such that $\theta^{\tilde{\omega}}(sa) = \theta^{\tilde{\omega}}(s'b)$. Then, the rest of proof follows the case of (14) with $w = \theta^{\omega}(s)$. If (16) is true,

by (10) and (11), we have $s'b\sigma \in L''$ such that $\theta^{\tilde{\omega}}(sa) = \theta^{\tilde{\omega}}(s'b)$. Then, the rest of proof also follows the case of (14) with $w = \theta^{\omega}(s)$. □

When $\mathsf{P}(\tilde{L}, \tilde{K}, \tilde{\Sigma}_{uc}, \tilde{\omega}, 0, 0)$ is true, there exists a supervisor $\tilde{S} : \theta^{\tilde{\omega}}(L) \to 2^{\tilde{\Sigma}}$ under which $\tilde{K}$ can be exactly obtained. Theorem 2 says that this implies $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega, N_c, N_o)$ is true, and the supervisor S for controlling $\mathscr{L}(G)$ to obtain K can be implement using $\tilde{S}$ as, for all $s \in K$, $S(\theta^{\omega}(s)) = \tilde{S}(\theta^{\tilde{\omega}}(s)) \cap \Sigma$.

3.2 Delay upper-bound is a function over the state space

In the rest of this paper, we assume that G is a finite-state automaton and $K = \mathscr{L}(H)$, in which $H = (Q_D, \Sigma, \delta_D, \Gamma_D, q_0)$ is a subautomaton of G, denoted by $H \sqsubseteq G$, and the upper bound of observation-plus-control delay is state-dependent, rather than depending on the full string so far.

The following definition introduces a new parameterization $N^{\uparrow}$ of observation and control delay. Each control decision is in response to an observation. $N^{\uparrow}$ expresses the delay bound as a function of the state the system is in at the end of the substring that causes the control decisions to be issued. In that sense, it considers delay looking forward. In contrast, N_{o+c} measures the same delay bound as a function of the longest string following which the control decision could be in effect. In that sense, it considers delay looking backward. By parameterizing the delay upper bound as $N^{\uparrow}$, we can construct a model such that if the supervisor uses this model and ignores the possibility of delay, it will make the correct decisions to implement the desired language deterministically in the true system with nondeterministic delay.

Definition 12 (State-based upper bound of observation-plus-control delay) *Given $G = (Q, \Sigma, \delta, \Gamma, q_0)$ that is finite state, $K = \mathscr{L}(H)$ with $H \sqsubseteq G$, and N_{o+c}, let $N^{\uparrow} : Q \to \mathbb{N}$ denote the function such that, for all $q \in Q$,*

$$\begin{aligned} N^{\uparrow}(q) = \max\{|t| : (\exists s \in \mathscr{L}(G))[\delta(q_0, s) = q \\ \wedge st \in \mathscr{L}(G) \wedge |t| \in [N_{o+c}(st)]]\} \end{aligned} \tag{17}$$

Then, we call $N^{\uparrow}$ the state-based upper bound of observation-plus-control delay if its corresponding N_{o+c} satisfies

$$\begin{aligned} &(\forall q \in Q)(\forall s \in \mathscr{L}(G))[\delta(q_0, s) = q \implies \\ &(\forall t)([st \in \mathscr{L}(G) \wedge |t| \in [N^{\uparrow}(q)]] \implies |t| \in [N_{o+c}(st)])] \end{aligned} \tag{18}$$

If $N^{\uparrow}$ is a state-based upper-bound of observation-plus-control delay, Definition 12 says that, given observation mapping ω, for all $s \in \mathscr{L}(G)$, a control command issued after the supervisor observes $\theta^{\omega}(s)$ could be implemented following st if st can occur and $|t| \le N^{\uparrow}(\delta(q_0, s))$, but is not implemented following st otherwise. Lemma 3 presents bounds on $N^{\uparrow}$.

Lemma 3 *Let $T = \max\{|t| : \delta(\delta(q, \sigma), t)$ is defined$\}$. Then for all $q \in Q$ and all $\sigma \in \Gamma(q)$,*

$$\begin{cases} N^{\uparrow}(\delta(q, \sigma)) \ge N^{\uparrow}(q) - 1 & \text{if } T > N^{\uparrow}(q) - 1 \\ N^{\uparrow}(\delta(q, \sigma)) = T & \text{otherwise} \end{cases} \tag{19}$$

PROOF. Consider an arbitrary $q \in Q$ and an arbitrary $\sigma \in \Gamma(q)$. Let $s \in \mathscr{L}(G)$ such that $\delta(q_0, s) = q$ be arbitrary. Consider $N^{\uparrow}(q) = 0$. Since by definition $N^{\uparrow}(\delta(q, \sigma)) \ge 0$, we have $N^{\uparrow}(\delta(q, \sigma)) \ge N^{\uparrow}(q) - 1$. Consider $N^{\uparrow}(q) > 0$. If there exists string t such that $s\sigma t \in \mathscr{L}(G)$ and $|\sigma t| = N^{\uparrow}(q)$, then by (18), we have $|\sigma t| \in [N_{o+c}(s\sigma t)]$, which implies $|t| \in [N_{o+c}(s\sigma t)]$. Then, by (17), we have $N^{\uparrow}(\delta(q_0, s\sigma)) \ge |t|$, i.e., $N^{\uparrow}(\delta(q, \sigma)) \ge N^{\uparrow}(q) - 1$. Otherwise, applying (18) and then (17), we have $N^{\uparrow}(\delta(q, \sigma)) = \max\{t : \delta(\delta(q, \sigma), t)$ is defined$\}$. □

Given $N^{\uparrow}$, Definitions 13 and 14 specify the unique N_{o+c} that corresponds to it.

Definition 13 *Given $N^{\uparrow}$, define the following subsets of $\mathscr{L}(G)$:*

$$\pi_0 = \{s \in \mathscr{L}(G) : |s| \le N^{\uparrow}(q_0)\} \tag{20}$$

and, for each $q \in Q$ and each $\sigma \in \Gamma(q)$,

$$\begin{aligned} \pi(q, \sigma) = &\{s\sigma t \in \mathscr{L}(G) : \delta(q_0, s) = q \\ &\wedge N^{\uparrow}(q) - 1 < |t| \le N^{\uparrow}(\delta(q, \sigma))\} \end{aligned} \tag{21}$$

Lemma 4 *The set of nonempty subsets given in Definition 13 forms a partition Π of $\mathscr{L}(G)$.*

PROOF. For an $s \in \mathscr{L}(G)$, write $s = \sigma_1 \cdots \sigma_{|s|}$ for some $\sigma_1, \ldots, \sigma_{|s|} \in \Sigma$. Also write $s = s_{-(|s|-i)}t_{i+1}$ for $i \in [|s| - 1]$ such that the subscript of suffix t_{i+1} denotes t_{i+1} starting at $i + 1^{st}$ event occurrence in s.

Consider an arbitrary $s \in \mathscr{L}(G)$. If $N^{\uparrow}(q_0) \ge |s|$, then, by (20) in Definition 13, $s \in \pi_0$. To see that this is the only subset containing s, iteratively applying Lemma 3 for each $i = 1, \ldots, |s| - 1$ in this order gives $N^{\uparrow}(\delta(q_0, s_{-(|s|-i)})) - 1 \ge |s| - (i + 1) = |t_{i+2}|$ regardless of which case in the lemma applied, which implies, by (21), that $s \notin \pi(\delta(q_0, s_{-(|s|-i)}), \sigma_{i+1})$, as required. Otherwise, let $U = \{j \in [|s| - 1] : N^{\uparrow}(\delta(q_0, s_{-(|s|-j)}\sigma_{j+1})) \ge |s| - (j + 1)\}$. We have $|s| - 1 \in U$, which implies $U \neq \emptyset$. Let $k = \min U$, and we have $N^{\uparrow}(\delta(q_0, s_{-(|s|-k)}\sigma_{k+1})) \ge |s| - (k + 1) = |t_{(k+1)+1}|$ and $N^{\uparrow}(\delta(q_0, s_{-(|s|-(k-1))}\sigma_{(k-1)+1})) < |s| -$

$((k-1)+1) = |t_{(k+1)+1}| + 1$. Then, by (21), we have $s \in \pi(\delta(q_0, s_{-(|s|-k)}), \sigma_{k+1})$. To see that this is the only subset containing s, note that applying Lemma 3 iteratively from $i = k+1$ to $|s|-1$ gives $N^{\uparrow}(\delta(q_0, s_{-(|s|-i)})) - 1 \geq |s| - (i+1) = |t_{i+2}|$ regardless of which case in the lemma applies, which implies $s \notin \pi(\delta(q_0, s_{-(|s|-i)}), \sigma_{i+1})$ by (21). Thus, no element in Π other than $\pi(\delta(q_0, s_{-(|s|-k)}), \sigma_{k+1})$ contains s. Hence, there is a unique element in Π containing s. The lemma follows since $s \in \mathscr{L}(G)$ is arbitrary. □

Definition 14 *Given $N^{\uparrow}$ and the partition Π of $\mathscr{L}(G)$, define N_{o+c} as follows.*

- *For each $s \in \pi_0$, set $N_{o+c}(s) = |s|$.*
- *For each $q \in Q$, each $\sigma \in \Gamma(q)$, and each (s,t) such that $s\sigma t \in \pi(q,\sigma)$, $\delta(q_0,s) = q$, and $N^{\uparrow}(q) - 1 < |t| \leq N^{\uparrow}(\delta(q,\sigma))$, set $N_{o+c}(s\sigma t) = |t|$.*

Proposition 2 *Given a state-based upper-bound of observation-plus-control delay $N^{\uparrow}$, there is a unique N_{o+c} from which $N^{\uparrow}$ can be computed using Definition 12, and that N_{o+c} is computed by Definition 14.*

PROOF. For an $s \in \mathscr{L}(G)$, write $s = \sigma_1 \cdots \sigma_{|s|}$ for some $\sigma_1, \ldots, \sigma_{|s|} \in \Sigma$. Also write $s = s_{-(|s|-i)} t_{i+1}$ for $i \in [|s|-1]$ such that the subscript of suffix t_{i+1} denotes t_{i+1} starting at $i+1^{st}$ event occurrence in s.

Consider an $N^{\uparrow}$ satisfying (19) in Lemma 3. By Lemma 4, Π defined by Definition 13 is a partition of $\mathscr{L}(G)$. Then, by definition, N_{o+c} is a function that has domain $\mathscr{L}(G)$. We also need to show that N_{o+c} satisfies the two conditions in Lemma 1 as follows. The first condition is satisfied since, by definition, $N_{o+c}(s) \leq |s|$ for all $s \in \mathscr{L}(G)$. The proof that the second condition is satisfied will be by contradiction. Consider an arbitrary $s \in \mathscr{L}(G)$ with $|s| \geq 1$. Let $k = N_{o+c}(s)$ and $l = N_{o+c}(s_{-1})$. It is suffice to show $k \leq l+1$. Suppose instead $k > l+1$. If $k = |s|$, then $s \in \pi_0$. Since $k > l+1$ and $|s| = |s_{-1}| + 1$, $l < |s_{-1}|$. Thus, $s_{-1} \in \pi(\delta(q_0, (s_{-1})_{-(l+1)}), \sigma_{(|s|-1)-l})$. We write $s_{-1} = (s_{-1})_{-l} t_{(|s|-1)-l+1}$ and have $|t_{(|s|-1)-l+1}| = l$. Then, by Definition 14, we have $N^{\uparrow}(\delta(q_0, (s_{-1})_{-(l+1)})) - 1 < |t_{(|s|-1)-l+1}| = l$, which gives $N^{\uparrow}(\delta(q_0, (s_{-1})_{-(l+1)})) < l+1$. Then, by repeatedly applying Lemma 3, $N^{\uparrow}(\delta(q_0, \varepsilon)) = N^{\uparrow}(q_0) < l + 1 + (|s| - (1 + (l+1))) = |s| - 1$. This contradicts $N^{\uparrow}(q_0) \geq |s|$ required by $s \in \pi_0$. Otherwise, we have $s \in \pi(\delta(q_0, s_{-(k+1)}), \sigma_{|s|-k})$ and $s_{-1} \in \pi(\delta(q_0, (s_{-1})_{-(l+1)}), \sigma_{(|s|-1)-l})$. Then, by the same argument for the previous case, we have $N^{\uparrow}(\delta(q_0, (s_{-1})_{-(l+1)})) < l+1$. Then, by repeatedly applying Lemma 3, $N^{\uparrow}(\delta(q_0, s_{-k})) < l + 1 + (k - (1 + (l+1))) = k - 1$. This contradicts $N^{\uparrow}(\delta(q_0, s_{-k})) \geq k$ required by $s \in \pi(\delta(q_0, s_{-(k+1)}), \sigma_{|s|-k})$. Therefore, $k \leq l+1$, whence $N_{o+c}(s) \leq N_{o+c}(s_{-1}) + 1$. Since $s \in \mathscr{L}(G)$ with $|s| \geq 1$ is arbitrary, the second condition of in Lemma 1 holds. Overall, N_{o+c} is an upper-bound of observation-plus-control delay.

Using this N_{o+c}, the corresponding $N^{\uparrow}$ can be recovered using (17) subject to (18). So far, we have established that N_{o+c} computed by Definition 14 from which $N^{\uparrow}$ computed using Definition 12 is an upper-bound of observation-plus-control delay.

The uniqueness of N_{o+c} follows from two facts: By (17), for an arbitrary $s \in \mathscr{L}(G)$, increasing $N_{o+c}(s)$ by $i > 0$ with $i + N_{o+c}(s) \leq |s|$ strictly increases $N^{\uparrow}(\delta(q_0, s_{-(N_{o+c}(s)+i)}))$. While keeping $N^{\uparrow}$ fixed, decreasing $N_{o+c}(s)$ for any $s \in \mathscr{L}(G)$ causes $N^{\uparrow}(\delta(q_0, s_{-N_{o+c}(s)}))$ violating (18). □

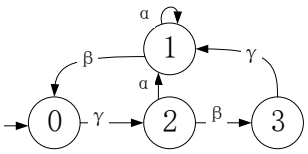


Fig. 2. Automata G in Example 3.

Example 3 *Consider system G Shown in Fig. 2. Suppose $N^{\uparrow}(0) = 1$, $N^{\uparrow}(1) = N^{\uparrow}(3) = 2$, and $N^{\uparrow}(2) = 0$. Then, by Definition 14, $\pi_0 = \{\varepsilon, \gamma\}$, $\pi(2,\beta) = \{s\beta t : \delta(q_0,s) = 2 \wedge t \in \overline{\{\gamma\alpha, \gamma\beta\}}\}$, $\pi(2,\alpha) = \{s\alpha t : \delta(q_0,s) = 2 \wedge t \in \overline{\{\alpha\alpha, \alpha\beta, \beta\gamma\}}\}$, $\pi(3,\gamma) = \{s\gamma t : \delta(q_0,s) = 3 \wedge t \in \{\alpha\alpha, \alpha\beta, \beta\gamma\}\}$, and $\pi(1,\alpha) = \{s\alpha t : \delta(q_0,s) = 1 \wedge t \in \{\alpha\alpha, \alpha\beta, \beta\gamma\}\}$. Note that, since $\pi(0,\gamma) = \pi(1,\beta) = \emptyset$ and partition Π only includes nonempty set, we have $\pi(0,\gamma), \pi(1,\beta) \notin \Pi$. N_{o+c} is specified in the following way. Since $\varepsilon, \gamma \in \pi_0$, $N_{o+c}(\varepsilon) = 0$, by definition, $N_{o+c}(\gamma) = 1$. Consider $s\alpha\beta \in \pi(2,\alpha)$ for instance, we have $N^{\uparrow}(2) - 1 = -1 < |\beta| \leq N^{\uparrow}(1) = 2$, which gives $N_{o+c}(s\alpha\beta) = |\beta| = 1$.* □

From now on, we use $N^{\uparrow}$ exclusively for denoting state-based upper-bound of observation-plus-control delay. Proposition 2 says that the N_{o+c} by Definition 14 is equivalent to the given $N^{\uparrow}$. Therefore, all results derived for N_{o+c} hold for $N^{\uparrow}$.

The supervisor that implements $\tilde{H}$ when run on $\tilde{G}$, if one exists, can be applied to the real system G. This yields the desired language $\mathscr{L}(H)$.

Due to delays in observation and in implementing decisions, the supervisor cannot apply the original models G and H directly to perform the control. Instead, given automaton $H = (Q_D, \Sigma, \delta_D, \Gamma_D, q_0) \sqsubseteq G$ that models the desired system and $N^{\uparrow}$, we use Algorithm 1 to compute $\tilde{G}$ and $\tilde{H} \sqsubseteq \tilde{G}$ such that $\tilde{L} = \mathscr{L}(\tilde{G})$ and $\tilde{K} = \mathscr{L}(\tilde{H})$, for $\tilde{L}$ and $\tilde{K}$ given by Definition 10. The supervisor that implements $\tilde{H}$ when run on $\tilde{G}$, if one exists, can be applied to the real system G. This yields the desired language $\mathscr{L}(H)$.

Due to observation delay, a control decision at time t is based on observation of a string at a previous time $t' \leq t$.

Algorithm 1 DELAY-OBS-TO-OBS

Input: Automata G and $H \sqsubseteq G$ that generate respective $\mathscr{L}(G)$ and K, state-dependent upper bound $N^{\uparrow}$ on observation-plus-control delay;

Output: Automata $\tilde{G} := (\tilde{Q}, \tilde{\Sigma}, \tilde{\delta}, \tilde{\Gamma}, q_0)$ and $\tilde{H} := (\tilde{Q}_D, \tilde{\Sigma}, \tilde{\delta}_D, \tilde{\Gamma}_D, q_0)$;

1: $\tilde{G} \leftarrow H$ and $\tilde{H} \leftarrow H$;
2: $\tilde{\Sigma} \leftarrow \Sigma \cup \{a,b\}$ with $a,b \notin \Sigma$;
3: $\tilde{Q}_D \leftarrow Q_D \cup \hat{A}(Q_D) \cup \hat{B}(Q_D) \cup \{d\}$ and $\tilde{Q} \leftarrow \tilde{Q}_D$, where $\hat{A}$ and $\hat{B}$ are both bijective, and Q_D, $\hat{A}(Q_D)$, $\hat{B}(Q_D)$, and $\{d\}$ are mutually exclusive;
4: **for each** $q \in Q_D$ **do**
5: $\quad \tilde{\delta}(q,a) \leftarrow \hat{A}(q)$ and $\tilde{\delta}_D(q,a) \leftarrow \hat{A}(q)$;
6: $\quad \tilde{\delta}(q,b) \leftarrow \hat{B}(q)$ and $\tilde{\delta}_D(q,b) \leftarrow \hat{B}(q)$;
7: $\quad$ For each $\sigma \in \Gamma(q) \setminus \Gamma_D(q)$, $\tilde{\delta}(q,\sigma) \leftarrow d$;
8: $\quad$ Compute $(\text{EN}(q), \text{DIS}(q)) \leftarrow \text{REACH}(q)$;
9: $\quad$ **for each** $\sigma \in \text{EN}(q) \cap \Sigma_c$ **do**
10: $\quad\quad \tilde{\delta}_D(\hat{A}(q),\sigma) \leftarrow d$ and $\tilde{\delta}(\hat{A}(q),\sigma) \leftarrow d$;
11: $\quad$ **end for**
12: $\quad$ **for each** $\sigma \in \text{DIS}(q) \cap \Sigma_c$ **do**
13: $\quad\quad \tilde{\delta}(\hat{B}(q),\sigma) \leftarrow d$;
14: $\quad$ **end for**
15: **end for**
16: **return** $\tilde{G} := (\tilde{Q}, \tilde{\Sigma}, \tilde{\delta}, \tilde{\Gamma}, q_0)$ and $\tilde{H} := (\tilde{Q}_D, \tilde{\Sigma}, \tilde{\delta}_D, \tilde{\Gamma}_D, q_0)$;

17: **function** REACH(q)
18: $\quad A \leftarrow \Gamma_D(q)$, $B \leftarrow \Gamma(q) \setminus \Gamma_D(q)$, $C \leftarrow \{q\}$, $D \leftarrow Q_D \setminus \{q\}$, and $k \leftarrow 1$;
19: $\quad$ **while** $k \leq N^{\uparrow}(q) \wedge D \neq \emptyset$ **do**
20: $\quad\quad C \leftarrow \{\delta_D(x,\sigma) \in D : x \in C \wedge \sigma \in \Gamma_D(x)\}$;
21: $\quad\quad A \leftarrow A \cup (\cup_{x \in C} \Gamma_D(x))$;
22: $\quad\quad B \leftarrow B \cup [\cup_{x \in C} (\Gamma(x) \setminus \Gamma_D(x))]$;
23: $\quad\quad D \leftarrow D \setminus C$ and $k \leftarrow k+1$;
24: $\quad$ **end while**
25: $\quad$ **return** (A,B);
26: **end function**

The key function REACH relates past states to present control decisions. The argument q of REACH is the state the system is in at time t'. The control decisions at t are to enable the set of events in the first return value of REACH(q), and disable those in the second return value. The algorithm incorporates auxiliary states $\hat{A}(q)$ and $\hat{B}(q)$ into $\tilde{G}$ and $\tilde{H}$ by setting $\tilde{\delta}(q,a) \leftarrow \hat{A}(q)$ and $\tilde{\delta}_D(q,a) \leftarrow \hat{A}(q)$ on line 5, and $\tilde{\delta}(q,b) \leftarrow \hat{B}(q)$ and $\tilde{\delta}_D(q,b) \leftarrow \hat{B}(q)$ on line 6. Then, through executing the for-each loops on lines 9 and 12, it calculates the set of controllable events that the supervisor should enable at $\hat{A}(q)$, and those that the supervisor should disable at $\hat{B}(q)$, respectively. Recall that both a and b are auxiliary events that are uncontrollable and unobservable. So the control required following them in $\hat{A}(q)$ and in $\hat{B}(q)$, respectively, must depend only on the string of events preceding them in q. A supervisor controlling $\mathscr{L}(G)$ by which $\mathscr{L}(H)$ can be obtained deterministically exists if and only if $\mathscr{L}(\tilde{H})$ is controllable and observable with respect to $\mathscr{L}(\tilde{G})$, $\tilde{\Sigma}_c$, and $\tilde{\omega}$ specified by Definition 11.

The remainder of this subsection proves that Algorithm 1 and Definition 11 solve Problem 1 with upper-bounds of observation-plus-control delay being state-dependent.

Theorem 3 *Given inputs G, K modeled by $H \sqsubseteq G$, Σ_{uc}, ω over $\mathscr{L}(G)$, and N_o and N_c that have corresponding $N^{\uparrow}$, let $\tilde{G}$ and $\tilde{H}$ be the outputs of Algorithm 1, $\tilde{\Sigma}_c = \Sigma_c$, and $\tilde{\omega}$ over $\mathscr{L}(\tilde{G})$ specified by Definition 11. Then,*

- *K is controllable w.r.t. $\mathscr{L}(G)$ and Σ_c iff $\mathscr{L}(\tilde{H})$ is controllable w.r.t. $\mathscr{L}(\tilde{G})$ and $\tilde{\Sigma}_c$.*
- *K is delay observable w.r.t. $\mathscr{L}(G)$, ω, Σ_c, N_o, and N_c iff $\mathscr{L}(\tilde{H})$ is observable w.r.t. $\mathscr{L}(\tilde{G})$, $\tilde{\Sigma}_c$, and $\tilde{\omega}$.*

PROOF. To apply Theorem 2, we prove $\mathscr{L}(\tilde{G}) = \tilde{L}$ and $\mathscr{L}(\tilde{H}) = K'$, where $\tilde{L}$ and K' are given by Definition 10, as follows. Given an arbitrary $q \in Q_D$, define

$$
\begin{aligned}
K(q) &= \{s : s \in K \wedge \delta_D(q_0,s) = q\} \\
K'(q) &= \{sa, sb : s \in K \wedge \delta_D(q_0,s) = q\} \\
K''(q) &= \{sa\sigma : s \in K \wedge \delta_D(q_0,s) = q \wedge \sigma \in \Sigma_c \\
&\qquad \wedge (\exists t)(st\sigma \in K \wedge |t| \in [N_{o+c}(st)])\} \\
L''(q) &= \{sb\sigma : s \in K \wedge \delta_D(q_0,s) = q \wedge \sigma \in \Sigma_c \\
&\qquad \wedge (\exists t)[st \in K \wedge st\sigma \in \mathscr{L}(G) \setminus K \wedge |t| \in [N_{o+c}(st)]]\} \\
L'(q) &= \{s\sigma : s \in K \wedge \sigma \in \Sigma \wedge \delta_D(q_0,s) = q \wedge \\
&\qquad s\sigma \in \mathscr{L}(G) \setminus K\}
\end{aligned}
$$

Then, by Definition 10, we have

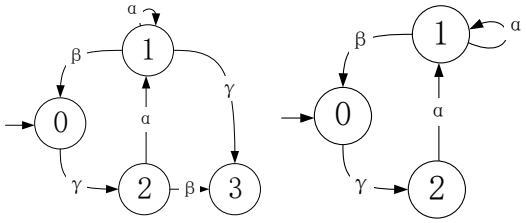
$$\tilde{L} = \bigcup_{q \in Q_D} (K(q) \cup K'(q) \cup K''(q) \cup L''(q) \cup L'(q)) \tag{22}$$

$$\tilde{K} = \bigcup_{q \in Q_D} (K(q) \cup K'(q) \cup K''(q)) \tag{23}$$

Since $K = \mathscr{L}(H)$, for each $q \in Q_D$, $K(q)$ is included into both $\mathscr{L}(\tilde{H})$ and $\mathscr{L}(\tilde{G})$ on line 1. Then, during each iteration for each $q \in Q_D$ on line 4, $\mathscr{L}(\tilde{H})$ and $\mathscr{L}(\tilde{G})$ are extended through adding transitions into $\tilde{H}$ and $\tilde{G}$, which is detailed as follows. $K'(q)$ is included into both $\mathscr{L}(\tilde{H})$ and $\mathscr{L}(\tilde{G})$ on lines 5 and 6. $L'(q)$ is included into $\mathscr{L}(\tilde{H})$ on line 7. Consider $K''(q)$ and $L''(q)$. In the iteration for q, EN(q) computed by REACH(q) is the union of the active event sets $\Gamma_D(p)$ with states p being reachable from q within $N^{\uparrow}(q)$ event occurrences in H, and DIS(q) is the union of the sets $\Gamma(p) \setminus \Gamma_D(p)$ for states that are reachable from q within $N^{\uparrow}(q)$ event occurrences. According to Definitions 12 and the definitions of $\tilde{K}$ and L'' in Definition 10, $K''(q)$ is included into both $\tilde{L}$ and $\tilde{K}$ by executing the for-each loop on line 9, and $L''(q)$ are included into $\tilde{L}$ by executing the for-each loop on line 12.

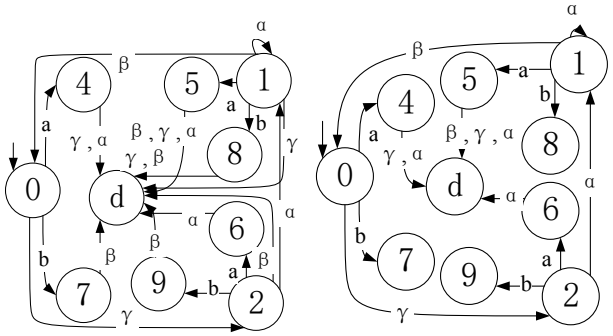
Since $q \in Q_D$ is arbitrary and the algorithm never delete transitions from intermediate $\tilde{G}$ and $\tilde{H}$ and no other strings are

added to the language by these operations, by (22) and (23), we have $L = \mathscr{L}(\tilde{G})$ and $\tilde{K} = \mathscr{L}(\tilde{H})$. Thus, $\tilde{\omega}$ over $\mathscr{L}(\tilde{G})$ by Definition 11 is well defined. Moreover, by Proposition 2, Definition 14 gives N_{o+c} that is equivalent to $N^{\uparrow}$. Then, the proof is completed by applying Theorem 2 with N_{o+c}. □



(a) System G (b) Desired system H

Fig. 3. Automata G and H in Example 4.



(a) Automaton $\tilde{G}$ (b) Automaton $\tilde{H}$

Fig. 4. Automata $\tilde{G}$ and $\tilde{H}$ in Example 4.

Example 4 *Consider system G and desired system H with* $\Sigma = \Sigma_c = \Sigma_o = \{\alpha, \beta, \gamma\}$ *shown in Fig. 3. Let* $\omega \equiv \omega^{all}$ *be the observation mapping. Assume* $N^{\uparrow}(0) = 1$, $N^{\uparrow}(1) = 2$, *and* $N^{\uparrow}(2) = 0$. *We can check that this* $N^{\uparrow}$ *satisfies conditions in Lemma 3, by Proposition 2, the corresponding* N_{o+c} *is given by Definition 14. We construct* $\tilde{G}$ *and* $\tilde{H}$ *from G and H using Algorithm 1 as shown in Fig. 4. It can be verified that* $\mathscr{L}(\tilde{H})$ *is not observable w.r.t.* $\tilde{\Sigma}_c$, $\tilde{\omega}$ *and* $\mathscr{L}(\tilde{G})$*: For each* $s \in \mathscr{L}(\tilde{H})$ *such that* $\tilde{\delta}_D(q_0, s) = 1$, *we have* $sa, sb, sa\beta \in \mathscr{L}(\tilde{H})$, $sb\beta \in \mathscr{L}(\tilde{G}) \setminus \mathscr{L}(\tilde{H})$, *and* $\theta^{\tilde{\omega}}(sa) = \theta^{\tilde{\omega}}(sb)$. *By the definition of observability,* $\mathscr{L}(\tilde{H})$ *is not observable. By Theorem 3,* $\mathscr{L}(H)$ *is not delay observable w.r.t.* ω, Σ_c, N *and* $\mathscr{L}(G)$.

We have Lemma 5 about the complexity of Algorithm 1.

Lemma 5 *The time complexity of Algorithm 1 is* $\mathscr{O}(|\Sigma||Q|^2)$.

PROOF. The number of iterations of the for-each loop on line 4 is no larger than $|Q|$, and, executing function REACH has complexity of order $|\Sigma||Q|$. The time complexity of Algorithm 1 w.r.t. $|Q|$ and $|\Sigma|$ is the product of these two, which gives $\mathscr{O}(|\Sigma||Q|^2)$. □

It is worth mention that the output systems $\tilde{G}$ and $\tilde{H}$ have sizes that are linear in the size of G in terms of both the state space and event set. When we apply the reduction to solve other observation problems using the same system model G, if the complexity of solving the corresponding problem without delay is polynomial and has an order not lower than that of the reduction ($\mathscr{O}(|\Sigma||Q|^2)$), the former determines the order of the complexity of the whole computation. In this case, solving the original problem and solving the problem without delay have the same order of complexity.

Next, we present how to use the reduction techniques to verify delay observability under transition-based dynamic observation that has state-dependent delays.

Definition 15 *Given G,* $H \sqsubseteq G$, *and* Σ_c, *we say that two states* $q, q' \in Q_D$ *have control conflicts if there exists a* $\sigma \in \Sigma_c$ *such that* $\sigma \in \Gamma(q) \setminus \Gamma_D(q)$ *but* $\sigma \in \Gamma_D(q')$. *Let* $\Lambda(G,H) = \{\{q, q'\} \in Q_D \times Q_D : (\exists \sigma \in \Sigma_c)[\sigma \in \Gamma_D(q) \wedge \sigma \in (\Gamma(q') \setminus \Gamma_D(q'))]\}$ *be the set of all pair sets, each of which consists of two states of H that have a control conflict.*

To proceed further, we define observation mapping over a partition of a language.

Definition 16 *Given language L with observable event set* Σ_o *and a partition* Π *of L, we say that an information mapping* $\omega : L \to 2^{\Sigma_o}$ *is also over* Π *if, for all* $\pi \in \Pi$ *and all* $s, s' \in \pi$, $\omega(s) = \omega(s')$. *With a slight abuse of notation, in this case, we define* $\omega : \Pi \to 2^{\Sigma_o}$ *as, for all* $\pi \in \Pi$ *and all* $s \in \pi$, $\omega(\pi) = \omega(s)$.

Given finite-state automaton $H = (Q_D, \Sigma, \delta_D, \Gamma_D, q_0)$, a partition Π of $\mathscr{L}(H)$ is called transition-based if each of its elements has the form $\pi(q) = \{s \in \mathscr{L}(H) : \delta_D(q_0, s) = q\}$. With a slight abuse of notation, in this case $q \in Q_D$ is used to represent $\pi(q)$ with the corresponding observation mapping given by $\omega : Q_D \to 2^{\Sigma_o}$.

Given G, $H \sqsubseteq G$, Σ_c, ω, $N^{\uparrow}$, $\tilde{G}$ and $\tilde{H}$ being the outputs of Algorithm 1, and $\tilde{\omega}$ over $\mathscr{L}(\tilde{G})$. We have $\tilde{H} \sqsubseteq \tilde{G}$. In this case, Wang et al. (2007) show that satisfying observability is equivalent to the supervisor always being able to distinguish two states in each pair set that have control conflicts. We repeat this result as Lemma 6.

Lemma 6 *Given* $\tilde{G}$ *and* $\tilde{H} \sqsubseteq \tilde{G}$, $\tilde{\omega}$, $\mathscr{L}(\tilde{H})$ *is observable iff, for all* $s, s' \in \mathscr{L}(\tilde{H})$, $\theta^{\tilde{\omega}}(s) = \theta^{\tilde{\omega}}(s') \Rightarrow \{\tilde{\delta}_D(q_0, s), \tilde{\delta}_D(q_0, s')\} \notin \Lambda(\tilde{G}, \tilde{H})$.

PROOF. The proof follows from $\tilde{H} \sqsubseteq \tilde{G}$ and Definition 9.

The verifier in (Wang et al. 2007, 2011) is for a transition-based observation mapping $\tilde{\omega} : \tilde{Q}_D \to 2^{\tilde{\Sigma}_o}$. Given ω over Q_D, by Definition 11, we specify $\tilde{\omega}$ as: for each $q \in Q_D$, $\tilde{\omega}(q) =$

$\tilde{\omega}(\hat{A}(q)) = \tilde{\omega}(\hat{B}(q)) \leftarrow \omega(q)$ and $\tilde{\omega}(d) \leftarrow \emptyset$. Then, applying Theorem 3, delay observability of $\mathscr{L}(H)$ is equivalent to observability of $\mathscr{L}(\tilde{H})$. As shown by Lemma 6, verifying observability can be reduced to distinguishing states.

4 Minimal sensor activation policy

To illustrate the applicability of this reduction, we adapts existing algorithms to calculate a minimal sensor activation policy (SAP) for deterministic networked control purposes. Let us consider the case in which not all sensors are active at all times. This may be done to reduce power consumption in a battery powered sensor. Alternatively, it may model the case that one physical sensor is shared between multiple sensing tasks, such as a camera that can observe different light wavelengths by applying different filters, but only one filter can be applied at a time. The challenge here, sensor activation, is to choose which sensors should be on when (Wang, Lafortune, Lin & Girard 2010, Wang 2016, Yin & Lafortune 2017*a*, Wang & Gong 2017). Next, we take methods in (Wang, Lafortune, Lin & Girard 2010, Yin & Lafortune 2017*a*) for calculating minimal SAPs that satisfies the requirements of supervisory control *without* control delay and apply them to calculate minimal SAPs that satisfies the requirements of supervisory control *with* control delay.

In the case of no delay in observation, for any strings that look the same to the supervisor, the sensor activation decisions and corresponding observations after these strings must be consistent (Wang, Lafortune, Lin & Girard 2010).

Definition 17 *Given some prefix-closed language L, for an observation mapping $\omega : L \to 2^{\Sigma_o}$ to be a sensor activation policy (SAP), it needs to satisfy*

$$(\forall s, s' \in L)[\theta^{\omega}(s) = \theta^{\omega}(s') \Rightarrow \omega(s) = \omega(s')] \tag{24}$$

Since (24) is only relevant to sensor activation decisions and irrelevant to implementing supervisory control decisions, if (24) is true without delay in implementing supervisory control decisions then it is also true with delay.

When considering SAPs, we assume the system G has finite-state, and the desired system is modeled by $H \sqsubseteq G$. Moreover, for implementation purposes, we partition $K = \mathscr{L}(H)$ so that each element of the partition corresponds to one set of active sensors. Formally, we specify a partition Π of $\mathscr{L}(H)$ first, and restrict the solution space of an SAP problem to be all functions over Π. So we only consider those SAPs ω over $\mathscr{L}(H)$ satisfying, for all $\pi \in \Pi$ and all $s, s' \in \pi$, $\omega(s) = \omega(s')$. The set of SAPs that are transition-based dynamic observations is the solution space of the SAP problem in (Wang, Lafortune, Lin & Girard 2010). Here we define Π as a partition of $\mathscr{L}(H)$ instead of $\mathscr{L}(G)$ because, under successful control, strings in $\mathscr{L}(G) \setminus \mathscr{L}(H)$ cannot occur and, thus, should not have an SAP associated with them.

In this paper, we assume no observation delay and the control delay is state-based when we optimize SAP.

Definition 18 *We say that N_c over $\mathscr{L}(H)$ is state-based if, for all $s, s' \in \mathscr{L}(H)$, $\delta_D(q_0, s) = \delta_D(q_0, s') \Rightarrow N_c(s) = N_c(s')$.*

Assumption 2 *When optimizing SAP, we assume $N_o \equiv 0$ and N_c is state-based.*

Lemma 7 *Assumption 2 holds iff the corresponding $N^{\uparrow}$ can be obtained by Definition 12 and $N^{\uparrow} \equiv N_c$.*

PROOF. Since $N_o \equiv 0$, by Definition 6, $N_{o+c} \equiv N_c$. Then, the proof follows from comparing Definition 12 with Definition 18. □

Lemma 7 says that we can directly use the results in Section 3.2 when optimizing SAP.In the rest of this section, we use $N^{\uparrow} \equiv N_c$ in the place of Assumption 2.

Problem 2 *Given G and $H \sqsubseteq G$ with Σ_o and Σ_{uc}, a partition Π of $K = \mathscr{L}(H)$, and state-based N_c, suppose $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega^{all}, N_c, 0)$ is true. Find an SAP $\omega^* : \Pi \to 2^{\Sigma_o}$ such that*

(a) $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega^*, N_c, 0)$ *is true.*
(b) For each SAP $\omega' : \Pi \to 2^{\Sigma_o}$, $\omega' \subset \omega^*$ implies $\neg\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega', N_c, 0)$.

Note that by definition, $\omega^{all} \equiv \Sigma_o$ is an SAP. We call a solution ω^* to Problem 2 a minimal SAP.

To proceed further, we need the concept of control conflict between two strings. We say two strings $s, s' \in K$ conflict with each other if there is a controllable event σ such that both $s\sigma$ and $s'\sigma$ are in the uncontrolled language, but only one of them is in K.

The reduction in Section 3 allows us to apply techniques from Wang, Lafortune, Lin & Girard (2010), Yin & Lafortune (2017*a*) to solve Problem 2. The problem remaining is that, in general $\mathscr{L}(\tilde{H})$ is a proper superset of $\mathscr{L}(H)$, and the part of SAP over $\mathscr{L}(\tilde{H}) \setminus \mathscr{L}(H)$ could affect its optimality and restrict the specification of SAP over $\mathscr{L}(H)$ through satisfying (24). This problem is further complicated by restricting SAPs over given Π. To resolve this, we devise a sublanguage of $\mathscr{L}(\tilde{H})$ and optimize SAP with no control delay over that sublanguage. Then, the optimal SAP under control delay over $\mathscr{L}(H)$ can be obtained from that result.

Definition 19 is used to specify the system to model the sublanguage mentioned above and the partition of that sublanguage that corresponds to the partition Π of $\mathscr{L}(H)$.

Definition 19 *Given G, $H \sqsubseteq G$, Σ_{uc}, and N_c with $N_c \equiv N^{\uparrow}$, partition Π of $\mathscr{L}(H)$, $\tilde{G}$ and $\tilde{H}$ computed by Algorithm 1,*

let $\hat{H} = (\hat{Q}_D, \tilde{\Sigma}, \hat{\delta}_D, \hat{\Gamma}_D, q_0) \sqsubseteq \tilde{H}$ *be the automaton obtained from* $\tilde{H}$ *by deleting state* d *and all transitions ending at* d *in* $\tilde{H}$*, and* $\eta(\Pi)$ *be a partition of* $\mathscr{L}(\hat{H})$ *such that, for all* $\pi \in \Pi$, $\eta(\pi) = \{s, sa, sb : s \in \pi\}$.

Proposition 3 says that to optimize SAP $\tilde{\omega}$ over $\mathscr{L}(\tilde{H})$, we only need to optimize SAP over $K \cup K'$, where K' is given by (8).

Proposition 3 *Suppose* G, $H \sqsubseteq G$, Σ_{uc}, ω, *and* N_c *with* $N_c \equiv N^{\uparrow}$, $\tilde{G}$, $\tilde{H}$, $\hat{H} \sqsubseteq \tilde{H}$, $\tilde{\omega}$ *over* $\mathscr{L}(\tilde{H})$, *and* $\Lambda(\tilde{G}, \tilde{H})$ *being the set of pair sets of states representing control conflicts are all given. Then,* $\mathscr{L}(H)$ *is delay observable w.r.t.* $\mathscr{L}(G)$, Σ_{uc}, ω, *and* $N^{\uparrow} \equiv N_c$ *iff, for all* $s, s' \in \mathscr{L}(\hat{H})$,

$$\theta^{\tilde{\omega}}(s) = \theta^{\tilde{\omega}}(s') \Rightarrow \{\hat{\delta}_D(q_0, s), \hat{\delta}_D(q_0, s')\} \notin \Lambda(\tilde{G}, \tilde{H}) \quad (25)$$

PROOF. For necessity, suppose $\mathscr{L}(H)$ is delay observable w.r.t. $\mathscr{L}(G)$, Σ_{uc}, ω, and $N^{\uparrow}$. By Theorem 3, $\mathscr{L}(\tilde{H})$ is observable w.r.t. $\mathscr{L}(\tilde{G})$, $\tilde{\Sigma}_c$, and $\tilde{\omega}$. Since $\mathscr{L}(\hat{H}) \subseteq \mathscr{L}(\tilde{H})$, by Lemma 6, (25) holds. For sufficiency, suppose (25) holds. By Algorithm 1, we have $\mathscr{L}(\hat{H}) = K \cup K'$ and $\tilde{\Gamma}_D(d) = \emptyset$. So, for an arbitrary $s \in \mathscr{L}(\tilde{H}) \setminus \mathscr{L}(\hat{H})$, we have $\tilde{\delta}_D(q_0, s) = d$. Then, since $\tilde{\Gamma}_D(d) = \emptyset$, Definition 15 implies that, for all $s' \in \mathscr{L}(\tilde{H})$, $\{\delta_D(q_0, s), \delta_D(q_0, s)\} \notin \Lambda(\tilde{G}, \tilde{H})$. Since $s \in \mathscr{L}(\tilde{H}) \setminus \mathscr{L}(\hat{H})$ is arbitrary, (25) implies, for all $t, t' \in \mathscr{L}(\tilde{H})$, $\theta^{\tilde{\omega}}(t) = \theta^{\tilde{\omega}}(t') \implies \{\tilde{\delta}_D(q_0, t), \tilde{\delta}_D(q_0, t')\} \notin \Lambda(\tilde{G}, \tilde{H})$. Then, by Lemma 6, $\mathscr{L}(\tilde{H})$ is observable w.r.t. $\mathscr{L}(\tilde{G})$, $\tilde{\Sigma}_c$, and $\tilde{\omega}$. Finally, by Theorem 3, K is delay observable w.r.t. $\mathscr{L}(G)$, Σ_{uc}, ω, and $N^{\uparrow}$. □

Under the assumption of no delay, i.e., $N_{o+c} = 0$, Wang, Lafortune, Lin & Girard (2010), Yin & Lafortune (2017*a*) each presents an offline algorithm for computing a minimal SAP that solves Problem 2 w.r.t. each specific type of SAP domain Π. Using one of these results with $\tilde{H}$ and $\tilde{G}$ modeling the desired system and the uncontrolled system, respectively, as a subroutine on line 4, Algorithm 2 computes a minimal SAP ω^* over Π with state-based upper-bound of control delay N_c offline. To simplify exposition, define function $\Psi(\hat{H}, \eta(\Pi), \Lambda(\tilde{G}, \tilde{H}))$ as the function that implements the algorithm of (Wang, Lafortune, Lin & Girard 2010, Yin & Lafortune 2017*a*) and returns a minimal SAP $\tilde{\omega}^* : \eta(\Pi) \to 2^{\tilde{\Sigma}_o}$ under which (25) holds.

To prove the correctness of Algorithm 2 in Theorem 4, we will use the following lemma proved in (Wang, Lafortune, Lin & Girard 2010).

Lemma 8 *Given* H, Σ_o, Σ_c, *and SAPs* ω *and* ω' *over* $\mathscr{L}(H)$ *with* $\omega \subset \omega'$, *then, for all* $s, s' \in \mathscr{L}(H)$, $\theta^{\omega'}(s) = \theta^{\omega'}(s')$ *implies* $\theta^{\omega}(s) = \theta^{\omega}(s')$.

We then have the following conclusions.

Algorithm 2 MIN-SAP-WITH-CONTROL-DELAY

Input: G and $H \sqsubseteq G$ with Σ_o and Σ_{uc}, Π, and N_c;
Output: An SAP $\omega^*_{N_c} : \Pi \to 2^{\Sigma_o}$;

1: **if** $\neg \mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega^{all}, N_c, 0)$, **then**
2: **return** "No solution exists";
3: **end if**
4: Run Algorithm 1 to compute $\tilde{G}$ and $\tilde{H}$ with $\tilde{\Sigma}_{uc}$ and $\tilde{\Sigma}_o$;
5: Compute $\Lambda(\tilde{G}, \tilde{H})$ and $\hat{H}$ (Definitions 15 and 19);
6: $\tilde{\omega}^* \leftarrow \Psi(\hat{H}, \eta(\Pi), \Lambda(\tilde{G}, \tilde{H}))$;
7: For each $\pi \in \Pi$, $\omega^*(\pi) \leftarrow \tilde{\omega}^*(\eta(\pi))$;
8: **return** ω^*;

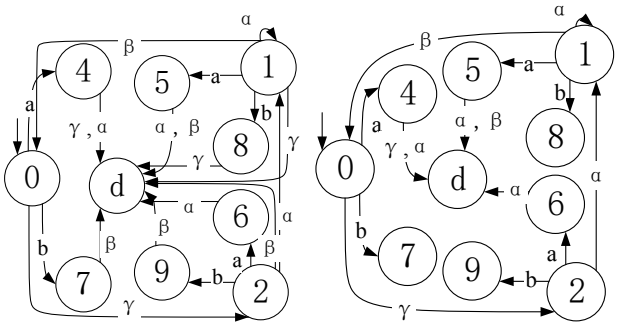
Theorem 4 *SAP* ω^* *computed by Algorithm 2 is a solution to Problem 2 if such a solution exists and "No solution exists" otherwise.*

PROOF. Consider that $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega^{all}, N_c, 0)$ is false (line 1). By Theorem 1, $\mathscr{L}(H)$ either is not controllable or is not delay observable with upper-bound of state-based observation delay $N^{\uparrow} \equiv N_c$. There are two cases.

If $\mathscr{L}(H)$ is not controllable, then, by Theorem 1, for each SAP ω, $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega, N_c, 0)$ is false, i.e., Problem 2 does not have a solution. If $\mathscr{L}(H)$ is controllable but is not delay observable, then, let $\tilde{\omega}^{all}$ over $\mathscr{L}(\hat{H})$ be the SAP that corresponds to ω^{all} by Definition 11. By Proposition 3, (25) does not holds under $\tilde{\omega}^{all}$. Then, by Lemma 8, there does not exist an SAP $\tilde{\omega}$ over $\mathscr{L}(\hat{H})$ such that (25) holds under $\tilde{\omega}$. Then, applying Proposition 3 and then Theorem 1, we conclude that there does not exist an SAP ω such that $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega, N_c, 0)$ is true, i.e., Problem 2 again does not have a solution and the algorithm returns "No solution exits".

Consider that $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega^{all}, N_c, 0)$ is true (line 1). We first prove ω^* is an SAP. By Definition 19, for all $\pi \in \Pi$, $\pi \subseteq \eta(\pi)$. Hence, by line 7, for all $s \in \mathscr{L}(H)$, $\omega^*(s) = \tilde{\omega}^*(s)$, which implies $\theta^{\omega^*}(s) = \theta^{\tilde{\omega}^*}(s)$ by Definition 1. Then, since SAP $\tilde{\omega}^*$ over $\mathscr{L}(\tilde{H})$ satisfies (24) and $\mathscr{L}(\hat{H}) \subseteq \mathscr{L}(\tilde{H})$, ω^* over $\mathscr{L}(H)$ satisfies (24), i.e., ω^* over Π is an SAP. For (a) of Problem 2, since $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega^{all}, N_c, 0)$ is true, by Theorem 1, $\mathscr{L}(H)$ is controllable w.r.t. $\mathscr{L}(G)$ and Σ_{uc}. Moreover, by line 6, $\tilde{\omega}^*$ satisfies (25). Then, by Proposition 3, $\mathscr{L}(H)$ is delay observable w.r.t. $\mathscr{L}(G)$, Σ_{uc}, ω, and $N^{\uparrow} \equiv N_c$. Hence, by Theorem 1, $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega^*_{N_c}, N_c, 0)$ is true as required. We prove (b) by proving the contrapositive. Suppose SAP ω^* computed on line 7 is not minimal. Then, there exists a minimal SAP, say $\omega' : \Pi \to 2^{\Sigma_o}$ with $\omega' \subset \omega^*$, under which $\mathsf{P}(\mathscr{L}(G), \mathscr{L}(H), \Sigma_{uc}, \omega', N_c, 0)$ is true. Define $\tilde{\omega}' : \eta(\Pi) \to 2^{\tilde{\Sigma}_o}$ as, for all $s \in \mathscr{L}(H)$, $\tilde{\omega}'(s) = \tilde{\omega}'(sa) = \tilde{\omega}'(sb) \leftarrow \omega'(s)$. Since $\omega' \subset \omega^*$, by the definitions of $\tilde{\omega}^*$, which is computed on line 6, and $\tilde{\omega}'$, $\tilde{\omega}' \subset \tilde{\omega}^*$. Then, by Theorem 1 and Proposition 3, (25) for $\mathscr{L}(\tilde{H})$ and $\tilde{\omega}'$ holds, which contradicts the fact that $\tilde{\omega}^*$ is a minimal SAP under which (25) holds. □

By Theorem 4, all the procedures in (Wang, Lafortune, Lin & Girard 2010, Yin & Lafortune 2017*a*) for computing a minimal SAP, under which observability is satisfied, can be used to calculate a minimal SAP for a deterministic networked control problem with control delay but without observation delay.



(a) Automaton $\tilde{G}$ (b) Automaton $\tilde{H}$

Fig. 5. Automata $\tilde{G}$ and $\tilde{H}$ in Example 5.

Example 5 *Consider system G and desired system H with* $\Sigma = \Sigma_c = \Sigma_o = \{\alpha, \beta, \gamma\}$ *shown in Fig.3. Also suppose* $N^{\uparrow}(0) = 1$ *and* $N^{\uparrow}(1) = N^{\uparrow}(2) = 0$*. Applying Algorithm 2, we solve Problem 2 for a minimal SAP that is a transition-based observation mapping presented at the end of Section 3.2. We have* $\mathsf{P}(\mathscr{L}(G), K, \Sigma_{uc}, \omega^{all}, N_c, 0)$ *is true on line 1. Executing line 4 gives* $\tilde{G}$ *shown in Fig. 5(a) and* $\tilde{H}$ *shown in Fig. 5(b). Executing line 5 gives* $\Lambda(\tilde{G}, \tilde{H}) = \{\{0,1\},\{0,8\},\{1,2\},\{1,7\},\{1,9\},\{4,1\},\{4,8\},\{5,2\},\{5,7\},\{5,9\}\}$*. Wang, Lafortune, Lin & Girard (2010) write transition-based SAP* ω *as a set* $\Omega = \{(q,\sigma) : q \in Q_D \wedge \sigma \in \omega(q) \cap \Gamma_D(q)\}$ *and compute a minimal set* Ω^* *under which observability holds. Using this form, executing line 6 gives minimal SAP* $\tilde{\Omega}^* = \{(1,\beta),(2,\alpha)\}$ *for* $\tilde{H}$*. The minimal SAP for H computed by line 7 is* $\Omega^* = \tilde{\Omega}^*$*.* □

In the case of observation delay, the delayed information mapping of a string could contain several strings, which complicates sensor activation decisions. Minimizing SAP with observation delay remains as future work.

5 Conclusion

We have extended the formulation of deterministic networked control problems in (Shu & Lin 2017) in the following two directions: 1) We allow the upper bounds of both observation delay and control delay to depend on strings, and show that important properties depend only on the sum of these two delays. 2) We have considered the deterministic networked control problem under dynamic observations rather than static observations. Based on the extended formulation, a new necessary and sufficient condition for the existence of a deterministic networked supervisor was introduced in Theorem 1, which is more general yet simpler than the one in the literature. Then, we have reduced the problem of deterministic networked supervisory control to the problem of supervisory control. As an immediate application, this allows us to use the verifier for observability to test the corresponding delay observability. Finally, this reduction was used to leverage techniques for optimizing sensor activations for supervisory control without control delay to optimize sensor activations for supervisory control with control delay.

This work has raised many questions. One such is how to represent delay. This work has considered delays bounded by the number of events, but in many cases it would be useful to bound the actual time of the delays. An interesting open problem is how to use models like those defined in (Balemi 1992, Rashidinejad et al. 2018) and extend this work for delay bounded above by time.

References

Balemi, S. (1992), Communication delays in connections of input/output discrete event processes, *in* '[1992] Proceedings of the 31st IEEE Conference on Decision and Control', IEEE, pp. 3374–3379.

Balemi, S. & Brunner, U. (1992), Supervision of discrete event systems with communication delays, *in* '1992 American Control Conference', IEEE, pp. 2794–2798.

Cassandras, C. G. & Lafortune, S. (2009), *Introduction to discrete event systems*, Springer Science & Business Media.

Cassez, F. & Tripakis, S. (2008), 'Fault diagnosis with static and dynamic observers', *Fundamenta Informaticae* **88**(4), 497–540.

Heymann, M. & Lin, F. (1994), 'On-line control of partially observed discrete event systems', *Discrete Event Dynamic Systems: Theory and Applications* **4**(3), 221–236.

Huang, Y., Rudie, K. & Lin, F. (2008), 'Decentralized control of discrete-event systems when supervisors observe particular event occurrences', *IEEE Transactions on Automatic Control* **1**(53), 384–388.

Ji, Y., Yin, X. & Lafortune, S. (2021), 'Optimal supervisory control with mean payoff objectives and under partial observation', *Automatica* **123**, 109359.

Ji, Y., Yin, X. & Lafortune, S. (2022), 'Local mean payoff supervisory control for discrete event systems', *IEEE Transactions on Automatic Control* **67**(5), 2282–2297.

Lin, F. (1993), 'Roubust and adaptive supervisory control of discrete event systems', *IEEE Transactions on Automatic Control* **38**(12), 1848–1852.

Lin, F. (2014), 'Control of networked discrete event systems: dealing with communication delays and losses', *SIAM Journal on Control and Optimization* **52**(2), 1276–1298.

Lin, F., Wang, W., Han, L. & Shen, B. (2019), 'State estimation of multichannel networked discrete event systems', *IEEE Transactions on Control of Network Systems* **7**(1), 53–63.

Lin, F. & Wonham, W. M. (1988*a*), 'Decentralized supervisory control of discrete-event systems', *Information Sciences* **44**(3), 199–224.

Lin, F. & Wonham, W. M. (1988*b*), 'On observability of

discrete-event systems', *Information Sciences* **44**(3), 173–198.

Liu, Z., Yin, X., Shu, S., Lin, F. & Li, S. (2022), 'Online supervisory control of networked discrete-event systems with control delays', *IEEE Transactions on Automatic Control* **67**(5), 2314–2329.

Meira-Góes, R., Kang, E., Kwong, R. H. & Lafortune, S. (2020), 'Synthesis of sensor deception attacks at the supervisory layer of cyber–physical systems', *Automatica* **121**, 109172.

Ouedraogo, L., Kumar, R., Malik, R. & Akesson, K. (2011), 'Nonblocking and safe control of discrete-event systems modeled as extended finite automata', *IEEE Transactions on Automation Science and Engineering* **8**(3), 560–569.

Park, S.-J. & Cho, K.-H. (2006), 'Delay-robust supervisory control of discrete-event systems with bounded communication delays', *IEEE Transactions on Automatic Control* **51**(5), 911–915.

Park, S.-J. & Cho, K.-H. (2007), 'Decentralized supervisory control of discrete event systems with communication delays based on conjunctive and permissive decision structures', *Automatica* **43**(4), 738–743.

Ramadge, P. J. & Wonham, W. M. (1987), 'Supervisory control of a class of discrete event processes', *SIAM journal on control and optimization* **25**(1), 206–230.

Rashidinejad, A. (2021), Model-based supervisory control synthesis of cyber-physical systems, PhD thesis, Eindhoven University of Technology.

Rashidinejad, A., Reniers, M. & Feng, L. (2018), 'Supervisory control of timed discrete-event systems subject to communication delays and non-fifo observations', *IFAC-PapersOnLine* **51**(7), 456–463.

Rashidinejad, A., van der Graaf, P., Reniers, M. & Fabian, M. (2020), 'Non-blocking supervisory control of timed automata using forcible events', *IFAC-PapersOnLine* **53**(4), 356–362.

Shu, S., Huang, Z. & Lin, F. (2013), 'Online sensor activation for detectability of discrete event systems', *IEEE Trans. Automation Science and Engineering* **10**(2), 457–461.

Shu, S. & Lin, F. (2014), 'Decentralized control of networked discrete event systems with communication delays', *Automatica* **50**(8), 2108–2112.

Shu, S. & Lin, F. (2015), 'Supervisor synthesis for networked discrete event systems with communication delays', *IEEE Transactions on Automatic Control* **60**(8), 2183–2188.

Shu, S. & Lin, F. (2016), 'Predictive networked control of discrete event systems', *IEEE Transactions on Automatic Control* **62**(9), 4698–4705.

Shu, S. & Lin, F. (2017), 'Deterministic networked control of discrete event systems with nondeterministic communication delays', *IEEE Transactions on Automatic Control* **62**(1), 190–205.

Su, R. (2018), 'Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations', *Automatica* **94**, 35–44.

Tripakis, S. (2004), 'Decentralized control of discrete-event systems with bounded or unbounded delay communication', *IEEE Transactions on Automatic Control* **49**(9), 1489–1501.

Wang, W. (2016), 'Online minimization of sensor activation for supervisory control', *Automatica* **73**, 8–14.

Wang, W. & Gong, C. (2017), 'Calculating all minimal transition-based sensor activation policies for the purpose of supervisory control', *IEEE Transactions on Automatic Control* **62**(11), 5894–5901.

Wang, W., Lafortune, S., Girard, A. R. & Lin, F. (2010), 'Optimal sensor activation for diagnosing discrete event systems', *Automatica* **46**(7), 1165–1175.

Wang, W., Lafortune, S. & Lin, F. (2007), 'An algorithm for calculating indistinguishable states and clusters in finite-state automata with partially observable transitions', *Systems & control letters* **56**(9), 656–661.

Wang, W., Lafortune, S. & Lin, F. (2008), 'Minimization of communication of event occurrences in acyclic discrete event systems', *IEEE Transactions on Automatic Control* **53**(9), 2197–2202.

Wang, W., Lafortune, S. & Lin, F. (2011), 'On codiagnosability and coobservability with dynamic observations', *IEEE Transactions on Automatic Control* **56**(7), 1551–1566.

Wang, W., Lafortune, S., Lin, F. & Girard, A. R. (2010), 'Minimization of sensor activation in discrete event systems for the purpose of control', *IEEE Transactions on Automatic Control* **55**(11), 2447–2461.

Wonham, W. M. & Ramadge, P. J. (1987), 'On the supremal controllable sublanguage of a given language', *SIAM journal on control and optimization* **25**(3), 637–659.

Yin, X. & Lafortune, S. (2016), 'Synthesis of maximally permissive supervisors for partially-observed discrete-event systems', *IEEE Transactions on Automatic Control* **61**(5), 1239–1254.

Yin, X. & Lafortune, S. (2017*a*), 'Minimization of sensor activation in decentralized discrete-event systems', *IEEE Transactions on Automatic Control* **63**(11), 3705–3718.

Yin, X. & Lafortune, S. (2017*b*), 'Synthesis of maximally-permissive supervisors for the range control problem', *IEEE Transactions on Automatic Control* **62**(8), 3914–3929.

Zhang, R., Cai, K., Gan, Y. & Wonham, W. M. (2016), 'Distributed supervisory control of discrete-event systems with communication delay', *Discrete Event Dynamic Systems* **26**(2), 263–293.



Weilin Wang received the M.S. degree in electrical engineering systems, the M.S.E. degree in industrial engineering, and the Ph.D. degree in electrical engineering systems from the University of Michigan, Ann Arbor, in 2003, 2006, and 2007, respectively. He is currently with the Department of Control Science and Engineering, University of Shanghai for Science and Technology. He is also with Zhejiang Research Center on Smart Rail Transportation, and with Monash University. He is a senior member of Institute of Electrical and Electronics Engineers. His research interests include system and control theory, discrete event systems, formal methods for control, machine learning, and application to intelligent transportation systems and multi-robot systems.



Yanwei Zang received his B.S. degree in civil engineering from Xinan Jiaotong University and M.S. degree in highway and railway engineering from Tongji University in 2004 and 2007, respectively. He is currently the vice president of Huadong Engineering Corporation Limited and the executive director of Zhejiang Research Center on Smart Rail Transportation. His research interests include optimization and control of urban rail systems, data analytics for municipal transportation, tunnel engineering, and intelligent transportation systems.



Shigemasa Takai received the B.E. and M.E. degrees from Kobe University, Kobe, Japan, in 1989 and 1991, respectively, and the Ph.D degree from Osaka University, Suita, Japan, in 1995. From 1992 to 1998, he was a Research Associate at Osaka University. In 1998, he joined Wakayama University, Wakayama, Japan, as a Lecturer, and became an Associate Professor in 1999. From 2004 to 2009, he was an Associate Professor with Kyoto Institute of Technology, Kyoto, Japan. Since 2009, he has been a Professor with Osaka University, Suita, Osaka, Japan. His research interests include supervisory control and fault diagnosis of discrete event systems.



Lachlan L. H. Andrew received the B.Sc., B.E., and Ph.D. degrees from the University of Melbourne in 1992, 1993, and 1997, respectively. He has been with the University of Melbourne since 2019. From 2014 to 2018, he was with Monash University and the Swinburne University of Technology, Australia, from 2008 to 2014. From 2010 to 2014, he was an ARC Future Fellow. He was a Senior Research Fellow with the University of Melbourne and a Lecturer with RMIT, Australia. From 2005 to 2008, he was a Senior Research Engineer with the Department of Computer Science, Caltech. His research interests include algorithmic aspects of energy management, and performance analysis of network resource allocation algorithms. He was a co-recipient of the ACM SIGMETRICS test of time award in 2021, The IEEE William R. Bennett Prize in 2014, and the best paper award at IGCC2012, IEEE INFOCOM 2011, and IEEE MASS 2007. He is a Member of the ACM.



Chaohui Gong received her M.S. degree in mathematical statistics from Wayne State University in 2007. She is currently with the University of Shanghai for Science and Technology and is with Zhejiang Research Center on Smart Rail Transportation. Her research interests include optimization algorithms, discrete event systems, machine learning, and application to intelligent transportation systems and multi-robot systems.