Localization-Based Distributed Control of Timed Discrete-Event Systems with Communication Delay

Renyuan Zhang[a], Kai Cai[b]

[a]School of Automation, Northwestern Polytechnical University, Xi'an, China;
[b]Department of Electrical and Information Engineering, Osaka City University, Osaka, Japan.

ARTICLE HISTORY
Compiled July 1, 2020

ABSTRACT
We study *localization-based* distributed control of timed multi-component discrete-event systems with *communication delay* in the Brandin-Wonham framework. First, we propose channel models for inter-component event communication with bounded and unbounded delays; the channel models are treated as plant components. In this formulation, there exist multiple distinct observable event sets; thus we employ timed relative coobservability to synthesize partial-observation decentralized supervisors. Then, we localize these supervisors into local controllers and preemptors, which provably tolerate the specified bounded and unbounded communication delays. Finally, the derived local controllers and preemptors are allocated to each plant component, and thus building a purely distributed control architecture of T-DES with communication delay. The above results are illustrated by a timed workcell example.

KEYWORDS
Timed Discrete-Event Systems; Communication Delay; Supervisor Localization

1. Introduction

In Cai and Wonham (2010a, 2010b, 2016), the authors developed a top-down approach, called *supervisor localization*, to the distributed control synthesis of multi-component discrete-event systems (DES). The essence of localization is the decomposition of the monolithic (optimal and nonblocking) supervisor into local controllers for the individual components. In (Zhang, Cai, Gan, Wang, & Wonham, 2013), supervisor localization was extended to timed DES (TDES) in the Brandin-Wonham framework (Brandin & Wonham, 1994); in addition to local controllers (corresponding to disabling actions), a set of local preemptors is obtained corresponding to clock-preempting actions. More recently in (Zhang, Cai, & Wonham, 2017) and (Zhang & Cai, 2020), supervisor localization was further extended to the case of partial observation. In particular, localization was combined with relative observability (Cai, Zhang, & Wonham, 2015) ((Cai, Zhang, & Wonham, 2016) for timed DES) to first synthesize a partial-observation monolithic supervisor, and then the supervisor was decomposed into partial-observation local controllers (and preemptors) whose state changes are

CONTACT R. Zhang. Email: ryzhang@nwpu.edu.cn

caused only by observable events. By an allocation policy described in (Zhang & Cai, 2020), each derived partial-observation local preemptor/controller is owned by exactly one plant component, thereby building a purely distributed control architecture. We refer to a plant component equipped with a set of partial-observation local controllers/preemptors as an *agent*. In these contributions, it is assumed that agents make independent observations and decisions, with instantaneous inter-agent communication. This assumption may be unrealistic in practice, where controllers are linked by a physical network subject to delays.

In this paper, we consider that inter-agent event communication is subject to delay. First, we introduce two types of channel models for inter-agent event communication. The introduced models are treated as plant components. In this formulation, the observable event sets of different agents are generally distinct. This is because the occurrence of a communication event and sending that event are observable only to the sender, but not observable to the receiver; on the other hand, receiving of a communication event is observable only to the receiver, but not observable to the sender. To deal with multiple observable event sets, we propose to employ the concept of timed relative coobservability (Cai et al., 2016) to first synthesize a set of partial-observation decentralized supervisors, and then decompose these decentralized supervisors into the respective local controllers/preemptors. Finally, we prove that the derived local controlled behavior is identical to that achieved by the partial-observation decentralized supervisors.

The main contributions of this work are as follows.

- A TDES channel model which can represent bounded and unbounded communication delays is proposed. Unlike (Zhang, Cai, Gan, & Wonham, 2016a, 2016b), the channel model is treated as plant component, and thus the communication delays are integrated into the plant behavior.
- Timed relative coobservability (Cai et al., 2016) is adopted to effectively compute decentralized supervisors tolerant of communication delays. Relative coobservability is stronger than coobservability, but permits existence of the supremal element; an algorithm in (Cai et al., 2016) effectively computes the supremal relatively coobservable sublanguage of a given (non-closed) language. The combination of timed relative coobservability and partial-observation supervisor localization is new, and leads to a computationally effective solution to delay-tolerant distributed control.
- It is established by Theorem 1 that the resulting partial-observation local *tick*-preemptors and local controllers are control equivalent to the computed decentralized supervisors, and thus the distributed control architecture built from them is guaranteed to tolerate specified bounded and unbounded communication delays.

Overall, the proposed supervisor localization for TDES with communication delay provides a top-down, computationally effective approach to the distributed control of timed DES with communication delay, which was not available in the literature.

We note that distributed/decentralized supervisory control with communication delay has been extensively studied. First, to capture communication delays in multi-component plant, there are mainly two approaches reported in the literature. The first is to model the communication by separate models, e.g. information structure (Barrett & Lafortune, 2000), FIFO queue (Hiraishi, 2009; Kalyon, Gall, Marchand, & Massart, 2011; Tripakis, 2004), shared medium communication model (Schmidt,

Schmidt, & Zaddach, 2007); then the plant behavior with communication delay will be obtained through appropriate composition operators on the plant components and the communication models. The other approach is to define observation maps (Lin, 2014; Park & Cho, 2007; Ricker & Caillaud, 2011; Xu & Kumar, 2008) on the plant behavior; then the plant behavior with communication delay is exactly the codomain of the observation maps. In this paper, we use a TDES channel model in which the communication delays are measured by number of *tick*s, and the delays at each transmission period are modeled separately. Compared with the models in the literature, our channel model is represented by TDES and treated as plant component; thus the plant behavior with delay can be obtained by synchronous product defined on (generalized) TDES, rather than by any newly defined composition operators.

Second, to synthesize distributed/decentralized supervisors that are able to tolerate specified communication delays, there are mainly two approaches reported in the literature. The first is a verification approach, e.g. (Sadid, Ricker, & Hashtrudi-Zad, 2015; Zhang et al., 2016a, 2016b), which first synthesizes delay-free distributed controllers, and then verifies whether the distributed controllers tolerate given communication delays. This approach is limited to verifying the robustness of derived controllers (Zhang et al., 2016a, 2016b) or that of existing communication protocols (Sadid et al., 2015), but does not supply any procedure to construct controllers that are able to tolerate given communication delays. The second approach is that of synthesis, e.g. (Barrett & Lafortune, 2000; Hiraishi, 2009; Lin, 2014; Park & Cho, 2007; Ricker & Caillaud, 2011; Tripakis, 2004), which first incorporates communication delays into the plant and specification models, and then applies decentralized control methods to synthesize distributed/decentralized controllers that tolerate given communication delay. In these works, *observability* (Barrett & Lafortune, 2000; Hiraishi, 2009), *joint observability* (Tripakis, 2004), *coobservability* (Ricker & Caillaud, 2011), *delay-coobservability* (Park & Cho, 2007), or *network observability* (Lin, 2014) are necessary for the existence of distributed controllers tolerant of communication delay. However, these observability properties are not closed under set union, and thus there generally does not exist the respective supremal sublanguage of a given language and if the given language does not satisfy these properties, co-normality (stronger than coobservability, but is closed under set union (Rudie & Wonham, 1992)), or conditional decomposability (Komenda & Masopust, 2017; Komenda, Masopust, & van Schuppen, 2012) may be applied to synthesize delay-tolerant decentralized/distributed controllers. By contrast, we employ the recently proposed timed relative coobservability, which is closed under set union and the supremal relatively coobservable sublanguage is effectively computable (Cai et al., 2016). Thus this approach will synthesize a set of local controller (and preemptors corresponding to clock-preempting actions (Zhang & Cai, 2020; Zhang et al., 2013)) tolerating prescribed communication delays; moreover, since relative coobervability is weaker than co-normality (Cai et al., 2016), the controlled system behavior will generally more permissive than its co-normality counterpart.

The paper is organized as follows. Section 2 reviews the concept of timed relative coobservability in the Brandin-Wonham TDES framework and the partial-observation localization procedure of TDES. Section 3 presents the two TDES communication channel models with bounded and unbounded delays, and Section 4 investigates partial-observation supervisor localization with communication delay by using the concept of timed relative coobservability. Section 5 illustrates the procedure by a timed workcell example. Finally Section 6 states our conclusions.

2. Preliminaries

This section reviews the concept of timed relative coobservability of decentralized supervisory control of TDES in the Brandin-Wonham framework ((Brandin & Wonham, 1994);(Wonham & Cai, 2019, Chapter 9)). First consider the untimed DES model $\mathbf{G}_{act} = (A, \Sigma_{act}, \delta_{act}, a_0, A_m)$; here A is the finite set of *activities*, Σ_{act} the finite set of *events*, $\delta_{act} : A \times \Sigma_{act} \to A$ the (partial) *transition function*, $a_0 \in A$ the *initial activity*, and $A_m \subseteq A$ the set of *marker activities*. Let $\mathbb{N}$ denote the set of natural numbers $\{0, 1, 2, ...\}$, and introduce *time* into $\mathbf{G}_{act}$ by assigning to each event $\sigma \in \Sigma_{act}$ a *lower bound* $l_{\mathbf{G},\sigma} \in \mathbb{N}$ and an *upper bound* $u_{\mathbf{G},\sigma} \in \mathbb{N} \cup \{\infty\}$, such that $l_{\mathbf{G},\sigma} \leq u_{\mathbf{G},\sigma}$. Also introduce a distinguished event, written *tick*, to represent "tick of the global clock". Then a TDES model

$$\mathbf{G} := (Q, \Sigma, \delta, q_0, Q_m), \tag{1}$$

is constructed from $\mathbf{G}_{act}$ (refer to (Brandin & Wonham, 1994) and (Wonham & Cai, 2019, Chapter 9) for detailed construction) such that Q is the finite set of *states*, $\Sigma := \Sigma_{act} \dot{\cup} \{tick\}$ the finite set of events, $\delta : Q \times \Sigma \to Q$ the (partial) *state transition function*, q_0 the *initial state*, and Q_m the set of *marker states*.

Let Σ^* be the set of all finite strings of elements in $\Sigma = \Sigma_{act} \dot{\cup} \{tick\}$, including the empty string ϵ. The transition function δ is extended to $\delta : Q \times \Sigma^* \to Q$ in the usual way. The *closed behavior* of $\mathbf{G}$ is the language $L(\mathbf{G}) := \{s \in \Sigma^* | \delta(q_0, s)!\}$ and the *marked behavior* is $L_m(\mathbf{G}) := \{s \in L(\mathbf{G}) | \delta(q_0, s) \in Q_m\} \subseteq L(\mathbf{G})$. Let $K \subseteq \Sigma^*$ be a language; its *prefix closure* is $\overline{K} := \{s \in \Sigma^* | (\exists t \in \Sigma^*) \ st \in K\}$. K is said to be $L_m(\mathbf{G})$-*closed* if $\overline{K} \cap L_m(\mathbf{G}) = K$. TDES $\mathbf{G}$ is *nonblocking* if $\overline{L_m(\mathbf{G})} = L(\mathbf{G})$.

A TDES $\mathbf{G}$ can be graphically represented by both its *activity transition graph* (ATG), namely the ordinary transition graph of $\mathbf{G}_{act}$, and its *timed transition graph* (TTG), namely the ordinary transition graph of $\mathbf{G}$, incorporating the *tick* transition explicitly.

For two TDES $\mathbf{G}_1$ and $\mathbf{G}_2$ with ATG $\mathbf{G}_{1,act}$ and $\mathbf{G}_{2,act}$ defined on $\Sigma_{1,act}$ and $\Sigma_{2,act}$ respectively, their *composition* $\mathbf{Comp}(\mathbf{G}_1, \mathbf{G}_2)$, is a new TDES $\mathbf{G}$ such that $\mathbf{G}_{act} = \mathbf{G}_{1,act} || \mathbf{G}_{2,act}$, where "$||$" denotes the synchronous product of two generators (Wonham & Cai, 2019). The time bounds on the events of $\mathbf{G}$ are determined by: if $\sigma \in \Sigma_{1,act} \cap \Sigma_{2,act}$, then $l_{\mathbf{G},\sigma} = max(l_{\mathbf{G}_1,\sigma}, l_{\mathbf{G}_2,\sigma})$ and $u_{\mathbf{G},\sigma} = min(u_{\mathbf{G}_1,\sigma}, u_{\mathbf{G}_2,\sigma})$; if $\sigma \in \Sigma_{1,act} \setminus \Sigma_{2,act}$, then $l_{\mathbf{G},\sigma} = l_{\mathbf{G}_1,\sigma}$ and $u_{\mathbf{G},\sigma} = u_{\mathbf{G}_1,\sigma}$; if $\sigma \in \Sigma_{2,act} \setminus \Sigma_{1,act}$, then $l_{\mathbf{G},\sigma} = l_{\mathbf{G}_2,\sigma}$ and $u_{\mathbf{G},\sigma} = u_{\mathbf{G}_2,\sigma}$. If this leads to $l_{\mathbf{G},\sigma} > u_{\mathbf{G},\sigma}$, the composition $\mathbf{G}$ does not exist.[1] Composition of more than two TDES can be similarly constructed.[2]

To use TDES $\mathbf{G}$ in (1) for supervisory control, first designate a subset of events, denoted by $\Sigma_{hib} \subseteq \Sigma_{act}$, to be the *prohibitible* events which can be disabled by an external supervisor. Next, and specific to TDES, specify a subset of *forcible* events, denoted by $\Sigma_{for} \subseteq \Sigma_{act}$, which can *preempt* the occurrence of event *tick*. Now it is

[1] We stress that $\mathbf{Comp}(\mathbf{G}_1, \mathbf{G}_2)$ is in general different from the result of $\mathbf{G}_1 || \mathbf{G}_2$, for the latter would force the synchronization of *tick* transition as it occurs in the components. Specifically, when $\Sigma_{1,act} \cap \Sigma_{2,act} = \emptyset$, $\mathbf{Comp}(\mathbf{G}_1, \mathbf{G}_2) \approx \mathbf{G}_1 || \mathbf{G}_2$ where $\approx$ denotes that the closed and marked behavior of the TDES coincide (Wonham & Cai, 2019).

[2] There also exist *generalized* TDES (as defined in (Wonham & Cai, 2019, Section 9.11)), which are represented by only TTG including *tick* in the alphabet. Namely, a generalized TDES does not have a corresponding ATG or timer information, and is simply an ordinary finite-state generator whose event set includes *tick*. Generalized TDES are often adopted to model temporal specifications and supervisors, and represent controlled plant behaviors. To compose two or more generalized TDES, we use the synchronous product "$||$", rather than **Comp**.

convenient to define the *controllable* event set $\Sigma_c := \Sigma_{hib} \dot{\cup} \{tick\}$. The *uncontrollable* event set is $\Sigma_{uc} := \Sigma \setminus \Sigma_c$. A sublanguage $K \subseteq L_m(\mathbf{G})$ is *controllable* if, for all $s \in \overline{K}$,

$$Elig_K(s) \supseteq \begin{cases} Elig_{\mathbf{G}}(s) \cap (\Sigma_{uc} \dot{\cup} \{tick\}) & \text{if} \quad Elig_K(s) \cap \Sigma_{for} = \emptyset, \\ Elig_{\mathbf{G}}(s) \cap \Sigma_{uc} & \text{if} \quad Elig_K(s) \cap \Sigma_{for} \neq \emptyset, \end{cases}$$

where $Elig_K(s) := \{\sigma \in \Sigma | s\sigma \in \overline{K}\}$ is the subset of eligible events after string s.

For partial observation, Σ is partitioned into Σ_o, the subset of observable events, and Σ_{uo}, the subset of unobservable events (i.e. $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$). Bring in the *natural projection* $P : \Sigma^* \to \Sigma_o^*$ defined by:

$$\begin{aligned} P(\epsilon) &= \epsilon; \\ P(\sigma) &= \begin{cases} \epsilon, & \text{if } \sigma \notin \Sigma_o, \\ \sigma, & \text{if } \sigma \in \Sigma_o; \end{cases} \\ P(s\sigma) &= P(s)P(\sigma), \quad s \in \Sigma^*, \sigma \in \Sigma. \end{aligned} \tag{2}$$

As usual, P is extended to $P : Pwr(\Sigma^*) \to Pwr(\Sigma_o^*)$, where $Pwr(\cdot)$ denotes powerset, i.e., for any language $K \in Pwr(\Sigma^*), P(K) := \{Ps | s \in K\}$. Write $P^{-1} : Pwr(\Sigma_o^*) \to Pwr(\Sigma^*)$ for the *inverse-image function* of P.

Let $\Sigma_{o,i} \subseteq \Sigma$ and the natural projections $P_i : \Sigma^* \to \Sigma_{o,i}^*$, $i \in \mathcal{I}$ ($\mathcal{I}$ is some index set). Also let $\Sigma_{hib,i} \subseteq \Sigma_{rem}$ and $\Sigma_{for,i} \subseteq \Sigma_{act}$. Define $\Sigma_{c,i} := \Sigma_{hib,i} \dot{\cup} \{tick\}$ be the controllable event set for each $i \in \mathcal{I}$ (thus $tick \in \Sigma_{c,i}$ for each $i \in \mathcal{I}$). We consider decentralized supervisor control where each decentralized supervisor $i \in \mathcal{I}$ observes events only in $\Sigma_{o,i}$, disables events only in $\Sigma_{hib,i}$, and uses forcible events only in $\Sigma_{for,i}$ to preempt *tick*.

A sublanguage $K \subseteq C \subseteq L_m(\mathbf{G})$ is *timed relatively coobservable* (with respect to C, $\mathbf{G}$ and P_i, $i \in \mathcal{I}$), or simply timed C-coobservable, if for every $i \in \mathcal{I}$ and every pair of strings $s, s' \in \Sigma^*$ with $P_i(s) = P_i(s')$ there holds

$$(\forall \sigma \in \Sigma_{c,i})\ s\sigma \in \overline{K}, s' \in \overline{C}, s'\sigma \in L(\mathbf{G}) \Rightarrow s'\sigma \in \overline{K}. \tag{3}$$

For an arbitrary sublanguage $E \subseteq L_m(\mathbf{G})$, write $\mathcal{CCO}(E \cap L_m(\mathbf{G}))$ for the family of relatively coobservable[3] (and controllable, $L_m(\mathbf{G})$-closed) sublanguages of $E \cap L_m(\mathbf{G})$. Then $\mathcal{CCO}(E \cap L_m(\mathbf{G}))$ is nonempty (the empty language $\emptyset$ belongs) and has a unique supremal element given by

$$L_m(\mathbf{COSUP}) := \sup \mathcal{CCO}(E \cap L_m(\mathbf{G})) = \bigcup \{K | K \in \mathcal{CCO}(E \cap L_m(\mathbf{G}))\},$$

which may be effectively computed (Cai et al., 2016). We call **COSUP** the *controllable and coobservable behavior* of $\mathbf{G}$ under the control of $|\mathcal{I}|$ decentralized supervisors. Since relative coobservability is stronger than coobservability, $L_m(\mathbf{COSUP})$ is guaranteed to be coobservable (and controllable, $L_m(\mathbf{G})$-closed), and thereby ensures the existence of decentralized supervisors jointly synthesizing $L_m(\mathbf{COSUP})$.

[3]In this paper, only timed relative coobservability (or timed C-coobservability) is used; thus for simplicity we shall henceforth often omit the word "timed.

3. Communication Channel Models

Consider that the plant $\mathbf{G}$ consists of N component TDES $\mathbf{G}_i$ ($i \in \mathcal{N} := \{1, 2, ..., N\}$), each with event set Σ_i ($tick \in \Sigma_i$). Then the plant

$$\mathbf{G} = \mathbf{Comp}(\mathbf{G}_1, ..., \mathbf{G}_N), \tag{4}$$

where **Comp** is the composition operator defined in Section 2 to build complex TDES from simpler ones. In our previous work (Cai and Wonham (2010a); Zhang and Cai (2020); Zhang et al. (2013, 2017)), we built a distributed control architecture for $\mathbf{G}$, in which each component $\mathbf{G}_i$ is controlled by its own local controllers/preemptors. We refer to $\mathbf{G}_i$ equipped with local controllers/preemptors as an *agent*, denoted by $\mathbf{AG}_i$. In the distributed architecture, each agent interacts with some other agents through communication of shared events.

However, in (Cai and Wonham (2010a); Zhang and Cai (2020); Zhang et al. (2013, 2017)) it was assumed that the communication delay of each shared event is negligible. While simplifying the design of distributed controllers, this assumption may be unrealistic in practice, where controllers are linked by a physical network subject to delay. To capture the inter-agent communication delays, we introduce a channel model in this section.

Let $\mathbf{AG}_l$ ($l \in \mathcal{N}$) be an agent (with respect to component $\mathbf{G}_l$), and denote by $\Sigma_{com,l}$ the subset of events to be communicated to $\mathbf{AG}_l$. Let $\mathbf{AG}_k$ ($k \in \mathcal{N}$, $k \neq l$) be another agent (with respect to component $\mathbf{G}_k$) that communicates some (shared) events to $\mathbf{AG}_l$; then the subset of events communicated from agent $\mathbf{AG}_k$ to $\mathbf{AG}_l$ is

$$\Sigma_{k,com,l} = \Sigma_k \cap \Sigma_{com,l} \tag{5}$$

where Σ_k is the event set of component $\mathbf{G}_k$. In the following we focus on non-zero communication delays (because the communication events are generally transmitted through physical channels), and represent by

$$\Sigma'_{k,com,l} \subseteq \Sigma_{k,com,l}$$

the subset of events whose communication delays are greater than zero. Those events in $\Sigma_{k,com,l} \setminus \Sigma'_{k,com,l}$ are transmitted with no delay, and thus can be observed directly by the receiver; hence in this case we do not employ channel models for their transmissions.

Specifically, consider that one of the communication events, say σ, is transmitted from agent $\mathbf{AG}_k$ to $\mathbf{AG}_l$ through some communication media and with non-zero delay, i.e. $\sigma \in \Sigma'_{k,com,l}$. Physically, the occurrence of event σ is observable only by the sender $\mathbf{AG}_k$, but not by the receiver $\mathbf{AG}_l$. Instead, through the communication media, $\mathbf{AG}_l$ will receive the occurrence of σ after some time (i.e. communication delay). Denote the event of receiving σ by a new event label σ'; thus σ' is observable by the receiver $\mathbf{AG}_l$ (but not by the sender $\mathbf{AG}_k$). As a result of delay, $\mathbf{AG}_k$ and $\mathbf{AG}_l$ has distinct observable event sets, and they must take their preemptive/control actions accordingly.

Now we propose a TDES channel model $\mathbf{CH}(k, \sigma, l)$, as displayed in Fig. 1. In $\mathbf{CH}(k, \sigma, l)$, (i) event σ denotes that σ occurs in $\mathbf{AG}_k$ and is sent to the communication channel; (ii) event σ' denotes that σ is received by $\mathbf{G}_l$, and an acknowledgement message is sent back to the channel; (iii) event σ'' denotes that $\mathbf{AG}_k$ receives the acknowledgement, which simultaneously resets the channel to be idle (i.e. the channel

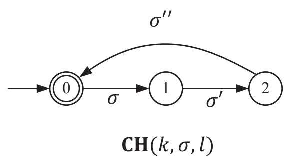

Figure 1. ATG of TDES channel model $\mathbf{CH}(k, \sigma, l)$

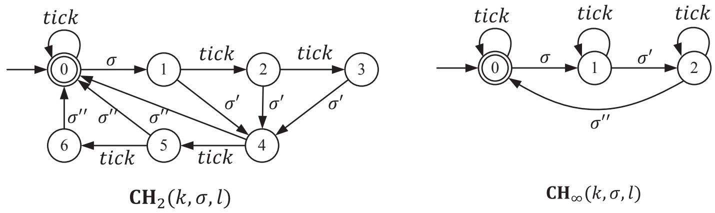

Figure 2. TTG of bounded channel model $\mathbf{CH}_d(k, \sigma, l)$ with delay $d = 2$ and unbounded channel model $\mathbf{CH}_\infty(k, \sigma, l)$; the TTG are obtained from the ATG displayed in Fig. 1 with $(l_\sigma, u_\sigma) = (0, 2)$ and $(l_\sigma, u_\sigma) = (0, \infty)$ respectively (by applying the constructing rules in (Wonham & Cai, 2019, Chapter 9)). In the models, first, the occurrence of event σ means that σ occurs in $\mathbf{G}_k$ and is sent to the communication channel; after some time delay (less than d), σ' will occur, which represents that the occurrence of σ is received by $\mathbf{G}_l$, and an acknowledgement message is sent back to the channel; finally after another time delay the occurrence of event σ'' denotes that $\mathbf{G}_k$ receives the acknowledgement, which simultaneously resets the channel to be idle.

is ready to send the next occurrence of σ). The lower and upper bounds of events σ' and σ'' are determined by the practical requirements on the communication delay bounds of σ. Note that the events σ' and σ'' are specific to $\mathbf{CH}(k, \sigma, l)$, which transmits event σ from $\mathbf{AG}_k$ to $\mathbf{AG}_l$. In other words, if we adopt another channel $\mathbf{CH}(k, \sigma, l')$ to transmit event σ from $\mathbf{AG}_k$ to $\mathbf{AG}_{l'}$, we will use other notation, e.g. $\hat{\sigma}'$ and $\hat{\sigma}''$, to replace σ' and σ'' respectively. Here for simplicity we adopt σ' and σ'' in $\mathbf{CH}(k, \sigma, l)$ as a generic case.

First, to meet a hard deadline of an operation or to ensure system's timely performance in practice, it may often be the case that the communication delay of event σ is bounded by $d \in \mathbb{N} - \{0\}$ *tick*s. In this case, the lower time bounds of σ' and σ'' are both set to be 0 and the upper bounds to be d, which means that the time consumed for transmitting the occurrence of σ from $\mathbf{AG}_k$ to $\mathbf{AG}_l$ *and* that for acknowledging the receival of σ from $\mathbf{AG}_l$ to $\mathbf{AG}_k$ should be both no more than d *tick*s.

Second, in case there happens to be no specific deadline requirement on transmission of event $\sigma \in \Sigma'_{k,com,l}$, or simply no *a priori* knowledge is available of a delay bound on σ, it may be reasonable to consider *unbounded delay* of σ-communication. This means that the transmission of σ may take *indefinite* time to complete, although it will complete eventually. So, in this case, σ' and σ'' both have lower bound 0 and upper bound ∞ (i.e. they may occur at any time after they become eligible to).

To distinguish the channel models $\mathbf{CH}(k, \sigma, l)$ of the above two cases, in notation we use $\mathbf{CH}_d(k, \sigma, l)$ to represent the channel with delay bound d, and $\mathbf{CH}_\infty(k, \sigma, l)$ the channel with unbounded delay. An example of bounded channel model of σ with delay bound $d = 2$ and unbounded channel model is given in Fig. 2.

In the channel models above, we make the following choices. (i) Both events σ' and σ'' are uncontrollable, because it is not reasonable (if not impossible) to disable the receipt of a communication or an acknowledgement; (ii) events σ, σ'' are observable to the sender $\mathbf{AG}_k$ but unobservable to the receiver $\mathbf{AG}_l$, while σ' is observable to $\mathbf{AG}_l$ but unobservable to $\mathbf{AG}_k$. This means that the agents generally have different subsets of observable events; this is a new feature of the current formulation with communication delay. Intuitively, to obtain local preemptors/controllers in this formulation, we need to iteratively apply the supervisor localization under the different partial observations; this can be realized by combining timed relative coobservability (Cai et al., 2016) with supervisor localization, as will be described in Section 4 below.

Remark 1. The communication channel models proposed above differ from those in (Zhang et al., 2016a) in the following two respects. First, the models in this paper are richer with adding an event label σ' to represent that the receiver has received the occurrence of event σ in the sender and sent an acknowledgement back to the channel. By this operation, the communication delay in transmitting the occurrence of σ and that in transmitting the acknowledgement information are modeled separately, while in (Zhang et al., 2016a) the communication delays are accumulated as a single value. Hence the models in this paper are more practical. Second, the channel models in this paper are considered as plant components and they together with the original components form the new plant to be controlled; namely, the delays will be considered as part of plant dynamics in the supervisor synthesis procedure. While in (Zhang et al., 2016a) the delays are not considered in the supervisor synthesis procedure, and thus it is not guaranteed that the synthesized supervisors can tolerate the given delays.

Remark 2. In this paper, we assume that the message or the acknowledge will eventually be received, i.e. we do not consider message lost in this paper. A new event σ_l may be added in our TDES channel model to represent that the message or the acknowledge is lost in the communication. The occurrence of σ_l may be defined on individual states (e.g. reseting the channel, or waiting). Also, σ_l can be designated to be observable or unobservable to the senders or receivers. The new model will be treated as a plant component, and the adding of σ_l will change the set of observable event set of each agent. Thus, the method proposed in this paper can be adopted to solve the distributed control problem in that case. We will investigate the details in future work.

4. Distributed Control of TDES with Communication Delay

Consider (again) that the multi-component plant $\mathbf{G}$ $(= \mathbf{Comp}(\mathbf{G}_1, ..., \mathbf{G}_N))$. Let $E \subseteq \Sigma^*$ be an arbitrary specification language imposed on $\mathbf{G}$. For $k, l \in \mathcal{N}$ $(= \{1, ..., N\})$, let $\Sigma'_{k,com,l}$ be the communication events transmitted from $\mathbf{AG}_k$ to $\mathbf{AG}_l$ subject to communication delays ($\mathbf{AG}_k$ and $\mathbf{AG}_l$ are the agents corresponding to $\mathbf{G}_k$ and $\mathbf{G}_l$ respectively). In this section, we propose a new approach to build a distributed control architecture for $\mathbf{G}$ with given communication delays.

4.1. New TDES Plant with Communication Delays

Now for $k, l \in \mathcal{N}$ let $\Sigma'_{k,com,l}$ be partitioned as $\Sigma'_{k,com,l} = \Sigma^{bd}_{k,com,l} \dot\cup \Sigma^{ud}_{k,com,l}$, where $\Sigma^{bd}_{k,com,l}$ is the subset of communication events with bounded delay and $\Sigma^{ud}_{k,com,l}$ the subset of those with unbounded delay. First, by the method proposed in Section 3, we create channel models $\mathbf{CH}(k, \sigma, l)$ for each event $\sigma \in \Sigma'_{k,com,l}$, $k, l \in \mathcal{N}$ (the delay bound d for each event is independent). The channel models are treated as plant components, and thus the new plant $\tilde{\mathbf{G}}$ includes both the plant components of $\mathbf{G}$ and the channels and is computed by

$$\begin{aligned}\tilde{\mathbf{G}} = \mathbf{Comp}(\mathbf{G}, \{&\mathbf{CH}(k,\sigma,l)|\sigma \in \Sigma^{bd}_{k,com,l}, k,l \in \mathcal{N}\},\\ &\{\mathbf{CH}(k,\sigma,l)|\sigma \in \Sigma^{ud}_{k,com,l}, k,l \in \mathcal{N}\}),\end{aligned} \tag{6}$$

where $\mathbf{CH}(k,\sigma,l)$ is the ATG displayed in Fig. 1. The event set $\tilde{\Sigma}$ of $\tilde{\mathbf{G}}$ is

$$\tilde{\Sigma} = \Sigma \ \cup \ \{\sigma', \sigma''|\sigma \in \Sigma'_{k,com,l}, k,l \in \mathcal{N}\}.$$

Since none of the added events σ' and σ'' is forcible, or prohibitible, the new subset of forcible events and prohibitible events are unchanged, i.e. $\tilde{\Sigma}_{for} = \Sigma_{for}$ and $\tilde{\Sigma}_{hib} = \Sigma_{hib}$.

The specification imposed on $\mathbf{G}$ is not changed, but should be extended to the new event set $\tilde{\Sigma}$, i.e. the specification

$$\tilde{E} = \tilde{P}^{-1}E, \tag{7}$$

where $\tilde{P} : \tilde{\Sigma}^* \to \Sigma^*$ is the natural projection.

4.2. Combing Relative Coobservability and Partial-Observation Supervisor Localization

As we have mentioned, a consequence of introducing the communication channels is that the agents $\mathbf{AG}_k$ $(k \in \mathcal{N})$ have distinct observable event sets $\tilde{\Sigma}_{o,k}$ $(k \in \mathcal{N})$ given by

$$\begin{aligned}\tilde{\Sigma}_{o,k} := (\Sigma_o \setminus \Sigma'_{com,k}) &\cup \{\sigma, \sigma''|\sigma \in \Sigma'_{k,com,l}, l \in \mathcal{N}, l \neq k\}\\ &\cup \{\sigma'|\sigma \in \Sigma'_{l,com,k}, l \in \mathcal{N}, l \neq k\}.\end{aligned}$$

In this subection, we introduce a new approach by combining relative coobservability (in Section 2) and partial-observation localization procedure presented in (Zhang & Cai, 2020) to synthesize local controllers/preemptors tolerant of communication delays.

First, for the new plant $\tilde{\mathbf{G}}$ (defined in (6)) with specification $\tilde{E}$ (defined in (7)), let $\tilde{P}_k : \tilde{\Sigma}^* \to \tilde{\Sigma}^*_{o,k}$ be the natural projection. Using relative coobservability (by an algorithm proposed in Cai et al. (2016)), we compute a coobservable and controllable sublanguage

$$L_m(\mathbf{COSUP}) := \sup \mathcal{CCO}(\tilde{E} \cap L_m(\tilde{\mathbf{G}})). \tag{8}$$

To exclude the trivial situation, we assume that $L_m(\mathbf{COSUP}) \neq \emptyset$.[4]

Second, with $\mathbf{COSUP}$ defined above, for each observable event set $\tilde{\Sigma}_{o,k}$ $(k \in \mathcal{N})$, we construct a *partial-observation decentralized supervisor* $\mathbf{COSUPO}_k$ defined over $\tilde{\Sigma}_{o,k}$. It is proved (Lin & Wonham, 1995; Rudie & Wonham, 1992) that such constructed

[4] The introduced bounded/unbounded communication delays may cause $L_m(\mathbf{COSUP}) = \emptyset$, which means that the delay requirements are too strong to be satisfied. In that case, we shall weaken the delay requirements by either decreasing delay bounds of bounded-delay channels (when the delay bound of an event σ needs to be decreased to 0, we do not employ a channel model for σ, and consequently events σ' and σ'' defined in the channel model are also removed from the alphabet) or reducing the number of unbounded-delay channels, until we obtain a nonempty $L_m(\mathbf{COSUP})$.

decentralized supervisors $\mathbf{COSUPO}_k$ collectively achieve the same controlled behavior as **COSUP** does.

Third, we adapt the *partial-observation supervisor localization procedure* proposed in (Zhang & Cai, 2020; Zhang et al., 2017) to construct local controllers/preemptors. The control actions of each decentralized supervisor $\mathbf{COSUPO}_k$ include (i) preempting event *tick* via forcible events in $\tilde{\Sigma}_{for,k} := \tilde{\Sigma}_{for} \cap \Sigma_k$, and (ii) disabling prohibitible events in $\tilde{\Sigma}_{hib,k} := \tilde{\Sigma}_{hib} \cap \Sigma_k$. Thus, each $\mathbf{COSUPO}_k$ is decomposed into a set of partial-observation local preemptor

$$\mathbf{LOC}^P_{\alpha,k} = (Y_{\alpha,k}, \Sigma_{\alpha,k}, \eta_{\alpha,k}, y_{0,\alpha,k}, Y_{m,\alpha,k}),\ \Sigma_{\alpha,k} \subseteq \tilde{\Sigma}_{o,k} \cup \{\alpha, tick\}$$

one for each forcible event $\alpha \in \tilde{\Sigma}_{for,k}$, and a set of partial-observation local controller

$$\mathbf{LOC}^C_{\beta,k} = (Y_{\beta,k}, \Sigma_{\beta,k}, \eta_{\beta,k}, y_{0,\beta,k}, Y_{m,\beta,k}),\ \Sigma_{\beta,k} \subseteq \tilde{\Sigma}_{o,k} \cup \{\beta\}$$

one for each prohibitible event $\beta \in \tilde{\Sigma}_{hib,k}$. The event set Σ_α (resp. Σ_β) is the set of events that cause state changes in η_α (resp. η_β).

By the above approach, the result is a set of partial-observation local preemptors $\mathbf{LOC}^P_{\alpha,k}$, one for each forcible event $\alpha \in \tilde{\Sigma}_{for,k}$, $k \in \mathcal{N}$, as well as a set of partial-observation local controllers $\mathbf{LOC}^C_{\beta,k}$, one for each $\beta \in \tilde{\Sigma}_{hib,k}$, $k \in \mathcal{N}$.

4.3. Main Result

The following is the main result of this section, which asserts that the collective controlled behavior of the resulting partial-observation local preemptors and local controllers, communicated through the introduced channels with bounded/unbounded delays, is identical to that of **COSUP**.

Theorem 4.1. *The set of partial-observation local preemptors* $\{\mathbf{LOC}^P_{\alpha,k} | \alpha \in \tilde{\Sigma}_{for,k}, k \in \mathcal{N}\}$ *and the set of partial-observation local controllers* $\{\mathbf{LOC}^C_{\beta,k} | \beta \in \tilde{\Sigma}_{hib,k}, k \in \mathcal{N}\}$ *derived above are equivalent to the controllable and coobservable behavior* **COSUP** *in (8) with respect to the plant* $\tilde{\mathbf{G}}$*, i.e.*

$$L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC}) = L(\mathbf{COSUP}) \tag{9}$$

$$L_m(\tilde{\mathbf{G}}) \cap L_m(\mathbf{LOC}) = L_m(\mathbf{COSUP}) \tag{10}$$

with

$$L(\mathbf{LOC}) := \Big(\bigcap_{\alpha \in \tilde{\Sigma}_{for,k}, k \in \mathcal{N}} P^{-1}_{\alpha,k} L(\mathbf{LOC}^P_{\alpha,k}) \Big) \cap \Big(\bigcap_{\beta \in \tilde{\Sigma}_{hib,k}, k \in \mathcal{N}} P^{-1}_{\beta,k} L(\mathbf{LOC}^C_{\beta,k}) \Big) \tag{11}$$

$$L_m(\mathbf{LOC}) := \Big(\bigcap_{\alpha \in \tilde{\Sigma}_{for,k}, k \in \mathcal{N}} P^{-1}_{\alpha,k} L_m(\mathbf{LOC}^P_{\alpha,k}) \Big) \cap \Big(\bigcap_{\beta \in \tilde{\Sigma}_{hib,k}, k \in \mathcal{N}} P^{-1}_{\beta,k} L_m(\mathbf{LOC}^C_{\beta,k}) \Big) \tag{12}$$

where $P_{\alpha,k} : \tilde{\Sigma}^* \to \Sigma^*_{\alpha,k}$ *and* $P_{\beta,k} : \tilde{\Sigma}^* \to \Sigma^*_{\beta,k}$.

The proof of Theorem 4.1, presented below, is similar to that of Theorem 1 in (Zhang & Cai, 2020), which relies on the facts that (i) for each forcible event, there is a corresponding partial-observation local preemptor that preempts event *tick* consistently with **COSUP**, and (ii) for each prohibitible event, there is a corresponding partial-observation local controller that disables/enables it consistently with **COSUP**.

Proof of Theorem 4.1: The equality of (10) and the ($\supseteq$) direction of (9) may be verified analogously as in the proof of Theorem 1 in (Zhang & Cai, 2020). Here we prove ($\subseteq$) of (9) by induction, i.e. $L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC}) \subseteq L(\mathbf{COSUP})$.

For the **base step**, note that none of $L(\tilde{\mathbf{G}})$, $L(\mathbf{LOC})$ and $L(\mathbf{COSUP})$ is empty; and thus the empty string ϵ belongs to all of them. For the **inductive step**, suppose that $s \in L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC})$, $s \in L(\mathbf{COSUP})$ and $s\sigma \in L(\tilde{\mathbf{G}}) \cap L(\mathbf{LOC})$ for arbitrary event $\sigma \in \Sigma$; we must show that $s\sigma \in L(\mathbf{COSUP})$. Since $\tilde{\Sigma} = \tilde{\Sigma}_{uc} \dot\cup \tilde{\Sigma}_{hib} \dot\cup \{tick\}$, we consider the following three cases.

(i) $\sigma \in \tilde{\Sigma}_{uc}$. Since $L(\mathbf{COSUP})$ is controllable, and $s\sigma \in L(\tilde{\mathbf{G}})$ (i.e. $\sigma \in Elig_{\tilde{\mathbf{G}}}(s)$), we have $\sigma \in Elig_{L_m(\mathbf{COSUP})}(s)$. That is, $s\sigma \in \overline{L_m(\mathbf{COSUP})} = L(\mathbf{COSUP})$.

(ii) $\sigma = tick$. By the hypothesis that $s, s.tick \in L(\mathbf{LOC})$, for every forcible event $\alpha \in \tilde{\Sigma}_{for,k}$, $k \in \mathcal{N}$, $s, s.tick \in P^{-1}_{\alpha,k} L(\mathbf{LOC}^P_{\alpha,k})$, i.e. $P_{\alpha,k}(s), P_{\alpha,k}(s).tick \in L(\mathbf{LOC}^P_{\alpha})$. Let $y = \eta_{\alpha}(y_{0,\alpha,k}, P_{\alpha,k}(s))$; then $\eta_{\alpha,k}(y, tick)!$. The rest of the proof is similar to case (ii) of proving Theorem 1 in (Zhang & Cai, 2020), with $\mathbf{LOC}^P_{\alpha}$ and P_{α} replaced by $\mathbf{LOC}^P_{\alpha,k}$ and $P_{\alpha,k}$ respectively.

(iii) $\sigma \in \tilde{\Sigma}_{hib}$. There must exist a partial-observation local controller $\mathbf{LOC}^C_{\sigma,k}$ for σ. It follows from $s\sigma \in L(\mathbf{LOC})$ that $s\sigma \in P^{-1}_{\sigma,k} L(\mathbf{LOC}^C_{\sigma,k})$ and $s \in P^{-1}_{\sigma,k} L(\mathbf{LOC}^C_{\sigma,k})$. So $P_{\sigma,k}(s\sigma) \in L(\mathbf{LOC}^C_{\sigma,k})$ and $P_{\sigma,k}(s) \in L(\mathbf{LOC}^C_{\sigma,k})$, namely, $\eta_{\sigma,k}(y_{0,\sigma,k}, P_{\sigma,k}(s\sigma))!$ and $\eta_{\sigma,k}(y_{0,\sigma,k}, P_{\sigma,k}(s))!$. Let $y := \eta_{\sigma,k}(y_{0,\sigma,k}, P_{\sigma,k}(s))$; then $\eta_{\sigma,k}(y, \sigma)!$ (because $\sigma \in \Sigma_{\sigma}$). The rest of the proof is similar to that in (Zhang et al., 2017) for untimed DES. □

By the above localization approach, each agent $\mathbf{G}_k$ ($k \in \mathcal{N}$) acquires a set of partial-observation local preemptors $\{\mathbf{LOC}^P_{\alpha,k} | \alpha \in \tilde{\Sigma}_{for,k}\}$ and a set of partial-observation local controllers $\{\mathbf{LOC}^C_{\beta,k} | \beta \in \tilde{\Sigma}_{hib,k}\}$. Thus we may obtain a distributed control architecture for multi-component TDES with communication delay as follows. For a fixed component $\mathbf{G}_k$, let $\tilde{\Sigma}_{k,for}, \tilde{\Sigma}_{k,hib} \subseteq \tilde{\Sigma}_k = \Sigma_k$ be its forcible event set and prohibitible event set, respectively. A convenient policy is to let each local preemptor (resp. controller) belong to the component $\mathbf{G}_k$ such that $\tilde{\Sigma}_{for,k}$ (resp. $\tilde{\Sigma}_{hib,k}$) contains the corresponding forcible (resp. prohibitible) event; an example is displayed in Fig. 3.

Up to now, we have obtained a distributed control architecture for multi-component TDES with communication delay. For illustration, the proposed combing approach is applied to study a timed workcell under bounded and unbounded communication delay in the following section.

5. Case Study: Timed Workcell with Communication Delay

We illustrate the proposed partial-observation supervisor localization procedure by a timed workcell example, adapted from (Wonham & Cai, 2019, Chapter 9). As displayed in Fig. 4, the workcell consists of two machines **M1** and **M2**, linked by

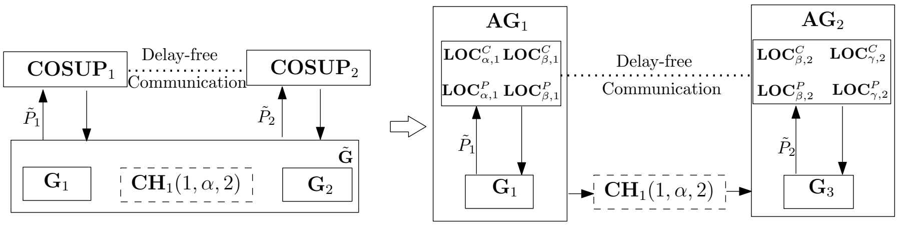

Figure 3. Example of distributed control by allocating local preemptors/controllers. Let plant $\mathbf{G}$ be composed of two components $\mathbf{G}_k$ with event sets Σ_k, $k \in [1,2]$. Suppose $\alpha, \beta \in \Sigma_{hib,1}$, $\beta, \gamma \in \Sigma_{hib,2}$, $\alpha, \beta \in \Sigma_{for,2}$ and $\beta, \gamma \in \Sigma_{for,2}$; thus $\mathbf{G}_1$ and $\mathbf{G}_2$ share event β and *tick*. Assume that event α is to be transmitted from $\mathbf{AG}_1$ to $\mathbf{AG}_2$ with delay bound 2; thus a TDES channel model $\mathbf{CH}_2(1,\alpha,2)$ is created for the transmission of α. Two partial-observation decentralized supervisors $\mathbf{COSUP}_1$ and $\mathbf{COSUP}_2$ are constructed, and then they are decomposed into partial-observation local preemptors and partial-observation local controllers, with respect to their corresponding forcible events and prohibitible events. Then a convenient allocation is displayed, where $\alpha, \beta \in \tilde{\Sigma}_{hib,1}$, $\beta, \gamma \in \tilde{\Sigma}_{hib,2}$, $\alpha, \beta \in \tilde{\Sigma}_{for,2}$ and $\beta, \gamma \in \tilde{\Sigma}_{for,2}$. Each local controller/preemptor is owned by exactly one component, and two agents $\mathbf{AG}_1$ and $\mathbf{AG}_2$ are finally created.

a one-slot buffer **BUF**; additionally, a worker **WK** is responsible for repairing **M1** and **M2**. The ATG of the machines and the worker are displayed in Fig. 5. The workcell operates as follows. Initially the buffer is empty. With the event α_1, **M1** takes a workpiece from the infinite workpiece source. Subsequently **M1** either breaks down (event λ_1), or successfully completes its work cycle, deposits the workpiece in the buffer (event β_1). **M2** operates similarly, but takes its workpiece from the buffer (event α_2), and deposits it when finished in the infinite workpiece sink. If a machine **Mi**, $i = 1$ or 2 breaks down (event λ_i), then the worker **WK** will start to repair the machine (event μ_i), and finish the repair (event η_i) in due time. Assign lower and upper time bounds to each event, with notation (event, lower bound, upper bound), as follows:

$$\begin{aligned}
&\textbf{M1}\text{'s timed events}:\\
&\quad(\alpha_1,0,\infty)\ \ (\beta_1,1,2)\ \ (\lambda_1,0,2)\ \ (\mu_1,0,\infty)\ \ (\eta_1,1,\infty)\\
&\textbf{M2}\text{'s timed events}:\\
&\quad(\alpha_2,0,\infty)\ \ (\beta_2,1,1)\ \ (\lambda_2,0,1)\ \ (\mu_2,0,\infty)\ \ (\eta_2,2,\infty)\\
&\textbf{WK}\text{'s timed events}:\\
&\quad(\mu_1,0,\infty)\ \ (\eta_1,1,2)\ \ (\mu_2,0,\infty)\ \ (\eta_2,2,3)
\end{aligned}$$

Then the TDES models of the two machines and the worker can be generated (Wonham & Cai, 2019); their joint behavior is the composition of the three TDES, which is the plant **PLANT** to be controlled, i.e.

$$\mathbf{PLANT} = \mathbf{Comp}(\mathbf{M1}, \mathbf{M2}, \mathbf{WK}).$$

Note that **Mi** ($i = 1, 2$) shares events μ_i and η_i with **WK**; so according to the composition rule described in Section II, the lower and upper bounds of μ_i and η_i are unified as: $(\mu_1, 0, \infty)$ $(\eta_1, 1, 2)$ $(\mu_2, 0, \infty)$ $(\eta_2, 2, 3)$.

To impose behavioral constraints on the two machine's joint behavior, we take $\Sigma_{for} = \Sigma_{hib} = \{\alpha_i, \mu_i | i = 1, 2\}$, and $\Sigma_{uc} = \{\beta_i, \lambda_i, \eta_i | i = 1, 2\}$, $\Sigma_{uo} = \{\mu_1, \eta_2\}$. We impose the following control specifications: (S1) **BUF** must not overflow or underflow; (S2) if **M2** goes down, its repair must be started "immediately", and

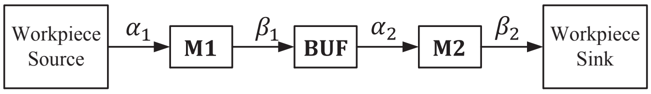

Figure 4. Workcell: system configuration

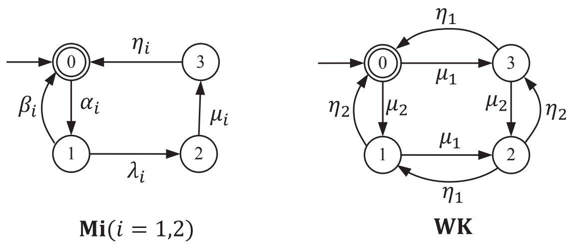

Figure 5. ATG of plant components

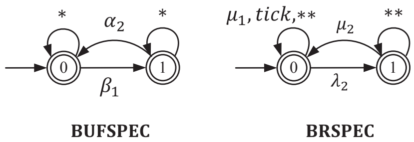

Figure 6. Control specifications: $* = \{tick, \alpha_1, \lambda_1, \mu_1, \eta_1, \beta_2, \lambda_2, \mu_2, \eta_2\}$, and $** = \{\alpha_1, \beta_1, \lambda_1, \eta_1, \alpha_2, \beta_2, \eta_2\}$

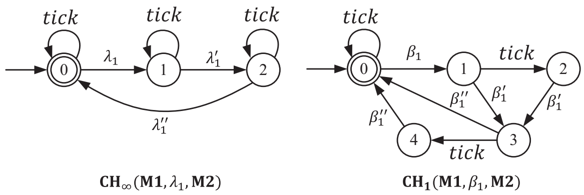

Figure 7. TTG of TDES channel models $\mathbf{CH}_\infty(\mathbf{M1}, \lambda_1, \mathbf{M2})$, and $\mathbf{CH}_1(\mathbf{M1}, \beta_1, \mathbf{M2})$.

prior to starting repair of **M1** if **M1** is currently down. These two specifications are formalized as generators **BUFSPEC** and **BRSPEC** respectively, as displayed in Fig. 6. So the overall specification imposed on the **PLANT** is represented by $\mathbf{SPEC} = \mathbf{BUFSPEC}||\mathbf{BRSPEC}$, where '||' denotes the synchronous product of two generators (Wonham & Cai, 2019).

Considering inter-agent communications, we assume that the transmissions of the events β_1, λ_1 are subject to non-zero delay (at least one of these two events must occur after **M1** has obtained a workpiece from the source). For the communication delays, consider that (i) event β_1 is transmitted from **M1** to **M2** with delay bound $d = 1$ ($tick$), and (ii) event λ_1 is transmitted from **M1** to **M2** with unbounded delay bound. The rest of the communication events are assumed (for simplicity) to be transmitted with no delay.

First, for event communications, we create TDES channel models $\mathbf{CH}(\mathbf{M1}, \beta_1, \mathbf{M2})$, and $\mathbf{CH}(\mathbf{M1}, \lambda_1, \mathbf{M2})$ to transmit events β_1 and λ_1, respectively. The lower and upper bounds of the newly added events are listed in Table 1, and the TTG of the channel models are displayed in Fig. 7.

Table 1. Time bounds of newly added events

event label	(lower, upper) bounds	event label	(lower, upper) bounds
β_1'	(0,1)	β_1''	(0,1)
λ_1'	(0,∞)	λ_1''	(0,∞)

Table 2. Subsets of observable, forcible, prohibitible events of each component

components	observable events	forcible events	prohibitible events
WK	$tick, \alpha_1, \beta_1, \lambda_1, \eta_1, \alpha_2, \beta_2, \lambda_2, \mu_2$	μ_1, μ_2	μ_1, μ_2
M1	$tick, \alpha_1, \beta_1, \beta_1'', \lambda_1, \lambda_1'', \eta_1, \alpha_2, \beta_2, \lambda_2, \mu_2$	α_1	α_1
M2	$tick, \alpha_1, \beta_1', \lambda_1', \eta_1, \alpha_2, \beta_2, \lambda_2, \mu_2$	α_2	α_2

Then, the new plant to be controlled is

$$\mathbf{NPLANT} = \mathbf{Comp}(\mathbf{M1}, \mathbf{M2}, \mathbf{WK}, \quad \mathbf{CH}(\mathbf{M1}, \beta_1, \mathbf{M2}), \mathbf{CH}(\mathbf{M1}, \lambda_1, \mathbf{M2}))$$

and the new specification is represented by **NSPEC**, modified from **SPEC** (representing E) by adding selfloops of β_1', β_1'', λ_1' and λ_1'' to each state of **SPEC** (as defined in Fig. 6). The subsets of observable events, forcible events and prohibitible events are listed in Table 2. With these event sets, we compute the controllable and coobservable controlled behavior **COSUP** as in (8), which has 45 states and 78 transitions.

Next, we apply the proposed partial-observation supervisor localization procedure presented in Section 4 to construct a set of partial-observation local preemptors, one for each forcible event in $\tilde{\Sigma}_{for}$ and a set of partial-observation local controllers, one for each prohibitible event in $\tilde{\Sigma}_{hib}$. The results are displayed in Fig. 9; it is inspected from the TTG of the local preemptors/controllers that for the communication events transmitted by the channels, only the events representing the receiving of an event occurrence (e.g. β_1' and λ_1') cause state changes in the local controllers/preemptors corresponding to the receivers. It is verified that the collective controlled behavior of these local preemptors and controllers is equivalent to **COSUP**. The control logics of the partial-observation local preemptors and controllers are affected by the communication delays (for comparison, we add the TTG of the partial-observation local preemptors and controllers in the delay-free case as shown in Fig. 8; their detailed control logics are referred to (Zhang & Cai, 2019)). For illustration, we consider the following two instances.

(i) Communication delays of β_1 and λ_1 affect the control logic of $\mathbf{NLOC}_{\alpha_2}^C$ and the preemptive logic of $\mathbf{NLOC}_{\alpha_2}^P$. According to the control logic of $\mathbf{LOC}_{\alpha_2}^C$ described in Fig. 8, **M2** will take a workpiece from the buffer if it observes (event β_1') that **M1** has deposited a workpiece into the buffer (event β_1). However, now $\mathbf{NLOC}_{\alpha_2}^C$ cannot observe β_1 directly, and may know (through the communication channels) the occurrence of event λ_1 before that of β_1. Namely, it cannot judge which event of β_1 and λ_1 has occurred if it does not receive their communicated events β_1' and λ_1', so the control logic of $\mathbf{NLOC}_{\alpha_2}^C$ becomes more complicated: it will enable/disable event α_2 according to the order of receiving of β_1 and λ_1. Due to the change of $\mathbf{NLOC}_{\alpha_2}^C$, now the occurrence of α_2 will not preempt the event *tick*, as described by $\mathbf{NLOC}_{\alpha_2}^P$ (note that the logic of $\mathbf{LOC}_{\alpha_2}^P$ (as in Fig. 8) is to preempt *tick* when the buffer is full and **M1** has taken a workpiece from the source).

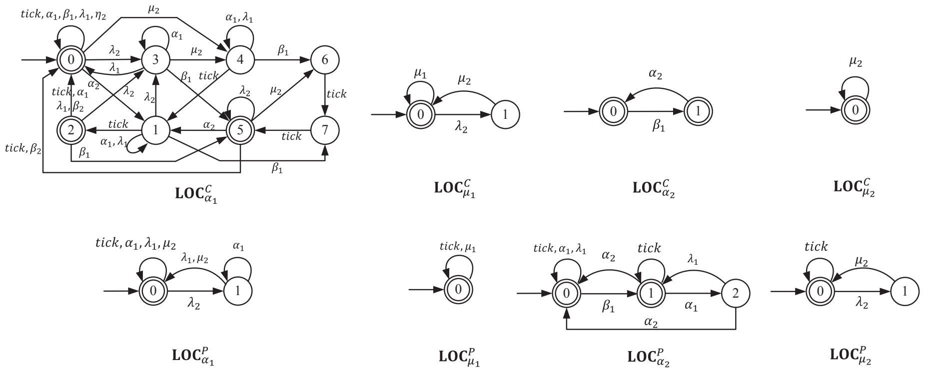

Figure 8. Local preemptors and local controller without communication delay (Fig. 2 in Zhang and Cai (2020))

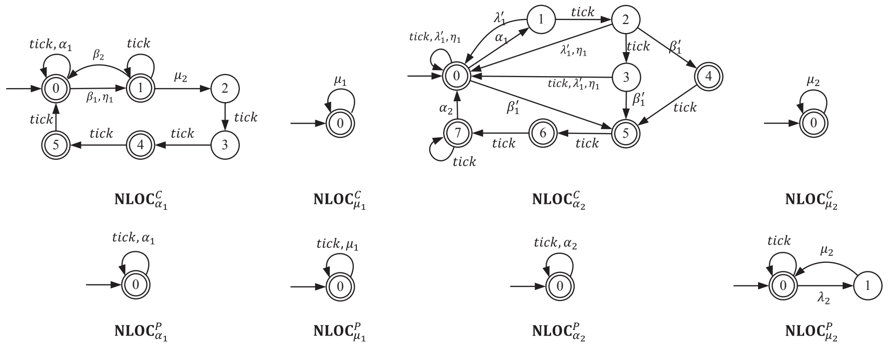

Figure 9. Local preemptors and local controllers subject to communication delays

(ii) The communication delays of β_1 and λ_1 also affect the control logic of $\mathbf{NLOC}^C_{\alpha_1}$ and the preemptive logic of $\mathbf{NLOC}^P_{\alpha_1}$. As described in (i), the occurrence of α_2 cannot preempt event *tick*; thus $\mathbf{NLOC}^C_{\alpha_1}$ will enable event α_1 only when the buffer is empty (the plant is at the initial state or the workpiece in the buffer has been taken away). This change also causes that the occurrence of α_1 need not preempt event *tick*, as described by $\mathbf{NLOC}^P_{\alpha_1}$ (local preemptor $\mathbf{LOC}^P_{\alpha_1}$ (as in Fig. 8) describes that the occurrence of α_1 may preempt *tick* event when **M2** breaks down).

Finally, by the allocation policy described in Section 3, we allocate the obtained local controllers and preemptors to the plant components **M1**, **M2**, and **WK**, thereby building a distributed control architecture under partial observation and communication delay for the timed workcell, as displayed in Fig. 10. A local preemptor/controller may observe directly an event from the agent owning it, and import an event from other agent through communication channels subject to delay. Note that we selected for simplicity only two communication events (β_1 and λ_1) to be transmitted through channels. By the same procedure described above, however, one may easily add more communication events transmitted through channels (i.e. by creating new channel models and then applying the localization procedure with communication delay again).

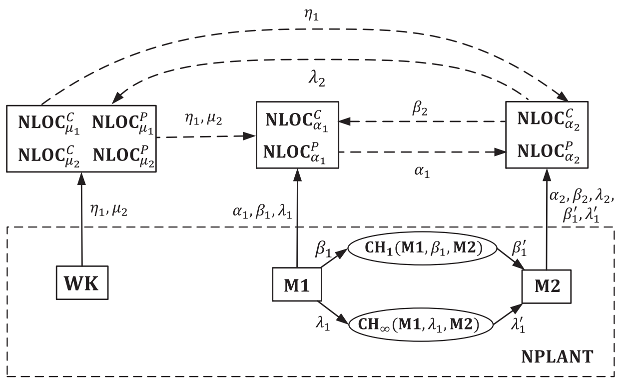

Figure 10. Distributed control architecture with communication delay.

6. Conclusions

In this paper, we have extended the partial-observation supervisor localization to the case where inter-agent event communication is subject to bounded and unbounded delay. To address communication delay, we have developed an extended localization procedure based on explicit channel models and relative coobservability. We have proved that the resulting local controllers/preemptors collectively satisfy the communication delay requirements. The above results are both illustrated by a timed workcell example.

In future research we shall extend the proposed partial-observation localization procedure to study distributed control of large-scale systems under partial observation and communication delays, by combing the proposed supervisor localization with some efficient heterarchical synthesis procedure, e.g. (Feng & Wonham, 2008).

Funding

This work was supported in part by the Fundamental Research Funds for the Central Universities of China, Grant no. 3102019ZDHKY11; the Postdoctoral Science Foundation of China, Grant no. 2019M663819;the National Natural Science Foundation of China, Grant no. 11772264; JSPS KAKENHI, Grant no. JP16K18122.

References

Barrett, G., & Lafortune, S. (2000). Decentralized supervisory control with communicating controllers. *IEEE Transactions on Automatic Control*, *45*(9), 1620–1638.

Brandin, B., & Wonham, W. (1994). Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, *39*(2), 329–342.

Cai, K., & Wonham, W. (2010a). Supervisor localization: a top-down approach to distributed control of discrete-event systems. *IEEE Transactions on Automatic Control*, *55*(3), 605–618.

Cai, K., & Wonham, W. (2010b). Supervisor localization for large discrete-event systems: case study production cell. *International Journal of Advanced Manufacturing Technology*, *50*(9-12), 1189–1202.

Cai, K., & Wonham, W. (2016). *Supervisor localization: A top-down approach to distributed control of discrete-event systems.* Lecture Notes in Control and Information Sciences, vol. 459, Springer.

Cai, K., Zhang, R., & Wonham, W. (2015). Relative observability of discrete-event systems and its supremal sublanguages. *IEEE Transactions on Automatic Control*, *60*(3), 659-670.

Cai, K., Zhang, R., & Wonham, W. (2016). Relative observability and coobservability of timed discrete-event systems. *IEEE Transactions on Automatic Control*, *61*(11), 3382-3395.

Feng, L., & Wonham, W. (2008). Supervisory control architecture for discrete-event systems. *IEEE Transactions on Automatic Control*, *53*(6), 1449-1461.

Hiraishi, K. (2009). On solvability of a decentralized supervisory control problem with communication. *IEEE Transactions on Automatic Control*, *54*(3), 468–480.

Kalyon, G., Gall, T. L., Marchand, H., & Massart, T. (2011). Synthesis of communicating controllers for distributed systems. In *Proc. 50th ieee conference on decision and control and european control conference.* Orlando, FL, USA.

Komenda, J., & Masopust, T. (2017). Computation of controllable and coobservable sublanguages in decentralized supervisory control via communication. *Discrete Event Dynamic Systems*, *27*(4), 585-608.

Komenda, J., Masopust, T., & van Schuppen, J. (2012). On conditional decomposability. *Systems & Control Letters*, *62*(12), 1260-1268.

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., & Wonham, W. (1995). Supervisory control of timed discrete-event systems under partial observation. *IEEE Transactions on Automatic Control*, *40*(3), 558-562.

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.

Ricker, L., & Caillaud, B. (2011). Mind the gap: expanding communication options in decentralized discrete-event control. *Automatica*, *47*(11), 2364-2372.

Rudie, K., & Wonham, W. (1992). Think globally, act locally: Decentralized supervisory control. *IEEE Transactions on Automatic Control*, *37*(11), 1692-1708.

Sadid, W., Ricker, L., & Hashtrudi-Zad, S. (2015). Robustness of synchronous communication protocols with delay for decentralized discrete-event control. *Discrete Event Dynamic Systems*, *25*(1), 159–176.

Schmidt, K., Schmidt, E., & Zaddach, J. (2007). A shared-medium communication architecture for distributed discrete event systems. In *Proc. mediterranean conference on control and automation* (p. 1-6). Athens, Greece.

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

Wonham, W., & Cai, K. (2019). *Supervisory control of discrete-event systems.* Springer.

Xu, S., & Kumar, R. (2008). Asynchronous implementation of synchronous discrete event control. In *Proc. 9th international workshop on discrete event systems* (p. 181-186).

Zhang, R., & Cai, K. (2019). *Supervisor localization of timed discrete-event systems under partial observation and communication delay.* (Technical Report. Available at `http://arxiv.org/abs/1603.02023`)

Zhang, R., & Cai, K. (2020). Supervisor localization of timed discrete-event systems under partial observation. *IEEE Transactions on Automatic Control*, *65*(1), 295-301.

Zhang, R., Cai, K., Gan, Y., Wang, Z., & Wonham, W. (2013). Supervision localization of timed discrete-event systems. *Automatica*, *49*(9), 2786–2794.

Zhang, R., Cai, K., Gan, Y., & Wonham, W. (2016a). Delay-robustness in distributed control of timed discrete-event systems based on supervisor localisation. *International Journal of Control*, *89*(10), 2055–2072.

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

Zhang, R., Cai, K., & Wonham, W. (2017). Supervisor localization of discrete-event systems under partial observation. *Automatica*, *81*, 142-147.